Etude des systèmes de recommandations et mise en pratique des algorithmes.

Auteur : Bastin, Jérémy
Promoteur(s) : Blavier, André
Faculté : HEC-Ecole de gestion de l'Université de Liège
Diplôme : Master en sciences de gestion (Horaire décalé)
Année académique : 2020-2021
URI/URL : http://hdl.handle.net/2268.2/11045

Etude des systèmes de recommandations et mise en pratique des algorithmes

Promoteur : André BLAVIER

Lecteur : Ashwin ITTOO

Travail de fin d'études présenté par **Jérémy BASTIN** en vue de l'obtention du diplôme de Master en Sciences de gestion

Option M60 Horaire décalé.

Année académique 2019 / 2020

Je tiens à remercier mon promoteur, Monsieur André BLAVIER, professeur à HEC-Liège, pour sa disponibilité, ses conseils et son suivi.

Je remercie mon lecteur, Monsieur Ashwin ITTOO, professeur à HEC-Liège, pour m'avoir encouragé à étudier la piste du machine learning.

Je remercie mon entreprise, EVS Broadcast Equipment, qui m'a donné la flexibilité nécessaire pour terminer ce travail.

Enfin, je remercie de tout cœur mes parents et ma famille pour leur soutien durant ces études.

Jérémy Bastin

1. Table des matières

1. Table des matières **4**

3. Définitions **8**

4. Introduction aux systèmes de recommandation **8**

4.1. Contexte 8

4.2. Valeur ajoutée 8

4.3. Formulation du problème 9

5. Historique **10**

5.1. De la popularité à l'engagement 10

6. Système de recommandations : Définition et interprétation **11**

6.1. L'ère de l'ultra personnalisation 12

7. Difficultés liées à la construction d'un système de recommandations **13**

7.1. Sérendipité 13

7.2. Sparsité 13

7.3. Démarrage à froid 14

7.3.1. Démarrage à froid des utilisateurs 14

7.3.2. Démarrage à froid des items 16

8. Processus de création d'un système de recommandations **16**

8.1. Définition des besoins business 16

8.2. Définition des prérequis 17

8.3. Prototypage 18

8.4. Déploiement 18

8.5. Développement continu par itérations 19

9. Notions théoriques pour la construction d'un système de recommandations **21**

9.1. Collecte des données 21

9.1.1. Caractéristiques des données explicites 21
9.1.2. Caractéristiques des données implicites 22
9.1.3. Différences d'interprétation entre les données explicites et implicites 23
9.2. Traitement des données 23
9.2.1. Retrait des mots vides 23
9.2.2. Transformation des données en vecteurs 24
9.2.3. Pondération des mots via la méthode TF-IDF 24
9.3. Calcul de similarité 25
9.3.1. Coefficient de corrélation de Pearson 25
9.3.2. Similarité basée sur le cosinus 26
9.4. Métriques de précision 27
10. Etude des algorithmes de recommandations 28
10.1. Les différentes approches 28
10.2. Approche basée sur le contenu 29
10.3. Approche basée sur le filtrage collaboratif 30
10.3.1. Approche basée sur la mémoire 31
10.3.2. Approche basée sur le modèle 33
10.4. Approche hybride 37
11. Systèmes de recommandations sensibles au contexte 37
12. Systèmes de recommandations basées sur le deep learning 39
12.1. Introduction au deep learning 39
12.2. Etude du système de Youtube 39
12.2.1. Architecture 41
12.2.2. Détails sur la génération de candidats 42
12.2.3. Détails sur le classement des prédictions 42
12.2.4. Impact du deep learning chez Youtube 42

12.3. Word embedding 43
12.4. Usage avancé du deep learning 44
12.4.1. Etude du cas Spotify 44
11.4.1. Traitement du langage naturel 44
11.4.2. Traitement des pistes audios 45
13. Impacts sur les comportements sociaux et culturels 45
13.1. Isolement intellectuel et désinformation 47
13.2. Du mentor au coach 48
14. Expérimentation des techniques de recommandations 49
14.1. Présentation et objectif de l'expérience 49
14.2. Méthodologie 49
14.3. Données 50
14.3.1. Sélection du jeu de données 50
14.3.2. Exploration des données 51
14.4. Technologies utilisées (lecture optionnelle) 58
14.5. Résultats et discussions 58
14.5.1. Recommandations basées sur le contenu 58
14.5.2. Recommandations item-item basées sur le filtrage collaboratif 63
14.5.3. Recommandations basées sur le filtrage collaboratif par facteurs latents 65
14.5.4. Recommandations collaboratives basées sur le deep learning 68
14.6. Discussions de l'expérimentation 75
15. Futur des systèmes de recommandations 76
15.1. Lorsque la recommandation devient le contenu 76
15.1.1. Techniques prometteuses 77
15.2. Les nouveaux coachs au quotidien 78
16. Conclusion 79

17. Bibliographie **81**

3. Définitions

- ➢ Corpus - ensemble fini de textes choisis comme base d'une étude.
- ➢ Cluster - regroupement de données, parfois appris par apprentissage automatique.
- ➢ Item - terme générique utilisé pour représenter tout type de produit ou de contenu pouvant être potentiellement recommandés.

4. Introduction aux systèmes de recommandation

4.1. Contexte

Revenez un instant sur vos activités de la semaine. Un algorithme aura certainement déterminé les chansons que vous aimeriez écouter, les plats à commander en ligne, les séries ou films que vous aimeriez regarder, les publications que vous voyez sur vos réseaux sociaux préférés, ainsi que la prochaine personne avec qui vous voudrez peut-être vous connecter. Les recommandations guident déjà tant d'aspects de notre vie sans que nous en soyons nécessairement conscients. Toutes ces applications sont pilotées par un type d'algorithmes communs : les systèmes de recommandations.

4.2. Valeur ajoutée

Le Harvard Business Review a qualifié les recommandations comme étant la distinction algorithmique la plus importante entre les entreprises historiques et les entreprises “nées du digital” (*Harvard Business Review*, 1 Aug. 2017). HBR a aussi décrit le cercle vertueux que ces dernières génèrent : plus les utilisateurs utilisent le système de recommandations d’une entreprise, plus sa valeur augmente et plus sa valeur augmente, plus les utilisateurs l’utilisent.

Nous sommes encouragés à examiner les systèmes de recommandations, non pas comme un moyen de vendre, mais plutôt comme une ressource renouvelable pour améliorer sans cesse les connaissances des clients et nos propres connaissances. De nombreuses entreprises existantes ont des milliers d'utilisateurs et donc des milliers de données et ne

sont pourtant pas parvenus à atteindre le succès espéré. La raison pour laquelle leur cycle vertueux n'a pas pris autant d'importance que ceux d'Amazon, Netflix ou Spotify est le manque de connaissances sur la façon de convertir leurs données utilisateur en informations exploitables, qui peuvent ensuite être utilisées pour améliorer leurs produits ou services.

Netflix a montré à quel point cela est crucial, 80% de ce que les gens regardent provient de recommandations (*The Netflix Recommender System*, Article 13). En 2015, l'un de leurs articles citait : "Nous pensons que l'effet combiné de la personnalisation et des recommandations nous permet d'économiser plus d'un milliard de dollars par an." Si nous regardons Amazon, 35% de ce que les clients achètent proviennent de recommandations de produits (*Two Decades of Recommender Systems at Amazon.com*). Sur Airbnb, le classement de recherche et les listes similaires génèrent 99% de toutes les conversions de réservation (*Listing Embeddings in Search Ranking*, Airbnb mars 2018).

4.3. Formulation du problème

Maintenant que nous avons vu l'immense valeur que les entreprises peuvent tirer des systèmes de recommandations, examinons le type de problème qu'elles peuvent résoudre. De manière générale, les entreprises technologiques essaient de recommander le contenu le plus pertinent à leurs utilisateurs. Cela pourrait signifier recommander des :

- ➢ Annonces de propriétés similaires (Airbnb, Zillow)
- ➢ Médias pertinents, par exemple photos, vidéos (Instagram)
- ➢ Séries et films pertinents (Netflix, Amazon Prime Video)
- ➢ Chansons et podcasts pertinents (Spotify)
- ➢ Vidéos pertinentes (YouTube)
- ➢ Utilisateurs similaires, publications (LinkedIn, Twitter, Instagram)
- ➢ Plats et restaurants pertinents (Uber Eats)

La formulation du problème est ici critique. La plupart du temps, les entreprises souhaitent recommander du contenu que les utilisateurs sont les plus susceptibles d'apprécier à l'avenir. La reformulation de ce problème, ainsi que les changements

algorithmiques passant de la recommandation de “ce que les utilisateurs sont les plus susceptibles de regarder” à “ce que les utilisateurs sont le plus susceptible de regarder à l'avenir”. “Les chercheurs d'Amazon ont constaté que l'utilisation de réseaux de neurones pour générer des recommandations de films fonctionnait beaucoup mieux lorsqu'ils traitaient les données d'entrée par ordre chronologique et les utilisaient pour prédire les futures préférences de films sur une courte période (une à deux semaines).” (*The history of Amazon's recommendation algorithm*).

5. Historique

5.1. De la popularité à l’engagement

Jusqu’en 2012, YouTube classait les vidéos selon un paramètre unique, le nombre de vues. Alors que cette méthode visait à récompenser les vidéos de meilleure qualité et placer les plus populaires sous les yeux du public, elle a au lieu de ça, entraîné un problème de piège à clics. Si le titre d’une vidéo était trompeur, il pouvait inciter l’utilisateur à cliquer dessus, mais ce dernier mettait rapidement fin à son visionnage. Cette stratégie avait donc des répercussions négatives sur la qualité, et donc pour les annonceurs et pour la plateforme elle-même. YouTube a donc agi et retravaillé son algorithme pour privilégier la durée de visionnage ainsi que le temps passé sur la plateforme en général (autrement dit, la durée de session).

En 2016, YouTube publia un livre blanc célèbre pour expliquer le rôle du deep learning et du machine learning dans son système de recommandations (*Deep Neural Networks for YouTube Recommendations*). Le système de recherche et de détection de YouTube vise deux objectifs : aider les internautes à trouver les vidéos qu’ils souhaitent visionner, ainsi que favoriser l’interaction avec les spectateurs et leur satisfaction à long terme. L’idée n’est pas d’identifier de “bonnes” vidéos, mais de proposer des vidéos qui correspondent aux envies des utilisateurs, le but ultime étant qu’ils passent le plus de temps possible sur la plateforme et visionnent donc plus de publicités. Depuis lors, cet objectif principal de recherche de l’engagement de l’utilisateur n’a jamais changé chez la plupart des principaux acteurs du marché.

6. Système de recommandations : Définition et interprétation

A quel moment peut-on vraiment parler de système de recommandations ? Google est-il un système de recommandations ou un moteur de recherche ? La définition de Wikipédia nous livre une première interprétation : “Les systèmes de recommandation sont une forme spécifique de filtrage de l'information visant à présenter les éléments d'information qui sont susceptibles d'intéresser l'utilisateur”. Cette définition nous donne les informations essentielles d’un système de recommandations, à savoir le filtrage de l’information combiné à la recherche d’intérêt de l’utilisateur.

Un pur moteur de recherche documentaire donnerait les résultats les plus pertinents en se basant uniquement sur la pertinence des documents, mais sans rechercher l'intérêt de l’utilisateur, celui-ci ne reste qu’un moteur de recherche. Google n’a jamais été un pur moteur de recherche puisque la première version utilisait déjà le célèbre algorithme du PageRank qui permet de classer les résultats en fonction de leur popularité. Cette recherche de popularité a grandement suscité l'intérêt des utilisateurs et a permis de propulser le site en tant que leader mondial de la recherche. Sur base de ces observations, on peut considérer Google en tant que système de recommandations axé sur la pertinence.

6.1. L'ère de l’ultra personnalisation

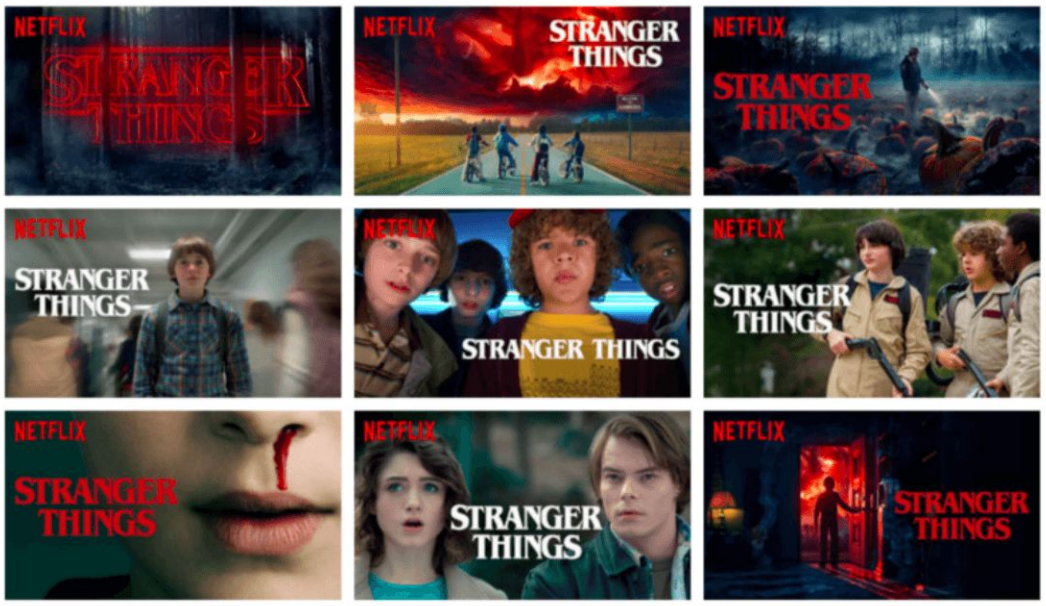

Figure 1. Netflix Atwork ; Chaque couverture du film “Stranger Things” reçoit 5% d’impression par l'algorithme de personnalisation (Artwork Personalization at Netflix, décembre 2017).

Netflix Atwork est un bon exemple de personnalisation moderne là où on ne s’y attend peut-être pas. En plus de personnaliser votre sélection de films, Netflix personnalise les couvertures selon votre historique de visualisation. Netflix y attache une attention particulière puisque c’est ce visuel qui sert principalement de première accroche. Pour la série “Stranger Things” dont une sélection de couvertures candidates est présentée ci-dessus, on retrouve des variantes ciblant différents types de personnalité. Un utilisateur qui aurait regardé de nombreux films romantiques pourrait être attiré par la couverture avec la fille et le garçon, alors qu’un amateur de films dramatiques serait sûrement davantage intéressé par le visage avec du sang. Ce genre d'avancée est permis grâce à l’apprentissage automatique qui a pris une grande importance dans les systèmes de recommandations modernes. Ces avancées technologiques nous ont fait entrer dans l'ère de l'ultra-personnalisation.

7. Difficultés liées à la construction d'un système de recommandations

Construire un système de recommandations est une tâche qui peut s'avérer particulièrement complexe pour les entreprises qui n'ont ni les mêmes besoins, ni les mêmes objectifs. Malgré ces différences, certains problèmes sont récurrents à la majorité des systèmes de recommandations et c'est ce que nous allons voir dans ce chapitre.

7.1. Sérendipité

La sérendipité désigne la faculté à faire des découvertes heureuses par accident. S'il peut être plus facile de rester dans sa zone de confort idéologique, il y a des avantages à embrasser le hasard de cette manière. Que cela nous plaise ou non, nous nous baignons réellement dans le hasard tous les jours. Par exemple, à la maison lorsque vous recherchez un objet spécifique pour se retrouver face à face avec un objet précédemment perdu. Ou le soir quand on cherche un ami, mais qu'on finit par en trouver un autre avec qui la discussion s'avère ennuyeuse. Le hasard n'est pas synonyme de découvertes heureuses mais peut produire des résultats à la fois négatifs et positifs. Par conséquent, un algorithme de recommandations efficace devrait non seulement recommander ce que nous sommes susceptibles d'apprécier, mais aussi suggérer des éléments aléatoires tout en étant objectifs pour nous aider à garder une fenêtre ouverte sur d'autres mondes et de nouvelles découvertes.

Bien qu'étant une qualité importante, le manque de sérendipité est un syndrome courant dans les systèmes de recommandations actuels qui privilégient souvent l'engagement à tout prix et préfèrent ne prendre aucun risque de décevoir l'utilisateur. Youtube est selon mon opinion le parfait exemple du système de recommandations proposant difficilement du contenu pour lequel il n'est pas sûr que vous vous engagiez.

7.2. Sparsité

Les utilisateurs d'une application ne vont généralement interagir qu'avec un pourcentage très faible du catalogue d'items proposé. Spotify compte 60 millions de chansons (*Spotify*

Company Info, 2020), Pinterest 200 milliards de pins (*Pinterest business*, 2020), autant dire que l'utilisateur ne rencontrera jamais la grande majorité du contenu proposé. Ces données une fois représentées sous forme de matrices ont un très fort degré d'interactions nulles. Ce genre de matrices est appelé matrice creuse et les algorithmes de recommandations collaboratives réalisent des calculs sur base de celles-ci. Or quand on veut manipuler ou stocker des matrices creuses à l'aide de programmes informatiques, il est avantageux, voire souvent nécessaire d'utiliser des algorithmes et des structures de données qui prennent en compte la structure éparse de ces matrices. Les opérations sur les matrices sont lentes et utilisent beaucoup de mémoire.

Les matrices creuses ont néanmoins l'avantage de pouvoir être facilement compressibles de par leur sparsité. La structure de données naïve utilisée pour stocker une matrice est un tableau bidimensionnel où chaque entrée du tableau représente un élément de la matrice. Pour une matrice $m \, x \, n$ il faut au moins $m \, x \, n$ espaces mémoires de taille fixe pour la représenter. Beaucoup, si ce n'est la majorité des entrées d'une matrice creuse, sont des zéros ou des valeurs nulles. L'idée de base est alors de ne stocker que les entrées non nulles de la matrice, plutôt que d'en stocker l'intégralité. En fonction du nombre et de la répartition des entrées non nulles, des structures de données différentes peuvent être utilisées et amènent de grandes économies dans la taille utilisée en mémoire par rapport à la structure naïve. Un exemple d'une telle représentation est le format "Yale Sparse Matrix" qui stocke une matrice de taille $m \, x \, n$ sous la forme de trois tableaux unidimensionnels.

7.3. Démarrage à froid

7.3.1. Démarrage à froid des utilisateurs

Le problème du démarrage à froid est souvent observé dans les systèmes de recommandations où les méthodes telles que le filtrage collaboratif reposent fortement sur les interactions utilisateur-item antérieures. Les entreprises sont confrontées au problème du démarrage à froid de deux manières selon les plateformes : démarrage à froid par l'utilisateur et par item. Lorsqu'un nouveau membre s'inscrit par exemple sur Netflix,

l'entreprise ne sait rien des préférences de ce nouveau membre. Comment l'entreprise peut-elle maintenir l'utilisateur engagé en lui fournissant d'excellentes recommandations ? Dans le cas de Netflix, les nouveaux membres bénéficient d'un essai gratuit d'un mois, au cours duquel les taux d'annulation sont les plus élevés et diminuent rapidement par la suite. C'est pourquoi toute amélioration du problème du démarrage à froid présente une immense opportunité commerciale afin d'augmenter l'engagement et la rétention au cours de ces 30 premiers jours.

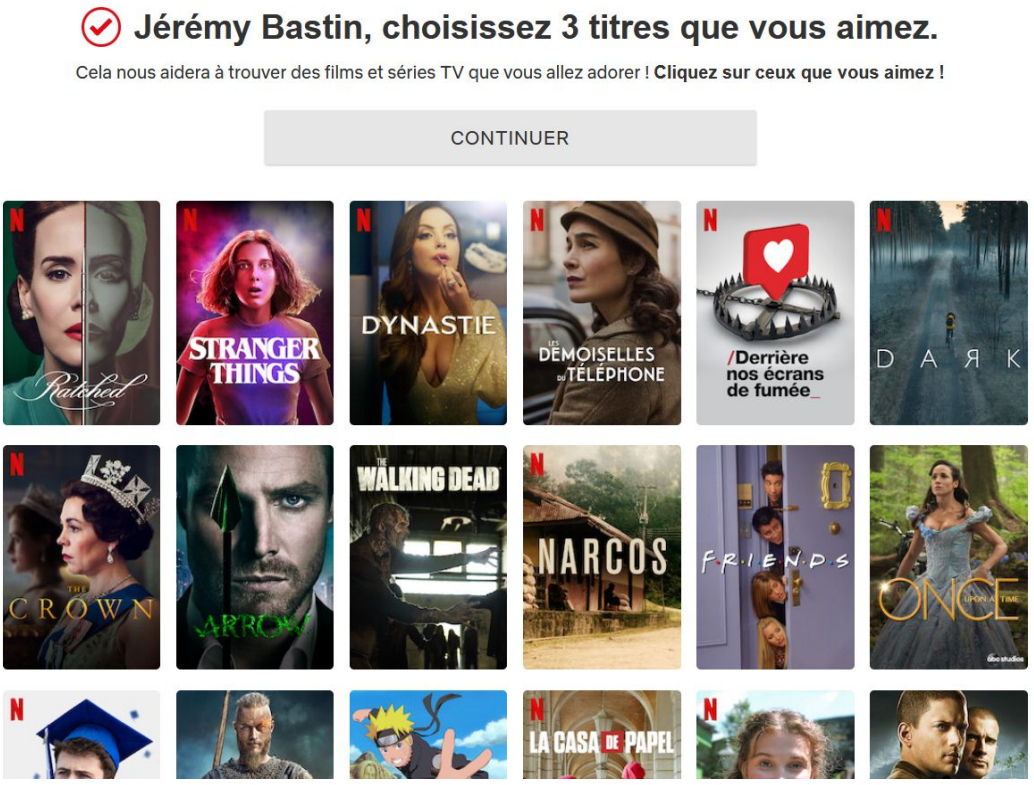

Figure 2. Sélection de préférences de l'utilisateur lors de la création d'un profil sur la plateforme Netflix (screenshot personnel, septembre 2020).

De nombreuses plateformes telles que Netflix, Medium ou Pinterest vous demandent de passer par une étape de sélection de vos intérêts avant de pouvoir utiliser l'application. Cet effort actif demandé à l'utilisateur permet de résoudre en partie le problème du démarrage à froid. Les applications font néanmoins attention à ce genre d'effort qui peut augmenter le taux d'abandon lors du processus d'inscription, ce qui explique pourquoi cette étape vient généralement en dernier lieu.

7.3.2. Démarrage à froid des items

Les entreprises sont confrontées à un défi similaire lorsque de nouveaux articles ou contenus sont ajoutés au catalogue. Des plateformes comme Netflix ou Prime Video contiennent un catalogue qui change “peu” fréquemment (il faut du temps pour créer des films ou des séries). Par conséquent, elles ont moins de difficultés comparées à Airbnb ou Zillow où de nouvelles annonces sont créées chaque jour. A ce moment-là, elles n'ont pas de possibilité d'apparaître car elles n'étaient pas présentes pendant le processus d'entraînement du système de recommandations. Airbnb résout ce problème de la manière suivante : “Pour créer des incorporations pour une nouvelle liste, nous trouvons 3 listes géographiquement les plus proches qui ont des incorporations, et sont du même type de liste et de la même gamme de prix que la nouvelle liste, et calculons leur vecteur moyen.” (*Listing Embeddings in Search Ranking*, Airbnb mars 2018).

8. Processus de création d’un système de recommandations

8.1. Définition des besoins business

La première étape de la construction d’un système de recommandations consiste à définir les objectifs et les paramètres du projet. L'objectif principal de cette phase est de mener une étude de faisabilité, tant d’un point de vue commercial que technique. La collaboration entre les équipes marketing, business et technique y est donc primordiale. Trouver la solution adéquate demande de se poser les bonnes questions afin de comprendre le problème à résoudre. La liste ci-dessous non exhaustive reprend une série de questions communes à tout projet de recommandations :

- ➢ Quel est le but final du projet ? L'idée est-elle de créer un système de recommandations pour augmenter directement les ventes / atteindre une taille de panier moyenne plus élevée / réduire le temps de navigation et faire un achat plus rapidement / réduire la longue traîne de contenu non consommé / améliorer l'engagement des utilisateurs avec le produit ?

- ➢ Une recommandation est-elle vraiment nécessaire ? C'est peut-être une question évidente, mais comme les systèmes peuvent être coûteux à construire et à entretenir, cela vaut la peine de se la poser. L'entreprise peut-elle atteindre son objectif final en favorisant la découverte via un ensemble de contenu statique (comme les choix du personnel / éditeur ou le contenu le plus populaire) ?
- ➢ À quel moment les recommandations se produiront-elles ? Si les recommandations ont un sens à plusieurs endroits (c'est-à-dire sur un écran d'accueil lors de la première visite de l'application ou du site ainsi qu'après l'achat ou la consommation de contenu), le même système sera-t-il utilisé aux deux endroits ou les paramètres et les besoins seront-ils distincts pour chacun ?
- ➢ Quelles sont les données disponibles sur lesquelles baser les recommandations ? Quel pourcentage approximatif d'utilisateurs sont connectés avec leur compte (auquel cas il peut y avoir beaucoup plus de données disponibles) par rapport aux anonymes (ce qui pourrait compliquer les choses pour la construction du système de recommandation) ?
- ➢ Tous les contenus ou produits doivent-ils être traités de la même manière ? Autrement dit, y a-t-il des produits ou des éléments de contenu particuliers que l'équipe commerciale souhaite (ou doit) promouvoir en dehors des recommandations organiques ?
- ➢ Comment segmenter les utilisateurs aux goûts similaires ? En d'autres termes, si vous utilisez le modèle basé sur la similitude des utilisateurs, comment décidez-vous de ce qui rend les utilisateurs similaires ?

8.2. Définition des prérequis

Selon les objectifs business choisis, certains prérequis vont apparaître avant l'implémentation d'un prototype du système de recommandations. Pour fonctionner, la plupart des algorithmes de recommandations ont besoin de connaître un historique d'interactions des utilisateurs. Il se peut que vous vous rendiez compte que vous n'avez pas ce genre d'informations à disposition. A ce moment, l'entreprise peut faire le choix de mettre à jour son système de collecte de données ou d'utiliser uniquement les données qu'elle a à sa disposition. Si l'entreprise ne possède pas un grand nombre d'interactions

ou si la plupart des utilisateurs sont inconnus, vous devrez peut-être vous fier à des sources de données externes ou à des données générales qui ne sont pas explicitement liées aux préférences, telle que la géolocalisation de l'utilisateur.

Fixer quelle est la quantité de données minimums pour faire fonctionner un système de recommandations est assez difficile à établir. Si l'entreprise ne peut avoir la quantité de données minimums pour obtenir des résultats pertinents, plusieurs questions peuvent se poser. N'a-t-elle pas assez de données car le service est nouveau? Il est alors nécessaire de mettre place une stratégie pour parvenir à combler ce manque. Si l'entreprise n'arrive pas à atteindre la quantité minimum requise pour d'autres raisons, la question de trouver une alternative à un système de recommandations peut se poser.

8.3. Prototypage

Au cours de cette phase, un prototype du système de recommandations est mis sur pied afin de confirmer les exigences et les choix de conception. Le prototype permet de tester la faisabilité globale du système, il peut s'agir d'une maquette conceptuelle ou d'un prototype basé sur un code. Cette phase permet de rapidement mettre en évidence de potentiels problèmes ainsi que de comparer les différents résultats obtenus selon les approches algorithmiques. Avant que le développement réel ne commence, il est obligatoire de calculer les risques. Les exigences principales étant discutées à ce stade, il est possible de fournir une estimation plus précise de la solution finale, en tenant compte des risques, de la mise en œuvre logicielle, du réglage des algorithmes et de la maintenance à long ou à court terme.

8.4. Déploiement

Une fois le système de recommandations fonctionnant dans un environnement de développement, il s'agit ensuite de le mettre en production afin de commencer à mesurer l'effet sur les objectifs commerciaux. Cette phase peut être plus ou moins longue en fonction de l'impact du système de recommandations dans son environnement. Afin de minimiser les dangers de cette phase, l'entreprise peut déployer progressivement son système tout en mesurant régulièrement l'évolution des résultats. Le déploiement peut se

faire en ciblant une zone géographique, un segment d'utilisateurs ou un nombre restreint d'utilisateurs.

Pour mesurer l'impact du système de recommandations par rapport à la version initiale, les entreprises peuvent recourir aux tests A/B. Cette technique marketing consiste à proposer deux variantes d'une même application à ses utilisateurs et de déterminer la version qui donne les meilleurs résultats par rapport à un objectif. Grâce à cette technique, il est possible de mesurer les performances réelles du système de recommandations auprès des utilisateurs. Une fois le déploiement réussi, nous avons besoin de nous assurer que le système de recommandations est conçu pour s'adapter et évoluer à l'ajout de nouvelles données. Surveiller régulièrement les performances est l'une des parties les plus importantes du processus car un système de recommandations doit s'adapter correctement aux goûts de l'utilisateur avec les nouvelles données au fil du temps.

8.5. Développement continu par itérations

Si les recommandations sont au cœur de l'entreprise, le déploiement initial ne représente que le début du processus. Pour des entreprises telles que Netflix ou Spotify ayant les recommandations au centre de leur business, chaque amélioration peut représenter un revenu potentiel supplémentaire. Mettre en place une stratégie d'amélioration continue peut donc s'avérer primordial. Ce que cela signifie en termes de coût, c'est que l'élaboration d'un système de recommandations n'est pas un investissement exceptionnel mais bien un poste pour lequel on va devoir consacrer des ressources humaines et financières de manière récurrente.

Le cycle de développement continu peut être schématisé par 5 étapes :

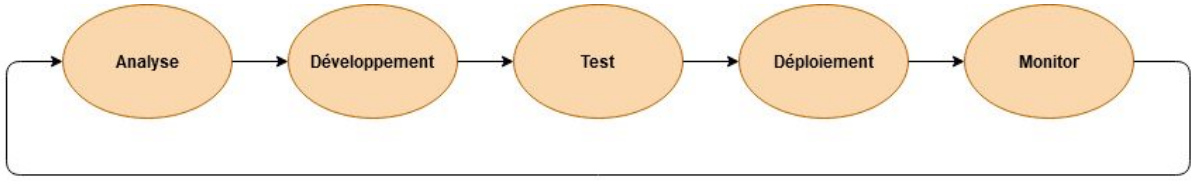

Figure 3. Cycle de développement continu.

Description des étapes :

- ➢ L'analyse permet d'ajuster les objectifs marketings et business selon les rapports établis par l'étape de monitoring. Ces rapports peuvent se présenter sous la forme de statistiques, de tests A/B, de visualisations, etc. C'est également le moment où de nouvelles idées peuvent être présentées et discutées. Tout ceci doit ensuite être transformé en une analyse technique permettant l'étape de développement.
- ➢ Le développement concrétiste les demandes établies au point précédent en code. La programmation des algorithmes n'est pas la seule tâche à réaliser. Les tests, les outils de déploiement et de monitoring font également partie des tâches.
- ➢ Les tests permettent de s'assurer de la qualité du code en vérifiant que le délivrable ne comporte pas de régressions par rapport aux versions précédentes. Ces tests sont généralement exécutés automatiquement avec un rapport en fin de course qui donne des informations sur d'éventuels problèmes.
- ➢ Le déploiement permet de mettre à jour les machines accueillant le système de recommandations selon la stratégie choisie (eg; géolocalisation, segmentation).
- ➢ Le monitoring permet la génération des rapports qui seront utiles pour l'étape d'analyse. Il permet également de surveiller les performances du système de recommandations après son déploiement.

Plus ces étapes seront automatisées, plus il sera possible pour l'entreprise de tester rapidement des versions améliorées de son système de recommandations avec une gestion des coûts maîtrisées.

Le constat est là, le processus de création d'un système de recommandations peut être fort différent selon les besoins. Certaines entreprises pourront trouver satisfaction avec une solution clé en main alors que d'autres devront mettre en place un processus d'amélioration continu afin de rechercher les performances voulues et les maintenir avec le temps. Les principaux prérequis que nous avons pu identifier sont principalement de l'ordre de la collecte des données. Les données sont essentielles pour les systèmes de recommandations, nous en discuterons donc plus en détails. Nous découvrirons également quels algorithmes peuvent convenir à des entreprises n'ayant pas des volumes de données importants.

9. Notions théoriques pour la construction d'un système de recommandations

9.1. Collecte des données

La collecte des données est une étape primordiale dans la construction des systèmes de recommandations car ceux-ci s'appuient principalement sur les données récoltées auprès des utilisateurs. Ces données permettent de construire un profil utilisateur qui sera ensuite utilisé par les algorithmes. A une large échelle, ces données représentent de grosses quantités de volume à gérer qui rentrent souvent dans le domaine du big data. A titre d'exemple, sur une base de 60 millions d'utilisateurs actifs en 2014, Spotify récoltait chaque jour 30 terabytes de données et en générerait 400 terabytes, soit plus de 430 terabytes au total (*The evolution of Big Data at Spotify*, 2014).

Une distinction importante est faite dans la manière dont les données sont récoltées, ce qui influencera ensuite les paramètres des algorithmes. Nous avons d'une part les données explicites et d'autre part les données implicites. La principale différence dans cette distinction réside dans le fait que les données explicites requièrent une action active de l'utilisateur alors que les données implicites n'en requièrent pas. Voyons leurs principales caractéristiques.

9.1.1. Caractéristiques des données explicites

Les données explicites sont comme le nom l'indique des données fournies de manière explicite par l'utilisateur. En voici des exemples : demander à l'utilisateur de classer une collection d'objets en fonction de sa préférence, présenter deux objets à un utilisateur et lui demander de choisir le meilleur, demander à un utilisateur de créer une liste d'articles qui l'intéressent.

Les données explicites requièrent dans tous les cas une action active de la part de l'utilisateur. Ces données sont donc par nature difficiles à récolter. Si l'on prend le cas d'une application de films, il y a de fortes chances pour que certains utilisateurs ne votent jamais pour un seul film, ou ne donnent des notes que de manière succincte. C'est

pourquoi de nombreuses entreprises se tournent vers les données implicites en abandonnant parfois complètement les données explicites. Cette observation de l'abandon des données explicites, ou tout du moins de leur importance s'explique entre autres par la recherche depuis plusieurs années de l'amélioration de l'engagement des utilisateurs plutôt que de miser sur la pertinence. Les données explicites souffrent également de problèmes de scalabilité dû à leur fort taux de sparsité. Les données explicites ont néanmoins l'avantage de pouvoir être facilement interprétées puisque celles-ci reflètent une intention précise de l'utilisateur, e.g. Je mets 5 étoiles à un film car je l'ai aimé, il n'y a donc aucune incertitude.

9.1.2. Caractéristiques des données implicites

Les données implicites sont des données récoltées de manière implicite par l'utilisateur. En voici des exemples : l'historique de navigation, le nombre de clics sur un lien, le nombre de fois qu'une chanson est écoutée, le pourcentage joué d'une vidéo, le pourcentage de scroll d'une page de description d'un produit, le temps que je reste à regarder une photo. Les données implicites ne reflètent pas directement l'intérêt d'un utilisateur mais agissent plutôt comme un proxy pour ses intérêts.

Les données implicites ont pour avantage de pouvoir être récoltées de manière bien plus régulière et nombreuse que les données explicites sans devoir perturber l'utilisateur d'une quelconque manière. De plus, les sources de données implicites peuvent être nombreuses et variées, ce qui explique pourquoi les applications actuelles utilisent les données implicites de manière presque exclusive. Si vous écoutez de la musique sur Spotify, l'application enregistrera le nombre de fois que vous avez écouté une chanson, que vous l'avez sauvé dans une liste, le pourcentage de la piste que vous avez écouté, si vous avez cliqué sur la page d'un artiste en écoutant une chanson, le moment et le lieu de l'écoute, etc. Ce large spectre de données permet de mesurer plus précisément l'intérêt réel d'un utilisateur pour un item.

Les données implicites ne peuvent cependant pas être interprétées de manière directe comme c'est le cas pour les données explicites. Un utilisateur qui regarde un film en entier ne signifie pas nécessairement qu'il l'a apprécié. Il n'y a généralement pas de

moyen de mesurer directement l'intérêt réel d'un utilisateur pour un item. C'est la répétition plus ou moins grande d'actions en faveur d'un item qui nous donnera un certain niveau de confiance sur les intérêts de l'utilisateur. Ainsi, pour une chanson écoutée 50 fois, nous pouvons avoir un taux de confiance élevé sur le fait que ce l'utilisateur aime réellement la chanson.

9.1.3. Différences d'interprétation entre les données explicites et implicites

Dans le cas des données explicites, les données manquantes restent manquantes puisqu'il n'y a aucun moyen de leur trouver une valeur alternative pertinente. En revanche, dans le cas des données implicites, les données manquantes sont prises en compte et sont initialisées à zéro plutôt que d'être ignorée. Une valeur mise à zéro ne signifie pas que l'utilisateur n'apprécie pas l'item, mais simplement que le niveau de confiance est nul puisqu'il n'y a eu aucune interaction. L'absence d'écoute d'une chanson indique par exemple que l'utilisateur n'est peut-être même pas au courant de l'existence de la chanson.

9.2. Traitement des données

Pour qu'un algorithme puisse utiliser des données, celles-ci doivent subir quelques transformations préalables. La liste de ces transformations possibles est large mais nous retrouvons en général le retrait des mots vides de sens suivi d'une transformation en valeurs numériques selon la technique choisie.

9.2.1. Retrait des mots vides

Lorsque l'on traite les données, on commence souvent par retirer les mots considérés comme vide de sens ("stopwords" en anglais). Les mots vides sont tellement communs qu'il est inutile de les indexer dans le vocabulaire du corpus étant donné leur distribution statistique uniforme dans les textes de la collection. Chaque langue a sa liste de mots vides, nous retrouvons par exemple en français les mots : "le", "la", "de", "du", "ce", etc.

9.2.2. Transformation des données en vecteurs

	La	caméra	est	très	bien	mauvaise
La caméra est bien	1	1	1	0	1	0
La mauvaise caméra est très mauvaise	1	1	1	1	0	2
La caméra est très bien	1	1	1	1	1	0

Figure 4. Représentation de valeurs textuelles en valeurs numériques. Chaque case indique le nombre d'occurrences d'un mot dans une phrase. Chaque ligne représente une phrase et chaque colonne représente un mot du vocabulaire du corpus.

Afin de pouvoir être comprises par les algorithmes, les descriptions textuelles des items doivent être transformées en valeurs numériques. La méthode la plus simple consiste à reprendre la fréquence brute des termes et à les placer dans une matrice où chaque ligne représente un item et chaque colonne un mot du vocabulaire du corpus. Cela signifie que si le vocabulaire fait 10.000 mots, il faudra autant de colonnes pour chaque item. Chaque case de cette matrice indique le nombre d'occurrences d'un mot pour un item. Cette simple représentation est suffisante pour que les algorithmes puissent manipuler ces données et effectuer des calculs de similarités, voir d'entraîner des algorithmes d'apprentissage automatique.

9.2.3. Pondération des mots via la méthode TF-IDF

Pour le moment, nous comptons les mots avec la même importance, or un mot n'a pas nécessairement la même pertinence face aux autres. Afin d'ajouter cette nuance, nous pouvons pondérer les mots via la mesure TF-IDF. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document et varie également en fonction de la fréquence du mot dans le corpus. Cette méthode vise à donner un poids plus important aux termes les moins fréquents, considérés comme plus discriminants.

$$w_{ij} = tf_{i,j} * log(\frac{N}{df_i})$$

Figure 5. Formule TF-IDF où i représente un item, j un mot et N le nombre d'items.

- ➢ Le premier terme de la formule représente la fréquence d'apparition d'un terme dans un item. Ceci se calcule par la division du nombre d'apparitions du terme dans l'item par le nombre total des termes dans l'item.
- ➢ Le deuxième terme de la formule représente la fréquence inverse d'un terme calculé par le logarithme de l'inverse de la proportion des items du corpus qui contiennent le terme.

9.3. Calcul de similarité

Avant de pouvoir étudier les algorithmes de recommandations, nous devons savoir comment nous allons calculer la similarité entre les utilisateurs et les items. Plusieurs mesures de similarité ont été proposées dans la littérature. Parmi celles-ci, nous retiendrons deux des plus utilisées, le coefficient de corrélation de Pearson et la similarité cosinus. Il existe d'autres mesures de similarité qui n'ont pas connu d'adoption significative et que par conséquent je n'aborderai pas.

9.3.1. Coefficient de corrélation de Pearson

Le coefficient de Pearson est un indice situé entre -1 et 1, reflétant une relation linéaire entre deux variables continues. S'il s'agit de calculer la similarité entre deux utilisateurs, leur corrélation est mesurée à l'aide de deux vecteurs représentants les interactions antérieures. Dans le cas de données explicites, seuls les items co-évalués sont incorporés dans les deux vecteurs. Un coefficient proche de -1 signifie que les utilisateurs a une similarité inverse et inversement, un coefficient proche de 1 signifie une similarité entière entre les deux utilisateurs. Un coefficient proche de 0 signifie que les utilisateurs partagent une similarité moyenne.

La similarité sim(u, v) entre les utilisateurs u et v est donnée par l'équation suivante:

$$sim(u,v) = \frac{\sum_{i \in I} (r_{ui} - \bar{r}_u) * (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{ui} - \bar{r}_u)^2 * \sum_{i \in I} (r_{vi} * \bar{r}_v)^2}}$$

Figure 6. Similarité par la corrélation Pearson entre deux utilisateurs u et v. I est l'ensemble des items qui ont été co-évalués par les utilisateurs u et v.

La similarité sim(i, j) entre les items i et j est donnée par l'équation suivante:

$$sim(i,j) = \frac{\sum_{u \in U} (r_{ui} - \bar{r}_i) * (r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{ui} - \bar{r}_i)^2 * \sum_{u \in U} (r_{uj} * \bar{r}_j)^2}}$$

Figure 7. Similarité par la corrélation Pearson entre deux items i et j. U est l'ensemble des utilisateurs qui ont co-évalué les items i et j.

9.3.2. Similarité basée sur le cosinus

La similarité cosinus permet de calculer la similarité en déterminant le cosinus de l'angle entre deux vecteurs de même dimension. Cette similarité se situe entre 0 et 1 où le 0 signifie aucune similarité et 1 une similarité parfaite.

La similarité sim(u, v) entre les utilisateurs u et v est donnée par l'équation suivante:

$$sim(u,v) = cos(r_u, r_v) = \frac{\sum_{i \in I} r_{ui} * r_{vi}}{\sqrt{\sum_{i \in I} r_{ui}^2 * \sum_{i \in I} r_{vi}^2}}$$

Figure 8. Similarité cosinus entre deux utilisateurs u et v. I est l'ensemble des items qui ont été co-évalués par les utilisateurs u et v.

La similarité sim(i, j) entre les items i et j est donnée par l'équation suivante:

$$sim(i,j) = cos(r_i, r_j) = \frac{\sum_{u \in U} r_{ui} * r_{uj}}{\sqrt{\sum_{u \in U} r_{ui}^2 * \sum_{u \in U} r_{uj}^2}}$$

Figure 9. Similarité cosinus entre deux items i et j. U est l'ensemble des utilisateurs qui ont co-évalué les items i et j.

9.4. Métriques de précision

Les systèmes de recommandations basés sur le filtrage collaboratif font des prédictions sur la matrice des interactions des utilisateurs. Pour mesurer la précision de celles-ci, les interactions estimées sont comparées avec les interactions réelles, c'est-à-dire celles qui ont été créées par l'utilisateur. La précision d'un système de recommandations est généralement évaluée par deux mesures principales : l'erreur quadratique moyenne (RMSE) et l'erreur absolue moyenne (MAE). Ces deux métriques permettent une interprétation facile de par leur même échelle que les notes d'origine. Cependant, il peut être préférable d'en utiliser une ou l'autre selon le contexte.

MAE

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Figure 10. Erreur moyenne absolue - Mean absolute error

L'erreur moyenne absolue a pour caractéristique qu'elle ne donne aucun biais aux erreurs extrêmes. S'il y a des valeurs aberrantes ou des termes d'erreur importants, leur erreur pèsera de la même manière que les autres erreurs de prédictions. Par conséquent, la métrique MAE doit être privilégiée lorsque l'on recherche davantage l'exactitude des notes plutôt que de donner de l'importance aux valeurs aberrantes. Pour obtenir une vue

ou une représentation holistique du système de recommandation, on utilisera donc la métrique MAE.

RMSE

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\widehat{y}_i - y_i)^2}{n}}$$

Figure 11. Erreur quadratique moyenne - Root mean squared error

L'erreur quadratique moyenne a pour caractéristique principale qu'elle a tendance à pénaliser davantage les erreurs importantes puisqu'elles sont mises au carré. Cela signifie que la métrique RMSE est plus susceptible d'être affecté par des valeurs aberrantes ou de mauvaises prédictions. Par définition, la métrique RMSE ne sera jamais aussi petit que la métrique MAE. De plus, la métrique RSME n'utilise pas de valeurs absolues, ce qui est beaucoup plus pratique mathématiquement lors du calcul de la distance, du gradient ou d'autres mesures. C’est pourquoi la plupart des fonctions de coût de l’apprentissage automatique évitent d’utiliser la métrique MAE.

10. Etude des algorithmes de recommandations

10.1. Les différentes approches

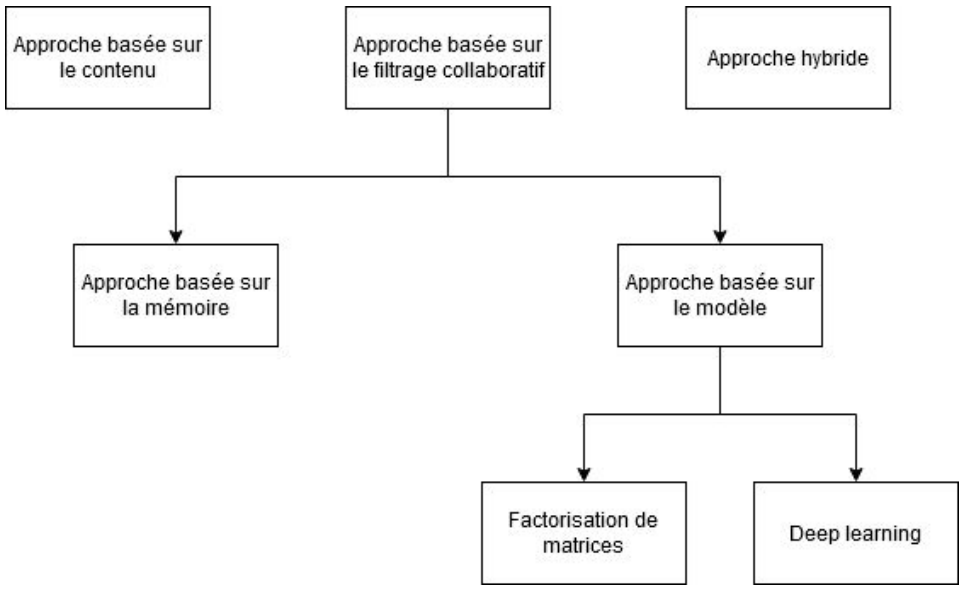

Figure 12. Les différentes approches des systèmes de recommandations.

L'étude des algorithmes de recommandations a apporté des centaines d'algorithmes qui peuvent être classés sur base de leurs approches. C'est sur cette base hiérarchique que nous allons explorer les algorithmes les plus connus. Seul la section du deep learning sera étudiée séparément étant donné l'importance prise par ce domaine.

Dans un premier temps, nous allons explorer le filtrage basé sur le contenu qui présente les algorithmes les plus simples. Nous étudierons ensuite l'approche basée sur le filtrage collaboratif qui représente la partie la plus importante de part ses performances et sa large adoption. Enfin, j'expliquerai brièvement l'approche hybride qui consiste à mélanger les recommandations des deux approches précédentes dans le but de pallier à certains défauts.

10.2. Approche basée sur le contenu

Le filtrage basé sur le contenu est une approche qui exploite les caractéristiques textuelles des items afin de trouver d'autres items similaires. Cette méthode permet l'analyse des

préférences du consommateur vis-à-vis des qualités ou des propriétés d'un produit. Préférences qui peuvent être déterminées par le comportement d'achat passé du client ou de manière isolée pour chaque item.

Pour trouver des recommandations, les items sont comparés par paire dans une matrice d'item-item. Les caractéristiques textuelles d'un item sont représentées dans une longue chaîne de caractères que l'on appelle généralement "sac de mots" (ou "bag of words" en anglais). Pour un film, on peut reprendre par exemple le titre, la description, les mots-clés, les genres, les noms des acteurs, le réalisateur, ainsi que tout élément pertinent pour la comparaison. Ce sac de mots est ensuite transformé en une représentation vectorielle ou par la méthode TF-IDF. On calcule ensuite pour chaque paire item-item leur similarité sur base de ces vecteurs. Cette similarité peut être calculée grâce aux formules telles que le coefficient de corrélation Pearson ou la similarité cosinus.

	0	1	2	3	...	16666	16667	16668	16669
0	1.0000	0.0357	0.0391	0.0000	...	0.0208	0.0000	0.0000	0.0000
1	0.0357	1.0000	0.0609	0.0630	...	0.0215	0.0000	0.0000	0.0330
2	0.0391	0.0609	1.0000	0.0000	...	0.0236	0.0000	0.0000	0.0000
...	...	...	...	...	...	...	...	...	...
16667	0.0000	0.0000	0.0000	0.0354	...	0.0725	1.0000	0.1913	0.0000
16668	0.0000	0.0000	0.0000	0.0000	...	0.0000	0.1913	1.0000	0.0000
16669	0.0000	0.0330	0.0000	0.0374	...	0.0383	0.0000	0.0000	1.0000

16670 rows × 16670 columns

Figure 13. Matrice de similarités calculée sur 16670 items. Chaque case représente la similarité cosinus calculée d'une paire item-item. La diagonale représente les items comparés avec eux-mêmes, d'où une similarité parfaite de 1.

Une fois la matrice de similarité M calculée, on peut connaître la similarité d'un item i par rapport à un autre item k en accédant à la valeur en M_{ij}. Pour avoir nos recommandations, il nous suffit de sélectionner les similarités les plus grandes liées à un item en particulier.

L’approche basée sur le contenu a pour avantage d’être une méthode relativement simple à mettre en place et qui ne nécessite pas d’interactions utilisateurs préalables. Cependant, construire des recommandations simplement sur base de comparaisons textuelles ne permet pas de faire des recommandations très excitantes pour l’utilisateur. Ce genre d’approche peut donc convenir pour des petites plateformes n’ayant pas ou peu d’interactions utilisateurs ou pour les contenus dont la popularité ne joue pas un rôle majeur.

10.3. Approche basée sur le filtrage collaboratif

Le filtrage collaboratif est une approche basée sur le partage d’opinions entre les utilisateurs. Il reprend le principe du "bouche à oreille" pratiqué depuis toujours par les humains pour se construire une opinion sur un produit ou un service qu’ils ne connaissent pas. L’hypothèse fondamentale de cette méthode est que les opinions des autres utilisateurs peuvent être utilisées pour fournir une prédiction raisonnable de la préférence de l’utilisateur sur un item qu’il n’a pas encore noté. Ces méthodes supposent que si des utilisateurs ont les mêmes préférences sur un ensemble d’items, alors ils auront probablement les mêmes préférences sur un autre ensemble d’items qu’ils n’ont pas encore noté. On distingue deux catégories de méthodes de filtrage collaboratif : les méthodes basées sur la mémoire et les méthodes basées sur un modèle qui sont plus élaborées.

10.3.1. Approche basée sur la mémoire

L'approche basée sur la mémoire utilise les interactions passées des utilisateurs pour calculer les similitudes entre ceux-ci ou entre les items. Pour trouver la note r qu'un utilisateur u donnerait à un item i, l'approche recherche les utilisateurs similaires à u qui ont noté l’item i et calcule la note r en fonction des notes des utilisateurs trouvés à l'étape précédente. Afin de trouver les U utilisateurs les plus similaires à l’utilisateur u, on calcule la similarité sur base des items communément notés avec l’utilisateur comparé en calculant leur distance ou leur similarité.

NB : Notez que deux utilisateurs A et B peuvent être considérés comme absolument similaires dans la métrique de similarité cosinus malgré des évaluations différentes. Un exemple serait un utilisateur critique de cinéma qui attribue toujours des notes inférieures à la moyenne, mais dont le classement des éléments de sa liste serait similaire à celui des évaluateurs moyens comme B. Pour tenir compte de ces préférences individuelles des utilisateurs, il faut amener tous les utilisateurs au même niveau en supprimant leurs préjugés. Ceci peut se faire en soustrayant la note moyenne donnée par cet utilisateur à tous les items de chaque item noté par cet utilisateur.

Après avoir déterminé une liste d'utilisateurs similaire à un utilisateur u, on calcule la note r que u donnerait à un certain item i.

On considère que la note r d'un utilisateur pour un item i sera proche de la moyenne des notes attribuées à i par les U utilisateurs les plus similaires à u. La formule mathématique de la note moyenne donnée par U utilisateurs se calcule avec formule dont la version la plus simple est :

$$\widehat{r}_{ui} = \frac{1}{n} * \sum_{u' \in U}^{n} r_{u',i}$$

Figure 14. r représente la note donnée par un utilisateur u à un item i et U représente le groupe d'utilisateurs similaires à u.

Il est également possible de multiplier la note par le degré de similarité entre les deux utilisateurs afin de donner plus de poids aux notes d'utilisateurs fort similaires à u :

$$\widehat{r}_{ui} = k * \sum_{u' \in U}^{n} sim(u, u') * r_{u',i}$$

Figure 15. k est un facteur de normalisation.

Enfin, on peut également prendre en compte les notes moyennes de l'utilisateur u dans le calcul étant donné que les utilisateurs peuvent avoir tendance à voter différemment les uns des autres :

$$\widehat{r}_{ui} = \overline{r}_u + k * \sum_{u' \in U}^{n} sim(u, u') * (r_{u',i} - \overline{r}_{u'})$$

Figure 16. $\overline{r_u}$ est la moyenne des notes de l'utilisateur u pour tous les items notés par u.

Différence entre l'approche basée sur les utilisateurs et les items

Il existe deux approches pouvant être utilisées pour trouver des recommandations avec le filtrage collaboratif. Ces deux approches sont mathématiquement assez similaires, mais il existe une différence conceptuelle entre les deux. Voici comment elles se comparent :

- ➢ Basé sur les utilisateurs : pour un utilisateur u, le vote pour un item i, qui n'a pas encore été voté par l'utilisateur u est trouvé en prenant les U utilisateurs les plus similaires qui ont noté l'item i et en calculant le vote basé sur les votes de ces U utilisateurs.
- ➢ Basé sur les items : pour un item i, le vote par un utilisateur u, qui ne l'a pas encore voté est trouvé en prenant les I items les plus similaires qui ont été noté par l'utilisateur u en calculant le vote basé sur les votes de ces I items

Dans un système où il y a plus d'utilisateurs que d'items, le filtrage basé sur les items est plus rapide et plus stable que celui basé sur les utilisateurs. Il est efficace car en général, la note moyenne reçue par un item ne change pas aussi rapidement que la note moyenne attribuée par un utilisateur à différents éléments. Il est également connu pour être plus performant que l'approche basée sur les utilisateurs lorsque la matrice de notation est fortement éparse.

10.3.2. Approche basée sur le modèle

Méthode des k plus proches voisins

Les 'k plus proches voisins' est une méthode dans laquelle le modèle mémorise les observations de l'ensemble d'apprentissage pour la classification des données de l'ensemble de test. Pour prédire la classe d'une nouvelle donnée d'entrée, le modèle va chercher ses k voisins les plus proches en utilisant la distance euclidienne, ou une mesure de similarité et choisira la classe des voisins majoritaires.

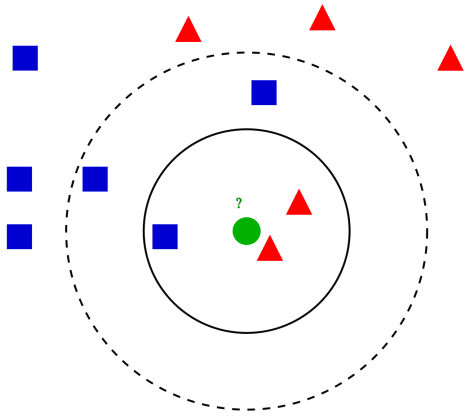

Figure 17. Exemple de classification k-NN. L'échantillon de test (cercle vert) pourrait être classé soit dans la première classe de carré bleu ou la seconde classe de triangles rouges. Si k = 3 (cercle en ligne pleine) il est affecté à la seconde classe car il y a deux triangles et seulement un carré dans le cercle considéré. Si k = 5 (cercle en ligne pointillée) il est affecté à la première classe (3 carrés face à deux triangles dans le cercle externe).

Sur base d'un nombre k fixé par l'utilisateur qui représentera le nombre de voisins à utiliser lors de l'apprentissage, l'application de cette méthode suit les étapes suivantes :

- ➢ On calcule les distances entre notre donnée et les autres via une mesure telle que la distance euclidienne ou la similarité cosinus.
- ➢ On sélectionne les k voisins les plus proches avec les distances les plus petites ou les similarités les plus grandes.
- ➢ Parmi les k voisins sélectionnés, on sélectionne la classe majoritairement représentée.

Pour choisir le nombre k optimal, l'heuristique du coude permet de déterminer le nombre de clusters optimal dans un ensemble de données. La méthode consiste à tracer la variation expliquée en fonction du nombre de clusters, et à choisir le coude de la courbe comme le nombre de clusters à utiliser.

La factorisation de matrices

La factorisation de matrices, ou décomposition de matrices est une méthode qui permet d'accélérer la recherche de recommandations. L'idée derrière la factorisation matricielle est de représenter les utilisateurs et les items dans un espace latent de dimension inférieure à celui de base en décomposant la matrice initiale en plusieurs autres matrices. Cette réduction de dimensionnalité permet de faire face aux problèmes de scalabilité du traitement de ces matrices qui peuvent se révéler volumineuses et très éparses. Pour retrouver la matrice originale, il suffira de faire le produit de ces matrices entre elles. Cette décomposition d'une matrice creuse en deux matrices denses de dimensions inférieures nous permet d'économiser du stockage et d'accélérer les calculs. Ces avantages en ont fait une méthode très utilisée dans le domaine du filtrage collaboratif.

Le modèle des facteurs latents

Le modèle des facteurs latents représente les items et les utilisateurs par des vecteurs de caractéristiques de même taille, les facteurs latents. Plus la correspondance entre les facteurs latents d'un utilisateur et d'un item est grande, plus il y a de chance pour que le film corresponde aux goûts de l'utilisateur. Bien que ces facteurs latents représente des caractéristiques, il ne faut pas faire l'erreur de vouloir mettre une étiquette dessus car nous ne pouvons faire que des suppositions sur leurs significations.

L'objectif principal des facteurs latents est d'approximer au mieux la matrice des relations user-item. L'estimation de cette matrice $\widehat{R}$ est égale à la multiplication de la matrice des facteurs latents des utilisateurs P par la matrice transposée des facteurs latents des items Q.

$$R \simeq P * Q^T = \widehat{R}$$

Figure 18. Approximation de la matrice R par la multiplication de la matrice P et Q transposée.

Pour trouver la correspondance entre un utilisateur et un film, on multiplie leurs facteurs latents :

$$\widehat{R}_{ui} = p_u^T * q_i = \sum_{k=1}^{k} p_{uk} * q_{ki}$$

Figure 19. Approximation d'une relation user-item par la multiplication des facteurs latents de l'utilisateur u et de l'item i. k représente le nombre de facteurs latents.

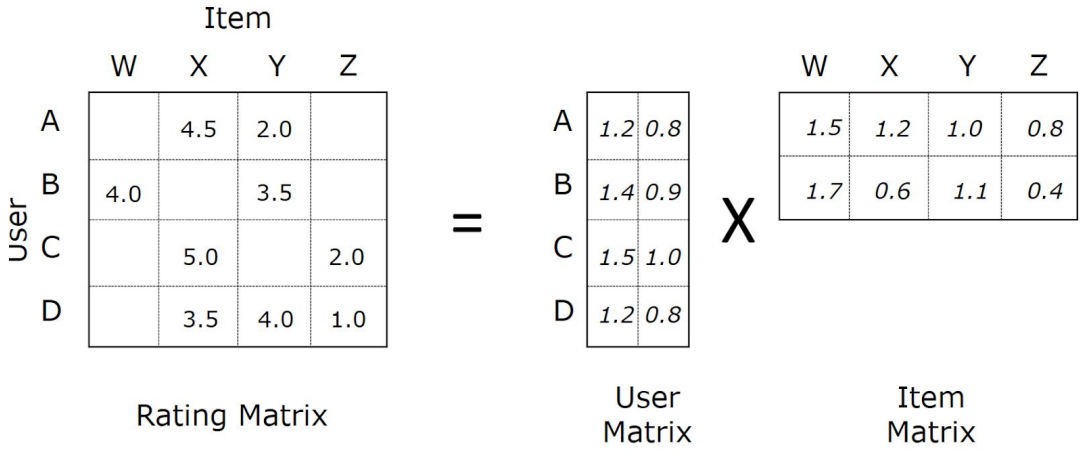

Figure 20. Représentation de la transformation d'une matrice creuse en matrices denses à 2 facteurs latents.

Méthodes d'apprentissage

Afin de trouver les facteurs latents qui estiment le mieux la matrice $\widehat{R}$, les matrices Q et P sont d'abord initialisées avec des valeurs aléatoires. Ensuite, on minimise l'erreur quadratique entre la relation estimée et la réelle grâce la formule :

$$min_{q_i, p_u} \sum_{u,\, i \in k} (r_{ui} - \widehat{r}_{ui})^2 + \lambda(|| q_i ||^2 + ||p_u||^2)$$

Figure 21. Formule de la minimisation de la différence entre la relation estimée et celle contenue dans la matrice originelle. Le deuxième terme est une régularisation qui permet de ne pas surestimer les données observées.

Afin de minimiser cette erreur, nous devons passer par un algorithme d'apprentissage automatique qui va progressivement diminuer l'erreur des estimations. Les deux techniques les plus utilisées sont l'apprentissage par descente de gradient stochastique et l'apprentissage par les moindres carrés alternés.

10.4. Approche hybride

La plupart des systèmes de recommandations utilise une approche hybride combinant le filtrage collaboratif et le filtrage basé sur le contenu. Il n'y a aucune raison pour que plusieurs techniques différentes du même type ne puissent pas être hybridées. Plusieurs études comparant empiriquement les performances de l'hybride aux méthodes purement collaboratives et basées sur le contenu ont démontré que les méthodes hybrides peuvent fournir des recommandations plus précises que les approches pures. Ces méthodes peuvent également être utilisées pour surmonter certains des problèmes tels que le démarrage à froid. Ces techniques d'hybridation comprennent les techniques :

- ➢ Pondéré : combinaison numérique du score de différents éléments de recommandation.
- ➢ Commutation : choisir parmi les composants de recommandation et appliquer celui sélectionné.
- ➢ Mixte : les recommandations de différentes recommandations sont présentées ensemble pour donner la recommandation.

11. Systèmes de recommandations sensibles au contexte

Nos envies diffèrent selon les conditions qui nous entourent, ce qui finit par affecter nos choix et préférences. Nous choisissons des musiques plus calmes lorsque nous voulons

nous détendre et des sons énergiques lorsque nous avons besoin de nous motiver. Se référer simplement à notre historique d'écoute dans des cas comme celui-là ne générera pas les meilleures recommandations auxquelles on pourrait s'attendre car nous aurons probablement un mélange des deux types de chansons. Pour faire face à ce genre de problème, des systèmes de recommandations ont été rendu sensibles au contexte, c'est-à-dire qu'ils utilisent toutes les informations de l'environnement externe disponibles pour améliorer l'exactitude des recommandations. Il existe trois façons dont les informations contextuelles peuvent être utilisées pour améliorer les recommandations:

- ➢ Pré-filtrage contextuel: avant de générer les recommandations, les éléments qui ne correspondent pas au contexte sont supprimés. Par exemple, la musique forte pendant une journée de travail.
- ➢ Post-filtrage contextuel: Après avoir généré les recommandations, les informations contextuelles sont utilisées pour filtrer ou réorganiser les éléments produits, cela peut être appliqué dans le même exemple que pour le pré-filtrage, mais en filtrant à la fin.
- ➢ Modélisation contextuelle: le contexte est inclus dans le modèle qui prédit les notes des items.

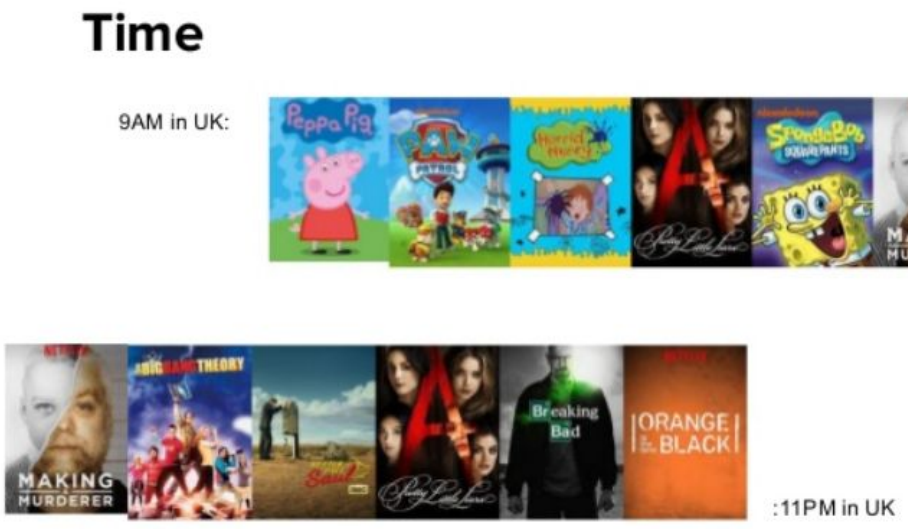

Figure 22. Influence de l'heure sur les recommandations proposées par Netflix (Context aware. Recommandations at Netflix, Mai 2018).

Netflix a présenté leurs avancées sur leur utilisation du contexte (*Context aware. Recommandations at Netflix,* Mai 2018). La figure ci-dessus montre l'adaptation des recommandations en fonction de l'heure. Chez Netflix, l'heure n'est qu'un élément contextuel pris parmi bien d'autres, eg ; météo, événements culturels et politiques, appareil de visionnage utilisé, etc. Netflix est également capable d'inférer des données telles que si vous êtes en compagnie d'un ami(e) ou non.

12. Systèmes de recommandations basées sur le deep learning

12.1. Introduction au deep learning

Le deep learning ou apprentissage profond regroupe les méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des transformations non linéaires. Ses perspectives visent au remplacement de certains algorithmes construit par les humains, encore relativement laborieux. Certaines de ces méthodes s'inspirent des dernières avancées en neuroscience en termes d'interprétation du traitement de l'information et des modèles de communication du système nerveux. Celles-ci sont représentées par des réseaux composés de plusieurs dizaines, voire milliers de neurones artificiels avec de nombreuses interconnexions. Ces réseaux complexes ne peuvent généralement pas expliquer eux-mêmes leur façon de « penser ». Les calculs aboutissant à un résultat ne sont pas visibles pour les programmeurs qui ont créé le réseau neuronal. Lorsque ces réseaux sont composés de nombreuses couches de neurones denses, on parle de réseaux profonds (ou "deep learning" en anglais), ce qui donne à ces réseaux la capacité d'apprendre des caractéristiques plus complexes, mais sont également plus difficile à entraîner.

Le deep learning a récemment apporté des succès majeurs dans plusieurs domaines tels que la vision par ordinateur, la compréhension de la parole ou la traduction automatique. La capacité à intégrer des sources multiples d'information et à tirer profit d'importants volumes de données a permis au domaine des systèmes de recommandations d'offrir des recommandations de plus en plus personnalisées.

12.2. Etude du système de Youtube

Pour comprendre comment le deep learning est utilisé dans les systèmes de recommandations, nous allons étudier le papier blanc publié en 2016 par Youtube expliquant leur approche du problème (*Deep Neural Networks for YouTube Recommendations*). Recommander du nouveau contenu pour Youtube n'est pas une tâche simple, le site est le deuxième site web le plus visité aux États-Unis, avec plus de 500 heures de contenu téléchargé par minute (*Statista*, mai 2019). Youtube a beaucoup fait évoluer son algorithme depuis lors, mais ce papier reste un exemple concret de la manière dont on peut utiliser le deep learning par rapport aux techniques que nous avons utilisées jusqu'à présent.

L'article explique une approche de recherche de recommandations en deux étapes dans laquelle un premier réseau génère des candidats et un deuxième réseau classe ces candidats générés. Cette approche est assez réfléchie étant donné que la recommandation de vidéos peut être posée comme un problème extrême de classification multiclasse. Disposer d'un réseau pour réduire la cardinalité de la tâche de quelques millions de vidéos à quelques centaines de vidéos permet au réseau de classement de tirer parti de fonctionnalités plus sophistiquées que le modèle de génération de candidats ne pourrait apprendre. Ainsi, l'approche du deep learning chez YouTube pour son système de recommandations repose sur deux facteurs principaux :

- ➢ Scalabilité : en raison de l'immense sparsité des matrices, il est difficile pour les approches de factorisations matricielles précédemment vues de se mettre à l'échelle.
- ➢ Cohérence : de nombreuses autres équipes basées sur les produits chez Google sont passées sur le deep learning comme cadre général pour les problèmes d'apprentissage. Google, et donc Youtube, ont tendance à désormais utiliser le deep learning pour toutes sortes de problèmes, donnant la plupart du temps de meilleurs résultats.

12.2.1. Architecture

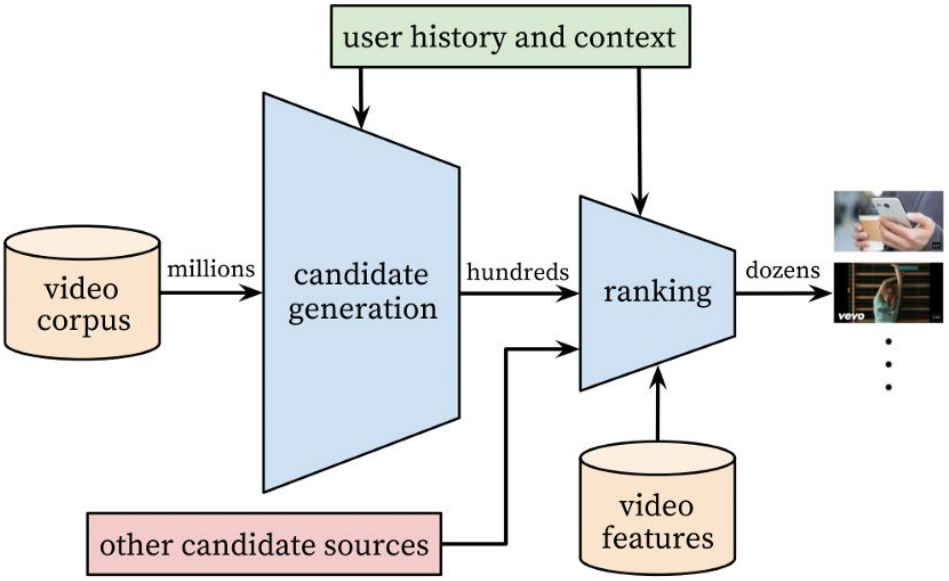

Figure 23. Architecture haut niveau du système de recommandations de Youtube (*Deep Neural Networks for YouTube Recommendations*).

Dans l'architecture représentée, deux réseaux sont en jeu :

- ➢ Le premier réseau traversé par le corpus vidéo est l'étape de génération des candidats. Les algorithmes derrière cette étape permettent de repérer les paradigmes entre les requêtes recherchées et les vidéos candidates. Le réseau de génération de candidats prend en compte l'énorme corpus contenant des millions de vidéos en entrée, le journal d'activité de l'utilisateur et distribue quelques centaines de vidéos en sortie qui sont considérées pour les recommandations de l'utilisateur.
- ➢ Le deuxième réseau que les éléments récupérés traversent après le réseau de génération de candidats permet le classement (ou "ranking" en anglais). Dans ce réseau, un ensemble complet de fonctionnalités vidéo est pris en compte ainsi que le contexte et l'historique de l'utilisateur pour attribuer en sortie un score aux vidéos. Le réseau vise à détecter des spécificités plus complexes des vidéos.

Une fois le réseau formé de bout en bout et entraîné, il reçoit l'historique de temps d'un utilisateur jusqu'à un certain temps t, et on demande au réseau ce qu'il aimerait regarder au temps $t+1$. Les auteurs pensent que prédire les vidéos que l'utilisateur va regarder à l'instant $t+1$ est l'une des meilleures façons de recommander des vidéos compte tenu de la nature épisodique des vidéos sur YouTube.

12.2.2. Détails sur la génération de candidats

Le problème de génération de candidats est un problème extrême de classification multiclasse, où le problème de prédiction devient "classer avec précision un certain temps de visionnage $w(t)$ à un moment t", pour un élément donné i parmi un corpus de vidéos V en tenant compte du contexte C et de l'utilisateur U. La formalisation de l'objectif est décrite par la formule :

$$P(w_t = i \mid U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

Figure 24. Probabilité de la distribution du temps de visionnage.

12.2.3. Détails sur le classement des prédictions

L'idée fondamentale derrière la partition du système de recommandations en deux réseaux est de donner au réseau de classement la possibilité d'examiner chaque vidéo avec plus de subtilité que le modèle de génération des candidats. L'objectif du classement du réseau est de maximiser le temps de visionnage attendu pour une recommandation donnée.

12.2.4. Impact du deep learning chez Youtube

Les auteurs ont démontré l'impact d'un réseau neuronal plus large et plus profond sur la perte par utilisateur. La perte par utilisateur correspond à la quantité totale de temps de visionnage mal estimée, par rapport au temps de visionnage total sur les données sorties. Plus les réseaux sont larges et profonds, plus ils ont tendance à réduire considérablement

les pertes. Ces réseaux plus complexes sont également plus difficiles à entraîner, d'où la recherche d'un bon équilibre à trouver entre performance et coût.

12.3. Word embedding

Le word embedding est une méthode d'apprentissage de représentation de mots qui constitue une avancée majeure dans le traitement de texte. Cette méthode surpasse la méthode TF-IDF précédemment vue à bien des égards. Les word embeddings permettent de représenter chaque mot d'un dictionnaire par un vecteur de nombres réels de taille fixe et de dimension réduite. Cette représentation a ceci de particulier que les mots apparaissant dans des contextes similaires possèdent des vecteurs qui sont relativement proches. On pourrait par exemple s'attendre à ce que les mots « chien » et « chat » soient représentés par des vecteurs relativement peu distants dans l'espace vectoriel où sont définis ces vecteurs. Cette technique est basée sur l'hypothèse qui veut que les mots apparaissant dans des contextes similaires aient des significations apparentées. La technique des words embeddings diminue également la dimension de la représentation des mots, facilitant leur usage comme données d'entrée pour les algorithmes d'apprentissage automatique qui étaient auparavant soumis au fléau des grandes dimensions.

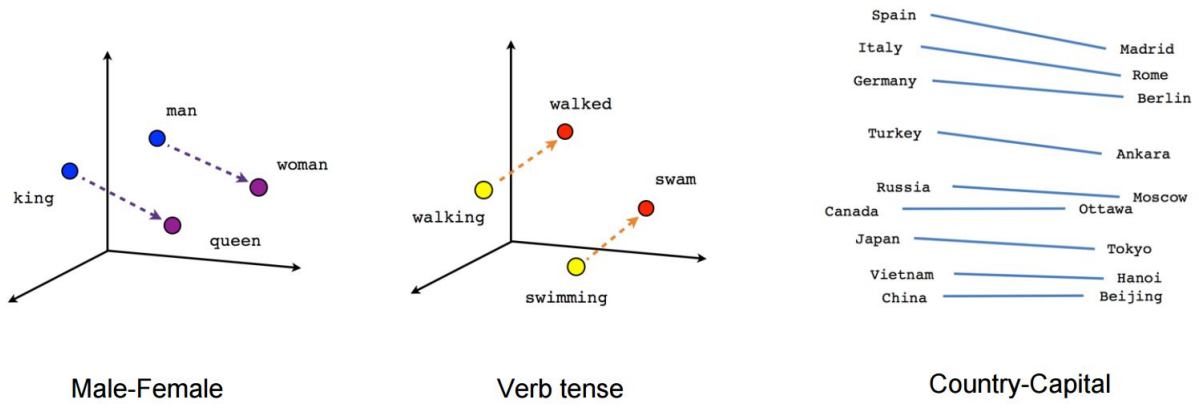

Figure 25. Les word embeddings ont des caractèrsitiques qui leur permettent par exemple de faire des translations linéaires telles que: vecteur('roi') + vecteur('homme') - vecteur('femme') = vecteur('reine').

La plupart des systèmes de recommandations utilisent désormais les word embeddings pour obtenir des recommandations de meilleures qualités. Les embeddings possèdent de nombreux avantages qui en ont fait une représentation très utilisée de manière générale dans le domaine de l'analyse textuelle, en permettant par exemple le développement d'outils de traduction performants ou l'analyse de sentiments.

12.4. Usage avancé du deep learning

12.4.1. Etude du cas Spotify

Spotify est un exemple représentatif de la tendance des applications actuelles à utiliser le deep learning partout où elles le peuvent. Pour améliorer la pertinence des chansons recommandées, Spotify a réussi à combiner l'utilisation de diverses techniques plutôt originales reposant sur le deep learning. Outre le filtrage collaboratif qui reste un pilier important de leur système de recommandations, Spotify utilise le deep learning pour faire du traitement du langage naturel et du traitement de pistes audios qui viennent donner davantage de richesse aux recommandations (*From Idea to Execution: Spotify's Discover Weekly,* Novembre 2015) . Explorons plus en détails ces techniques.

11.4.1. Traitement du langage naturel

Spotify explore le Web à la recherche d'articles de blog et de textes écrits sur la musique pour comprendre ce que les gens disent à propos des chansons et des artistes. Il détermine quels adjectifs et langues sont utilisés pour les décrire et quels artistes et chansons sont discutés. Spotify analyse ensuite les principaux termes qui décrivent une chanson ou un artiste en particulier. Chaque artiste et chaque chanson peuvent avoir des milliers de termes les décrivant. Ces termes sont ajoutés au modèle de chaque chanson et artiste, qui est ensuite utilisé par des réseaux de neurones pour modéliser les chansons à recommander à un utilisateur.

11.4.2. Traitement des pistes audios

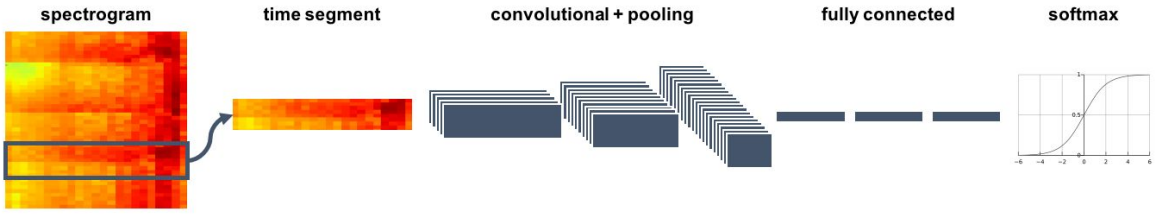

Figure 26. Analyse d'une piste audio par un réseau neuronal.

Spotify analyse également l'audio brut des chansons, c'est-à-dire les signaux sonores envoyés par une chanson. Pour analyser cette audio brut, la piste passe par le même type de réseau neuronal qui analyse les images, appelé réseaux de neurones convolutionnels. Ce réseau traite l'audio brut et produit des caractéristiques telles que la signature rythmique, la tonalité, le mode, le tempo et le volume. Après avoir été traitées par le réseau, les chansons similaires sont classées ensemble.

13. Impacts sur les comportements sociaux et culturels

Une étude menée en 2018 par le Conseil supérieur de l'audiovisuel a étudié la diversité des informations proposées par les algorithmes de Youtube (*Capacité à informer des algorithmes de recommandations. Expérience sur le service Youtube*, novembre 2018). L'étude visait en outre à comprendre les algorithmes sélectionnant les vidéos qui se lancent en lecture automatique afin de déterminer dans quelle mesure la plateforme est un espace d'information pluraliste. Pour mener son étude, le CSA a demandé à 39 membres volontaires, ainsi qu'à 4 nouveaux profils créés pour l'occasion, de regarder des vidéos en utilisant leur compte Google. Ces contenus tournaient autour de 23 sujets identifiés comme « présentant un clivage d'opinion marqué », tels que le véganisme, la laïcité, ou encore la drague de rue. Au total, l'organisme a ainsi pu étudier un peu plus de 39 000 recommandations.

Titre	Occurrences
Téléphones portables et très jeunes enfants - hi-tech	57
« Grand remplacement » : complot, fantasme ou réalité	57
L'homme a t-il réellement marché sur la Lune ?	56
Faut-il interdire l'usage du téléphone portable dans les (...)	53
Complément d'enquête. Les millionnaires du bitcoin (...)	50
Comment sécuriser vos Bitcoins ?	45
SNCF : D'où vient (vraiment) la dette ?	44
Galères de train : qui veut tuer le rail français ?	43
Michael Jackson en vie (retour le 07/07/2016)	42
Peut-on donner la parole à un théoricien du « grand (...)	42
Martin Weill rencontre le rédacteur en chef de Valeurs (...)	41
Le 3 minutes - Laïcité à l'école' #flashtalk	40
Marion Sigaut - Le programme LGBT de Macron	40
Mort de Michael Jackson, les rumeurs les plus folles	40

Figure 27. Liste des vidéos les plus recommandées sur base de 39007 recommandations (Capacité à informer des algorithmes de recommandations. Expérience sur le service Youtube, novembre 2019). Les vidéos à caractère complotiste sont coloriées en orange.

Les conclusions de l'expérience du CSA ont permis de diviser les recommandations en deux parties. Tout d'abord, la sélection des deux premières vidéos présentées automatiquement semblait s'effectuer en accordant une importance particulière « aux mots-clés associés au thème de départ », ainsi qu'à l'engagement des communautés. Le nombre de vues et la date de publication paraissent jouer un rôle moins crucial. Il est à noter que plus d'un tiers des vidéos recommandées « expriment le même point de vue que la vidéo de départ », ce qui semble conduire l'utilisateur dans une bulle de filtres avec la répétition d'une seule et même idée, au détriment d'autres points de vue, confortant ainsi l'utilisateur dans ses certitudes. En revanche, il y aurait une bascule à partir du troisième contenu recommandé. Les outils auraient alors tendance à s'éloigner du sujet de départ, privilégiant des vidéos majoritairement récentes et avec un grand nombre de vues.

13.1. Isolement intellectuel et désinformation

La bulle de filtres est un phénomène qui tend à ne recommander à un utilisateur que des objets en rapport avec les centres d'intérêts que le système lui connaît (*Bulle de filtres*, Wikipédia). Cet effet se renforce de lui-même avec le temps. Un amateur de films d'horreur aura peu de chances de se voir proposer des comédies romantiques ou, dans un autre contexte, un partisan d'Hillary Clinton à se voir exposer à des articles en faveur de Donald Trump. Des requêtes similaires peuvent également donner des résultats très différents selon les utilisateurs. Supposons par exemple que deux personnes, une plutôt politiquement à droite et l'autre plutôt à gauche, recherchent le terme « BP ». Les utilisateurs « de droite » trouveront des informations sur les investissements dans la British Petroleum. Les utilisateurs « de gauche » obtiendront des informations sur la marée noire dans le golfe du Mexique. Ce phénomène tendrait à s'auto-entretenir en reproduisant majoritairement les opinions, croyances et perspectives de l'utilisateur en formant un cercle vicieux. La bulle de filtre réduirait par conséquent le champ informationnel de l'internaute, notamment pour les utilisateurs consultant la presse exclusivement en ligne ou principalement en suivant des liens postés par leurs amis sur les réseaux sociaux, c'est-à-dire sans consulter la « une » ou le sommaire des journaux.

L'idée que le développement des réseaux sociaux aurait entraîné un rapprochement des gens autour des opinions qu'ils partagent, et surtout un éloignement de ceux qui ne les partagent pas, favorisent de plus en plus les extrêmes. C'est ainsi que des situations surviennent comme celle de Susanna Lazarus, une londonienne de 27 ans, persuadée que le Royaume-Uni allait rester dans l'union européenne, car même si elle connaissait des gens du parti "Leave" (quitter l'Union Européenne), elle ne voyait que des gens pour le "Stay" dans son fil d'actualité Facebook. En découvrant cela, elle s'est sentie trompée par les réseaux sociaux (*Bulles de filtre et démocratie,* Les Mondes Numériques). Cette histoire est révélatrice du constat qu'une part importante des utilisateurs est probablement sans indices sur le phénomène de bulle dans laquelle ils s'enferment. Donner les armes nécessaires aux utilisateurs afin de prendre conscience de ces bulles et de s'en détourner doit passer par une prise de conscience et un enseignement de ce phénomène dès le plus jeune âge.

13.2. Du mentor au coach

Tous les algorithmes utilisés par les leaders technologiques se disputent chaque jour un peu plus notre attention. Chacun cherchant à maximiser le temps passé sur sa plateforme. Ce que cela signifie, c'est que nous contrôlons de moins en moins nos désirs. Regardons YouTube ; nous avons tous des objectifs en venant sur le site. Nous pourrions vouloir écouter de la musique, regarder quelque chose de drôle ou apprendre quelque chose de nouveau. Mais tout le contenu qui nous est recommandé, que ce soit par le biais des recommandations de la page d'accueil, du classement de recherche ou de la lecture automatique est optimisé pour nous garder le plus longtemps possible sur le site avec du contenu qui n'est pas toujours des plus pertinents.

Au lieu de choisir la métrique à optimiser, que se passerait-il si les entreprises confiaient à l'utilisateur la responsabilité de choisir sa propre fonction objective ? Et qu'elles prenaient en compte les objectifs personnels du profil de l'utilisateur et lui demandaient ce que vous voudriez réaliser ? À l'heure actuelle, cette technologie est presque comme notre patron et nous ne la contrôlons pas. Ne serait-il pas incroyable de tirer parti de la puissance de système de recommandations pour ressembler davantage à un mentor, un coach ou un assistant ? Le défi consisterait à aligner cela sur les modèles commerciaux existants et à concevoir la bonne interface pour permettre à l'utilisateur de faire ce choix, et de changer à mesure que ses objectifs évoluent. YouTube pourrait par exemple faire des petits pas dans cette direction en chargeant l'utilisateur de sélectionner les catégories pour lesquelles il souhaite voir des recommandations. Mais serait-ce la voie à suivre ou serait-ce juste le rêve du consommateur ? Un tel raisonnement pourraient coûter des millions de dollars à ce genre d'entreprises qui auraient alors moins d'emprise sur nos comportements et nos désirs.

14. Expérimentation des techniques de recommandations

14.1. Présentation et objectif de l'expérience

L'expérimentation qui va être présentée a pour but de développer chez le lecteur une intuition par rapport au potentiel des différents algorithmes de recommandations. Il peut s'avérer difficile de se représenter la qualité des recommandations au travers des métriques de performances. Ces métriques restent essentielles d'un point de vue technique afin de juger de la performance d'un algorithme, mais ne nous donnent pas une représentation concrète en termes de résultats. L'objectif de cette expérience n'est donc pas de faire une comparaison purement technique mais bien de développer une intuition chez le lecteur sur les qualités de chaque algorithme en termes de sérendipité, pertinence et attractivité.

14.2. Méthodologie

Pour mener cette expérience, j'ai mis à contribution mes compétences de programmeur afin de mettre en pratique les algorithmes précédemment vus. Les paramètres utilisés par chaque algorithme seront indiqués ainsi que le score obtenu pour chaque recommandation. Tout au long de l'expérimentation, le même jeu de données sera utilisé afin de pouvoir faire une comparaison des résultats. Seules les recommandations basées sur le contenu auront une variante afin d'accéder à plus de champs textuels.

Pour les algorithmes demandant un historique d'interactions, deux profils "utilisateur" ont été créés avec des interactions antérieures. Ces profils ont volontairement été orientés sur un goût particulier afin de mesurer le degré de personnalisation. Un nombre de 10 interactions pour chaque utilisateur a été choisi afin de ne pas tomber dans le problème du démarrage à froid. Pour les résultats demandant un item en entrée, 3 items aux caractéristiques différentes ont été choisis afin d'observer comment les algorithmes s'y adaptent.

14.3. Données

14.3.1. Sélection du jeu de données

Pour créer des recommandations, nous devons d'abord trouver un jeu de données sur lequel nous puissions travailler. Lors de mes recherches, j'ai pu constater la difficulté à trouver un jeu de données pertinent incluant des données implicites. J'ai donc orienté mes recherches sur des données explicites.

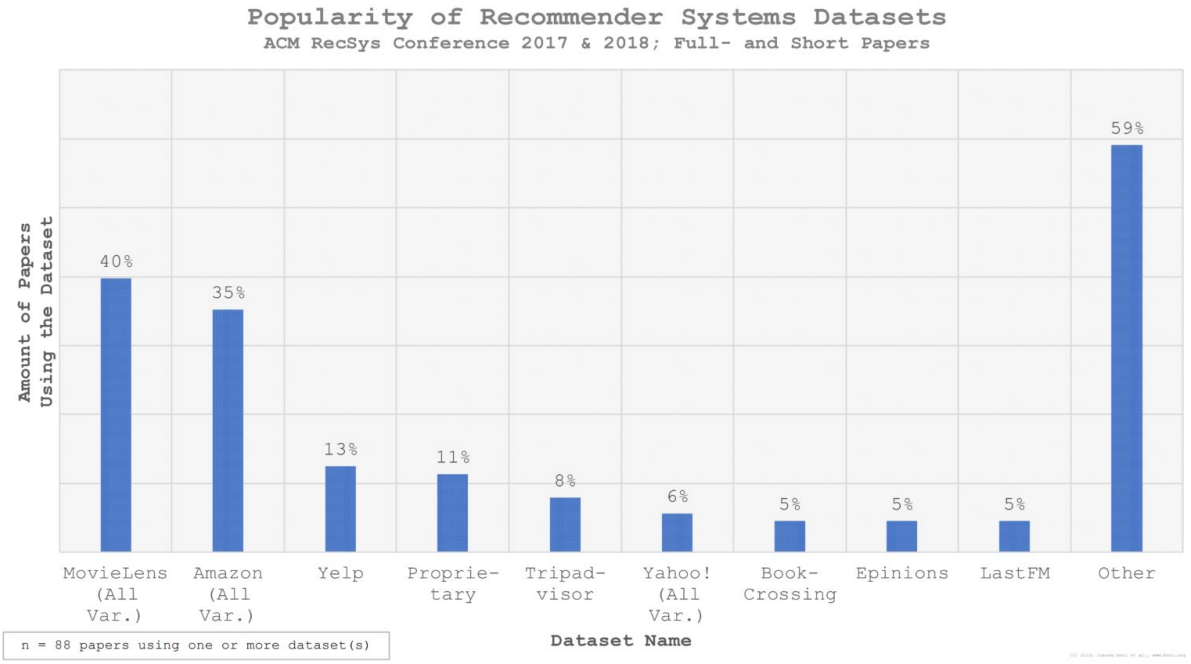

Figure 29. Popularité des jeux de données dans les études des systèmes de recommandations (Data Pruning in Recommender Systems Research: Best-Practice or Malpractice ?).

La figure ci-dessus nous montre les jeux de données les plus utilisés pour l'étude des systèmes de recommandations. La popularité a été mesurée sur base du nombre d'apparitions de chaque jeu de données dans des papiers scientifiques. On peut constater que deux jeux prédominent dans le choix des chercheurs, "MovieLens" qui contient des notes d'utilisateurs pour des films et "Amazon" qui contient des avis d'utilisateurs pour des produits.

Après réflexion, mon choix s'est porté sur le jeu de données fourni librement par MovieLens (*GroupLens*). La thématique des données devait permettre à chacun de pouvoir se faire aisément une opinion sur les résultats présentés, c'est pourquoi mon choix s'est orienté sur la cinématographie. MovieLens est une plateforme crée par une université du Minnesota ayant pour but principal de collecter des données pour l'étude des systèmes de recommandations. Le site web de MovieLens permet de collecter des votes d'utilisateurs qui sont ensuite rendus publics au travers de différents jeux de données. Ce projet à but non lucratif a connu un gros succès et a ainsi pu mettre à disposition plus de 25 millions de votes. Les jeux proposés varient selon leur taille et leur date de mise à jour. Nous utiliserons quant à nous le jeu de données le plus récent qui a été mis à jour le 9/2018 regroupant un peu plus de 100.000 votes et aux alentours de 6500 films (*https://grouplens.org/datasets/movielens/ml-latest-small.zip*). La taille de ce jeu de données est loin de celui qui est le plus complet avec les 25 millions de votes mais est néanmoins suffisante pour obtenir des résultats pertinents tout en permettant de s'affranchir des problèmes liés à la manipulation de volumes de données conséquents.

14.3.2. Exploration des données

Maintenant que nous avons notre jeu de données, nous allons explorer ce qu'il contient. Cette analyse va nous permettre de comprendre la structure de nos données et de tirer des statistiques qui pourront nous servir à l'interprétation des résultats. Notre jeu de données contient deux fichiers textuels ; movies.csv et ratings.csv.

Table des films

	movieId	title	genres	year
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1995
1	2	Jumanji (1995)	Adventure|Children|Fantasy	1995
2	3	Grumpier Old Men (1995)	Comedy|Romance	1995
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	1995
4	5	Father of the Bride Part II (1995)	Comedy	1995
...	...	...	...	...
9737	193581	Black Butler: Book of the Atlantic (2017)	Action|Animation|Comedy|Fantasy	2017
9738	193583	No Game No Life: Zero (2017)	Animation|Comedy|Fantasy	2017
9739	193585	Flint (2017)	Drama	2017
9740	193587	Bungo Stray Dogs: Dead Apple (2018)	Action|Animation	2018
9741	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy	1991

9742 rows × 4 columns

Figure 30. Table des films provenant du fichier "movies.csv".

Le premier fichier "movies.csv" contient la table des films. La colonne movieId représente l'identifiant du film. Nous avons ensuite le titre, les genres attribués au film et la date de release qui a été extraite du titre. Nous avons au total une sélection de 9.742 films par rapport au catalogue complet de MovieLens qui contient 58.000 films. Movielens n'indique malheureusement pas sa méthode de sélection, mais nous retrouvons à priori tous les films populaires ainsi que certains films anciens qui remontent jusqu'à l'année 1902.

L'aperçu des films ci-dessus pourrait nous faire penser que les films commencent à partir de 1995 mais ceux-ci sont en réalité triés sur base de leur date d'ajout et non leur date de release.

Table des votes

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931
...	...	...	...	...
100831	610	166534	4.0	1493848402
100832	610	168248	5.0	1493850091
100833	610	168250	5.0	1494273047
100834	610	168252	5.0	1493846352
100835	610	170875	3.0	1493846415

100836 rows × 4 columns

Figure 31. Table des votes provenant du fichier "ratings.csv".

Le deuxième fichier "ratings.csv" contient la table des votes des utilisateurs pour les films. La colonne movieId est l'identifiant du film, rating est le vote pour le film allant de 0.5 à 5 et le timestamp est la date avec l'heure du vote. Nous voyons qu'il existe aussi un champ userId, néanmoins Movielens ne fournit pas de table par rapport à aux utilisateurs. Nous ne pourrons donc pas utiliser les informations des utilisateurs pour tenter d'améliorer la précision des recommandations.

Degré de sparsité de la matrice user-item

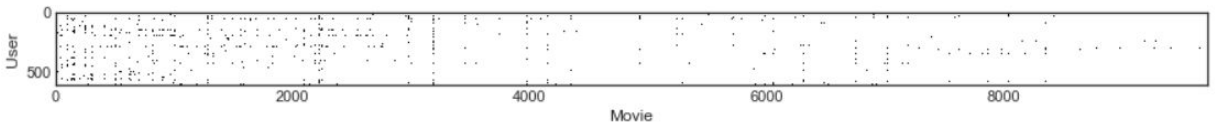

Figure 32. Représentation visuelle de la densité de la matrice des votes (utilisateurs x films). Chaque point représente un vote.

La représentation visuelle de la densité de la matrice des votes (*utilisateurs x films*) nous donne une idée du taux de remplissage de la matrice. Chaque point noir sur la matrice représente un vote d'un utilisateur pour un film. On constate qu'il y a énormément de blanc et donc que la matrice a un taux de densité très faible. Il paraît clair que les utilisateurs n'ont pas pu voir les 9742 films. Le taux de remplissage est de 1,7%, ce qui correspond tout de même à un nombre moyen de votes par utilisateur de 165, ce qui est étonnamment élevé. Après analyse, il s'est avéré qu'un groupe restreint de cinéphiles votent pour une grosse quantité de films.

On constate également que la densité de la matrice diminue en allant vers la droite, c'est-à-dire par rapport aux films les plus récemment ajoutés. La site Movielens a été lancé très tôt en 1995, il se peut que l'application ait perdu de son attrait au fur et à mesure des années par rapport à l'émergence de nombreuses plateformes concurrentes telles que IMDB ou AlloCiné pour les francophones.

Distribution des votes

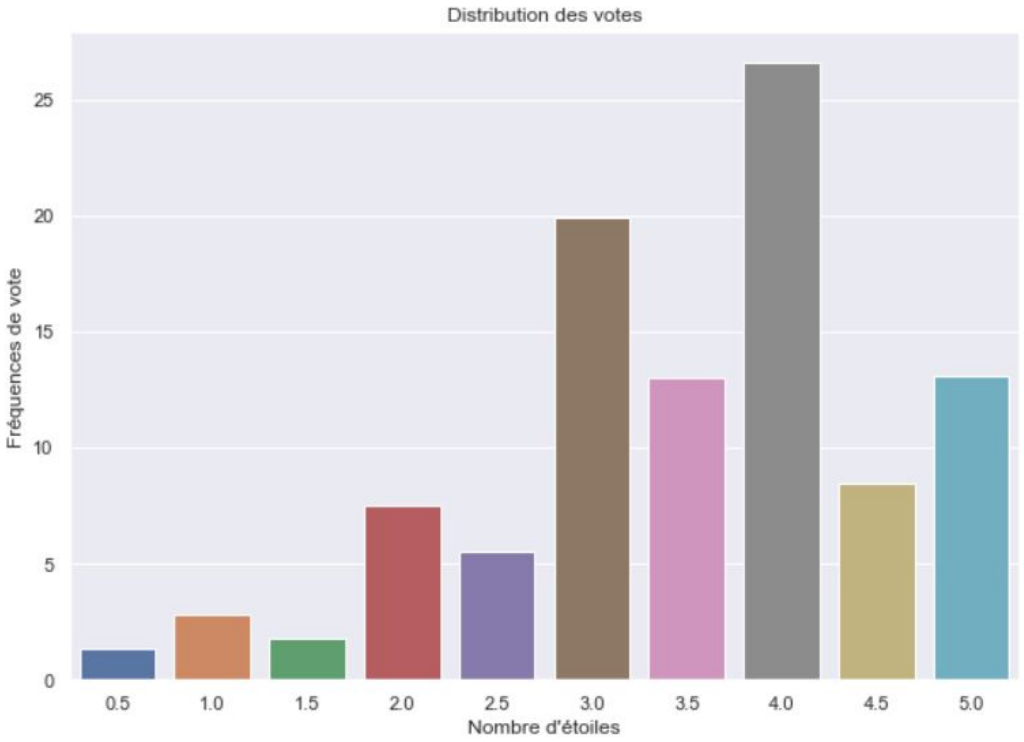

Figure 33. Distribution des votes.

La distribution des votes nous expose la manière dont les utilisateurs votent pour les films. On remarque que les utilisateurs votent assez peu souvent en dessous de 3 étoiles. Les films de mauvaise qualité sont probablement vite ignorés des utilisateurs. Il y a moins de chance pour que vous regardiez un film mal noté ou que vous recommandiez à un ami un film que vous n'avez pas aimé. On remarque également que les votes avec des nombres entiers sont plus récurrents. Cela pourrait s'expliquer par la présentation de la sélection des votes dans l'interface graphique ou bien par simple mécanisme psychologique.

Distribution des votes par film

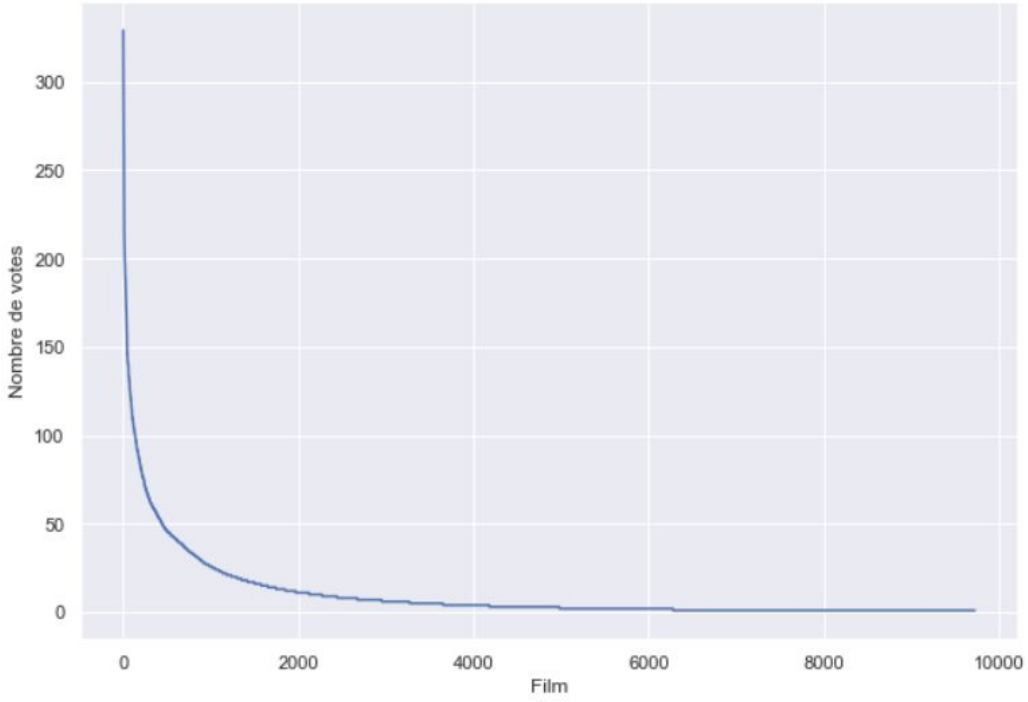

Figure 34. Distribution des votes par film.

La distribution des votes par film nous permet d'observer que seulement une petite partie des films récupèrent la majorité des votes. 4,4% des films ont plus de 50 votes alors que plus de 62,4% n'arrivent pas aux 5 votes. Ces derniers seront susceptibles d'être recommandés dans les algorithmes de filtrage collaboratif étant donné la difficulté de ces derniers à recommander les films ayant peu de notes. Pour permettre l'intégration des nouveaux films dans le filtrage collaboratif, il sera nécessaire de passer par des méthodes telles que les recommandations hybrides ou l'ajout de catégories telles que les nouveautés du moment.

Films les plus populaires

	title	weighted rating
0	Shawshank Redemption, The (1994)	4.337460
1	Godfather, The (1972)	4.162494
2	Fight Club (1999)	4.161576
3	Star Wars: Episode IV - A New Hope (1977)	4.137000
4	Usual Suspects, The (1995)	4.123749
5	Pulp Fiction (1994)	4.121515
6	Schindler's List (1993)	4.119782
7	Matrix, The (1999)	4.110118
8	Star Wars: Episode V - The Empire Strikes Back...	4.107505
9	Forrest Gump (1994)	4.095747
10	Raiders of the Lost Ark (Indiana Jones and the...	4.095093
11	Dark Knight, The (2008)	4.088557
12	Godfather: Part II, The (1974)	4.087090
13	Silence of the Lambs, The (1991)	4.081981
14	Princess Bride, The (1987)	4.077433
15	Goodfellas (1990)	4.075726
16	Departed, The (2006)	4.052881
17	American History X (1998)	4.051834
18	Dr. Strangelove or: How I Learned to Stop Worr...	4.049081
19	One Flew Over the Cuckoo's Nest (1975)	4.044288

Figure 35. Films les plus populaires sur base du vote pondéré.

La liste ci-dessus reprend les films les plus populaires sur base de la formule du vote pondéré développé par IMDB. Si nos systèmes de recommandations ont tendance à recommander des films populaires, nous pouvons nous attendre à retrouver certains de ces films dans les résultats.

Distribution des genres

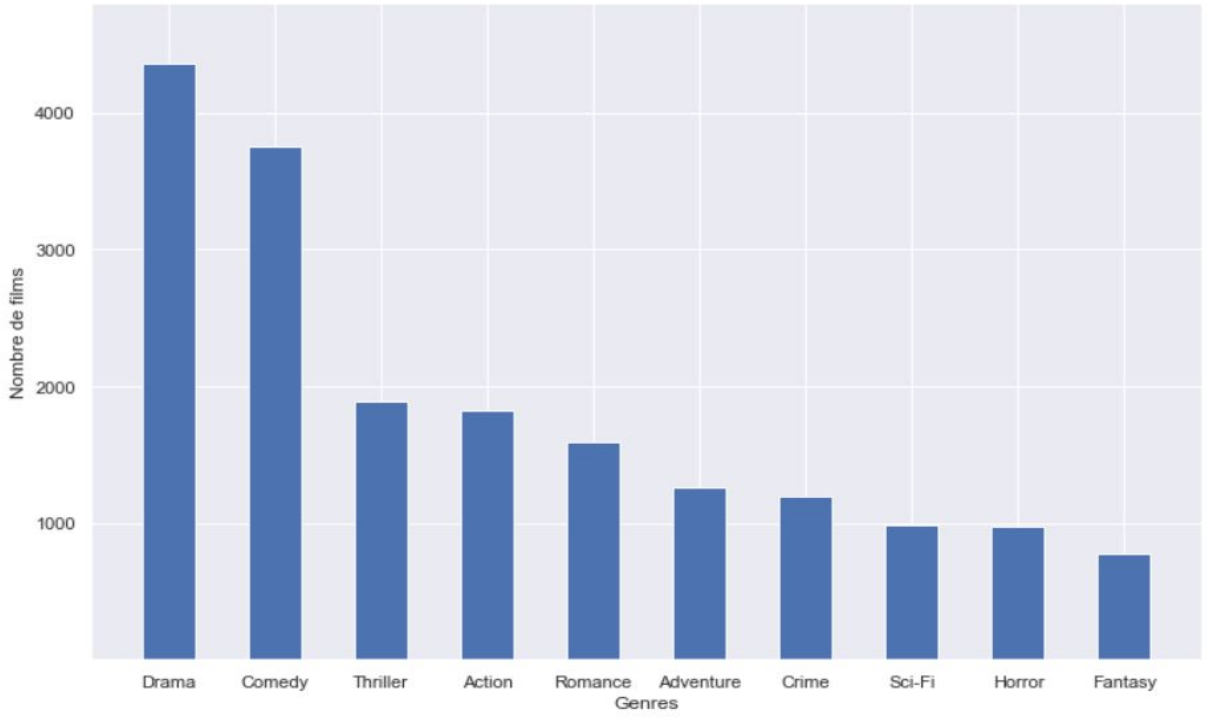

Figure 36. Distribution des genres les plus représentés.

La distribution des genres met en avant deux catégories qui sont davantage représentées que les autres ; les drames et les comédies. Une première intuition sur nos futures recommandations serait de penser que l'on va retrouver une même distribution de genres. Il est cependant difficile de faire de genre d'a priori. A titre d'exemple, on pourrait penser que l'on va retrouver la même proportion de comédies, il se peut cependant que ce genre soit davantage sujet aux critiques et donc puisse être plus facilement ignoré. On peut également citer le nombre relativement faible de films d'animation qui sont pourtant représentés par des films très populaires tels que ceux produits par Pixar. Ces films d'animation pourraient donc être mieux représentés dans les recommandations que le laisse entendre la distribution de genres. Il semble donc en définitive assez difficile de pouvoir tirer des leçons de cette distribution.

14.4. Technologies utilisées (lecture optionnelle)

D'un point de vue technique, les algorithmes ont été entièrement écrit dans le langage Python qui est utilisé de manière courante dans le secteur des sciences de données, notamment grâce à sa communauté importante. Les principales librairies utilisées pour mener cette expérience sont :

- ➢ La librairie Scikit-learn qui a été utilisée pour développer les algorithmes basés sur le contenu et celui sur les voisins les plus proches.
- ➢ La librairie PySpark qui a été utilisée pour la factorisation de matrices à facteurs latents. PySpark propose des algorithmes permettant de manipuler des volumes de données importants et incorpore également par défaut des algorithmes de factorisation de matrices par la méthode d'apprentissage ALS.
- ➢ La librairie Tensorflow qui a été utilisée pour toute la partie deep learning. Cette librairie développée par Google est actuellement l'un des outils les plus utilisés dans le domaine de l'apprentissage automatique.

14.5. Résultats et discussions

14.5.1. Recommandations basées sur le contenu

NB: Les recommandations basées sur le contenu seront les seules basées sur un jeu de données différent. Ce jeu de données comporte des champs supplémentaires dont nous allons nous servir pour construire le sac de mots. Les champs supplémentaires sont la description du film, ses mots-clés, ses acteurs et son réalisateur. Le jeu de données contient également plus de films puisqu'il en compte 45466 au total.

Lien du jeu de données: https://www.kaggle.com/rounakbanik/the-movies-dataset.

Recommandations pour le film: Save the private Ryan

Le sac de mots comprend le titre, la description, les mots clés, les acteurs et le réalisateur.

Similarités cosinus calculées sur base de représentations vectorielles

Titre	***Taux de similarité***	***Genres***
A Time to Love and a Time to Die	23,68%	drama, war, romance
Fortress of War	20,05%	war, drama, history
Grand Illusion	18,85%	drama, history, war
Intimate Enemies	18,61%	drama, war, history
The Great Raid	18,18%	action, history, war
Band of Brothers	18,14%	action, drama, war
The Unknown Soldier	18,12%	drama, war
1898. Our Last Men in the Philippines	18,07%	war, drama, history
War and Peace	17,66%	war, drama, history
Men in War	17,50%	action, drama, war

Similarités cosinus calculées sur base de représentations TF-IDF

Titre	***Taux de similarité***	***Genres***
A Time to Love and a Time to Die	13,53%	drama, war, romance
The Great Raid	12,49%	action, history, war
The Fourth World War	11,00%	documentary
Shoulder Arms	10,98%	comedy, war
Winning Time: Reggie Miller vs. The New York Knicks	10,21%	documentary
Ryan	10,18%	fantasy, documentary, animation
Band of Brothers	9,95%	action, drama, war
Prahaar: The Final Attack	9,79%	
The Odessa File	9,70%	drama, thriller
Men in War	9,59%	action, drama, war

Recommandations pour le film: Toy Story

Le sac de mots comprend le titre, la description, les mots clés, les acteurs et le réalisateur.

Similarités cosinus calculées sur base de représentations vectorielles

Titre	***Taux de similarité***	***Genres***
Toy Story 2	51,20%	animation, comedy, family
Toy Story 3	50,56%	animation, family, comedy
Toy Story of Terror!	26,68%	animation, comedy, family
Small Fry	24,44%	animation, family
Toy Story That Time Forgot	23,12%	animation, family
The Champ	21,55%	drama
Andy Hardy Meets Debutante	18,93%	comedy, family, romance
Love Finds Andy Hardy	18,15%	comedy, romance
Partysaurus Rex	17,11%	animation, comedy, family
You're Only Young Once	15,88%	comedy, romance

Similarités cosinus calculées sur base de représentations TF-IDF

Titre	***Taux de similarité***	***Genres***
Toy Story 3	56,65%	animation, family, comedy
Toy Story 2	53,42%	animation, comedy, family
Small Fry	25,74%	animation, family
Toy Story of Terror!	24,15%	animation, comedy, family
Toy Story That Time Forgot	20,16%	animation, family
The Champ	19,58%	drama
Andy Hardy Meets Debutante	16,96%	comedy, family, romance
Wabash Avenue	15,13%	music
Rebel Without a Cause	14,49%	drama
Burt's Buzz	13,78%	documentary

Recommandations pour le film: Star Wars

Le sac de mots comprend le titre, la description, les mots clés, les acteurs et le réalisateur.

Similarités cosinus calculées sur base de représentations vectorielles

Titre	***Taux de similarité***	***Genres***
The Empire Strikes Back	45,97%	adventure, action, sciencefiction
Star Wars: The Force Awakens	32,50%	action, adventure, sciencefiction
Return of the Jedi	26,73%	adventure, action, sciencefiction
The Swan Princess	20,21%	animation
Star Wars: Episode III - Revenge of the Sith	20,02%	sciencefiction, adventure, action
Empire of Dreams: The Story of the Star Wars Trilogy	18,83%	documentary
Rogue One: A Star Wars Story	18,34%	action, adventure, sciencefiction
Robot Chicken: Star Wars	17,54%	animation, comedy, sciencefiction
Family Guy Presents: Something, Something, Something, Dark Side	16,54%	animation, comedy, sciencefiction
The Princess Bride	16,14%	adventure, family, fantasy

Similarités cosinus calculées sur base de représentations TF-IDF

Titre	***Taux de similarité***	***Genres***
The Empire Strikes Back	49,22%	adventure, action, sciencefiction
Return of the Jedi	25,89%	adventure, action, sciencefiction
Star Wars: The Force Awakens	24,35%	action, adventure, sciencefiction
Empire of Dreams: The Story of the Star Wars Trilogy	16,10%	documentary
Star Wars: Episode III - Revenge of the Sith	15,56%	sciencefiction, adventure, action
Robot Chicken: Star Wars	15,25%	animation, comedy, sciencefiction
The Swan Princess	13,84%	animation
Family Guy Presents: Something, Something, Something, Dark Side	12,89%	animation, comedy, sciencefiction
Rogue One: A Star Wars Story	11,37%	action, adventure, sciencefiction
The Princess and the Pirate	11,34%	romance, comedy, adventure

Les recommandations semblent à première vue assez pauvres. L'algorithme recommande en premier lieu les films d'une même suite qui partagent généralement beaucoup de similarités en commun. Recommander une suite pourrait se faire sur base d'une simple requête en base de données si les films sont liés, ce qui n'est pas le cas dans notre jeu de données. Pour notre situation, l'algorithme reste donc une solution simple pour palier à ce problème.

Les résultats nous donnent également un aperçu de ce qui peut potentiellement être recommandé dans une approche hybride où les recommandations basées sur le contenu et les recommandations basées sur le filtrage collaboratif se mélangent. Bien que les recommandations basées sur le contenu soient pauvres à elles seules, elles restent toujours une bonne solution pour l'intégration de films ayant peu de notes.

Les différences entre la représentation vectorielle et la méthode TF-IDF ne semblent pas montrer beaucoup de différences dans les résultats. La longueur des textes comparés n'est sans doute pas suffisamment importante pour que l'algorithme TF-IDF puisse donner de meilleurs résultats. Finalement, la simplicité de l'algorithme est sans doute son principal avantage. Comme les résultats le montrent, seuls, nous avons davantage affaire à un moteur de recherche textuel qu'à un véritable système de recommandations.

14.5.2. Recommandations item-item basées sur le filtrage collaboratif

Recommandations pour le film: Saving Private Ryan (1998)		
Similarité cosinus. Valeur de k: 20		
Titre	***Similarité cosinus***	***Genres***
Matrix, The (1999)	0,32	Action, Sci-Fi, Thriller
Sixth Sense, The (1999)	0,39	Drama, Horror, Mystery
Star Wars: Episode VI - Return of the Jedi (1983)	0,39	Action, Adventure, Sci-Fi
Star Wars: Episode V - The Empire Strikes ...	0,39	Action, Adventure, Sci-Fi
Star Wars: Episode IV - A New Hope (1977)	0,41	Action, Adventure, Sci-Fi
Gladiator (2000)	0,41	Action, Adventure, Drama
Terminator, The (1984)	0,42	Action, Sci-Fi, Thriller
Men in Black (a.k.a. MIB) (1997)	0,42	Action, Comedy, Sci-Fi
Die Hard (1988)	0,43	Action, Crime, Thriller
Fight Club (1999)	0,43	Action, Crime, Drama, Thriller

Recommandations pour le film: Toy Story (1995)		
Similarité cosinus. Valeur de k: 20		
Titre	***Similarité cosinus***	***Genres***
Toy Story 2 (1999)	0,43	Adventure, Animation, Children
Jurassic Park (1993)	0,44	Action, Adventure, Sci-Fi
Independence Day (a.k.a. ID4) (1996)	0,44	Action, Adventure, Sci-Fi
Star Wars: Episode IV - A New Hope (1977)	0,44	Action, Adventure, Sci-Fi
Forrest Gump (1994)	0,45	Comedy, Drama, Romance
Lion King, The (1994)	0,46	Adventure, Animation, Children
Star Wars: Episode VI - Return of the Jedi (1983)	0,46	Action, Adventure, Sci-Fi
Mission: Impossible (1996)	0,46	Action, Adventure, Mystery
Groundhog Day (1993)	0,47	Comedy, Fantasy, Romance
Aladdin (1992)	0,47	Adventure, Animation, Children

Recommandations pour le film: Social Network, The (2010)		
Similarité cosinus. Valeur de k: 20		
Titre	***Sim. cosinus***	***Genres***
Hangover, The (2009)	0,43	Comedy, Crime
Inception (2010)	0,45	Action, Crime, Drama, Mystery
Inglourious Basterds (2009)	0,47	Action, Drama, War
Moneyball (2011)	0,48	Drama
Fighter, The (2010)	0,48	Drama
Django Unchained (2012)	0,48	Action, Drama, Western
Dark Knight Rises, The (2012)	0,49	Action, Adventure, Crime, IMAX
Dark Knight, The (2008)	0,49	Action, Crime, Drama, IMAX
Shutter Island (2010)	0,51	Drama, Mystery, Thriller
Black Swan (2010)	0,51	Drama, Thriller

Les recommandations basées sur le filtrage collaboratif item-item semblent relativement pertinentes par rapport au film donné en paramètre. On remarque que l'algorithme a néanmoins tendance à recommander des films connus au détriment d'autres films. Après avoir exploré les 50 premiers résultats, j'ai pu constater que le phénomène se maintenait.

Les genres recommandés semblent fortement liés au film de départ ; "Save the private Ryan" a 9 recommandations sur 10 ayant le genre "Action" et "Social Network, The (2010)" a 8 recommandations sur 10 ayant le genre "Drama". En revanche, pour le film "Toy Story (1995)", le genre "Children" ne ressort que 3 fois contre 7 fois pour le genre "Adventure". Il est probable que les utilisateurs de MovieLens soient principalement des adultes qui votent également pour d'autres films que les films pour enfants. Ces résultats nous indique donc probablement que les personnes aimant les films pour enfant aiment aussi les films d'aventure.

Enfin, on remarque que la date du film d'entrée semble influencer de manière importante les recommandations. Pour le film "Social Network, The (2010)", toutes les recommandations sont postérieures à 2008 alors que pour le film "Saving Private Ryan (1998)", toutes sont antérieures à 2001 et pour le film "Toy Story (1995)", antérieures à 2000. Il semblerait que les utilisateurs soient plus enclins à apprécier des films de mêmes époques.

14.5.3. Recommandations basées sur le filtrage collaboratif par facteurs latents

Recommandations pour l'utilisateur A

Nombre de facteurs latents: 25. Paramètre de régularisation: 0.1. Nombre d'itérations: 10. Stratégie de démarrage à froid: drop. Root-mean-square error: 1.277

Films précédemments notés

Titre	***Note***	***Genres***
Star Wars: Episode V - The Empire Strikes Back...	4	Action, Adventure, Sci-Fi
Saving Private Ryan (1998)	4,5	Action, Drama, War
Gladiator (2000)	5	Action, Adventure, Drama
Interstellar (2014)	4,5	Sci-Fi, IMAX
Lord of the Rings: The Return of the King, The...	4	Action, Adventure, Drama, Fantasy
Skyfall (2012)	4,5	Action, Adventure, Thriller, IMAX
Gone Girl (2014)	4,5	Drama, Thriller
The Martian (2015)	4,5	Adventure, Drama, Sci-Fi
Godfather: Part II, The (1974)	4,5	Crime, Drama
Social Network, The (2010)	5	Drama

Recommandations

Titres	***Note estimée***	***Genres***
Black Swan (2010)	4,93	Drama, Thriller
Godfather, The (1972)	4,91	Crime, Drama
Wallace & Gromit: A Close Shave (1995)	4,88	Animation, Children, Comedy
Girl with the Dragon Tattoo, The (2011)	4,87	Drama, Thriller
Fight Club (1999)	4,84	Action, Crime, Drama, Thriller
Intouchables (2011)	4,80	Comedy, Drama
Amadeus (1984)	4,73	Drama
Schindler's List (1993)	4,71	Drama, War
Big Short, The (2015)	4,70	Drama
Casablanca (1942)	4,67	Drama, Romance

Recommandations pour l'utilisateur B

Nombre de facteurs latents: 25. Paramètre de régularisation: 0.1. Nombre d'itérations: 10. Stratégie de démarrage à froid: drop. Root-mean-square error: 1.277

Films précédemments notés

Titre	***Note***	***Genres***
Toy Story (1995)	5	Adventure, Animation, Children
Aladdin (1992)	4,5	Adventure, Animation, Children
Monsters, Inc. (2001)	4	Adventure, Animation, Children
Up (2009)	4,5	Adventure, Animation, Children
Ratatouille (2007)	4,5	Animation, Children, Drama
WALL·E (2008)	4,5	Adventure, Animation, Children
Zootopia (2016)	4,5	Action, Adventure, Animation
Toy Story 3 (2010)	4,5	Adventure, Animation, Children
Incredibles 2 (2018)	4,5	Action, Adventure, Animation
Cinderella (2015)	4,5	Children, Drama, Fantasy

Recommandations

Titre	***Note estimée***	***Genres***
Graduate, The (1967)	6,24	Comedy, Drama, Romance
Do the Right Thing (1989)	5,96	Drama
Maltese Falcon, The (1941)	5,93	Film-Noir, Mystery
Usual Suspects, The (1995)	5,93	Crime, Mystery, Thriller
Sleeper (1973)	5,82	Comedy, Sci-Fi
All About My Mother (Todo sobre mi madre) (1999)	5,80	Drama
Raise the Red Lantern (Da hong deng long gao gao gua) (1991)	5,71	Drama
Moon (2009)	5,70	Drama, Mystery, Sci-Fi, Thriller
Eternal Sunshine of the Spotless Mind (2004)	5,66	Drama, Romance, Sci-Fi
Amadeus (1984)	5,66	Drama

NB : On remarque que certaines notes estimées dépassent la note maximale que peut donner un utilisateur. L'algorithme n'a pas de borne pour ses estimations et peut donc prédire des notes supérieures à la limite donnée aux utilisateurs. Une note supérieure à 5 étoiles peut être considérée comme un taux de confiance supplémentaire donné pour la recommandation.

Malgré la présence de nombreux films populaires, les recommandations pour l'utilisateur A et B semblent relativement intéressantes. Ces recommandations ne doivent être cependant prises que comme un échantillon des films pouvant être recommandés pour l'utilisateur étant donné que l'on retrouve tout de même 255 films ayant une note estimée d'au moins 4 étoiles pour l'utilisateur A et 635 films pour l'utilisateur B. Ces films sont autant de candidats potentiels pouvant élargir le choix proposé à l'utilisateur. Ces recommandations devrait donc être davantage pris comme un modèle de génération de candidats suivi d'un modèle de classement.

Recommandations pour le film: Saving Private Ryan (1998)		
ratings, regularization_coeff=0.5, gravity_coeff=1., embedding_dim=35, init_stddev=.05, num_iterations=2000, learning_rate=5, train_error_observed=1.720610, test_error_observed=2.578339, observed_loss=1.720610, regularization_loss=1.519472, gravity_loss=0.83660258		
Titre	***Similarité Cosinus***	***Genres***
Saving Private Ryan (1998)	1,00	Action, Drama, War
Sixth Sense, The (1999)	0,93	Drama, Horror, Mystery
Die Hard (1988)	0,92	Action, Crime, Thriller
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	0,92	Action, Adventure
Gladiator (2000)	0,91	Action, Adventure, Drama
Matrix, The (1999)	0,91	Action, Sci-Fi, Thriller
Hunt for Red October, The (1990)	0,91	Action, Adventure, Thriller
Terminator, The (1984)	0,91	Action, Sci-Fi, Thriller
Star Wars: Episode VI - Return of the Jedi (1983)	0,90	Action, Adventure, Sci-Fi
Godfather, The (1972)	0,90	Crime, Drama
Titre	***Produit***	***Genres***
Shawshank Redemption, The (1994)	9,49	Action, Drama, War
Saving Private Ryan (1998)	9,40	Drama, War
Schindler's List (1993)	9,37	Action, Adventure, Sci-Fi
Star Wars: Episode IV - A New Hope (1977)	9,35	Action, Sci-Fi, Thriller
Matrix, The (1999)	9,23	Crime, Horror, Thriller
Silence of the Lambs, The (1991)	9,15	Comedy, Crime, Drama, Thriller
Pulp Fiction (1994)	9,10	Comedy, Drama, Romance, War
Forrest Gump (1994)	9,05	Action, Adventure
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	8,93	Crime, Mystery, Thriller
Usual Suspects, The (1995)	8,91	Action, Adventure, Sci-Fi

Recommandations pour le film: Toy Story (1995)

ratings, regularization_coeff=0.5, gravity_coeff=1., embedding_dim=35, init_stddev=.05, num_iterations=2000, learning_rate=5, train_error_observed=1.720610, test_error_observed=2.578339, observed_loss=1.720610, regularization_loss=1.519472, gravity_loss=0.83660258

Titre	***Similarité Cosinus***	***Genres***
Toy Story (1995)	1,00	Adventure, Animation, Children, Comedy, Fantasy
Lion King, The (1994)	0,92	Adventure, Animation, Children, Drama, Musical, IMAX
Aladdin (1992)	0,90	Adventure, Animation, Children, Comedy, Musical
Beauty and the Beast (1991)	0,90	Animation, Children, Fantasy, Musical, Romance, IMAX
Hunchback of Notre Dame, The (1996)	0,90	Animation, Children, Drama, Musical, Romance
Mrs, Doubtfire (1993)	0,89	Comedy, Drama
Shawshank Redemption, The (1994)	0,89	Crime, Drama
Birdcage, The (1996)	0,89	Comedy
Babe (1995)	0,89	Children, Drama
Rock, The (1996)	0,88	Action, Adventure, Thriller
Titre	***Produit***	***Genres***
Shawshank Redemption, The (1994)	11,29	Crime, Drama
Toy Story (1995)	10,97	Adventure, Animation, Children, Comedy, Fantasy
Forrest Gump (1994)	10,39	Comedy, Drama, Romance, War
Silence of the Lambs, The (1991)	10,33	Crime, Horror, Thriller
Schindler's List (1993)	9,88	Drama, War
Pulp Fiction (1994)	9,64	Comedy, Crime, Drama, Thriller
Star Wars: Episode IV - A New Hope (1977)	9,62	Action, Adventure, Sci-Fi
Braveheart (1995)	9,49	Action, Drama, War
Usual Suspects, The (1995)	9,48	Crime, Mystery, Thriller
Fugitive, The (1993)	9,38	Thriller

Recommandations pour le film: Social Network, The (2010)		
ratings, regularization_coeff=0.5, gravity_coeff=1., embedding_dim=35, init_stddev=.05, num_iterations=2000, learning_rate=5, train_error_observed=1.720610, test_error_observed=2.578339, observed_loss=1.720610, regularization_loss=1.519472, gravity_loss=0.83660258		
Titre	***Similarité Cosinus***	***Genres***
Social Network, The (2010)	1,00	Drama
Django Unchained (2012)	0,94	Action, Drama, Western
Ex Machina (2015)	0,94	Drama, Sci-Fi, Thriller
Whiplash (2014)	0,93	Drama
Wolf of Wall Street, The (2013)	0,93	Comedy, Crime, Drama
V for Vendetta (2006)	0,93	Action, Sci-Fi, Thriller, IMAX
Gone Girl (2014)	0,93	Drama, Thriller
Inside Job (2010)	0,93	Documentary
Catch Me If You Can (2002)	0,93	Crime, Drama
Intouchables (2011)	0,93	Comedy, Drama
Titre	***Produit***	***Genres***
Shawshank Redemption, The (1994)	6,43	Crime, Drama
Fight Club (1999)	5,80	Action, Crime, Drama, Thriller
Matrix, The (1999)	5,80	Action, Sci-Fi, Thriller
Forrest Gump (1994)	5,74	Comedy, Drama, Romance, War
Pulp Fiction (1994)	5,74	Comedy, Crime, Drama, Thriller
Schindler's List (1993)	5,68	Drama, War
Dark Knight, The (2008)	5,59	Action, Crime, Drama, IMAX
American History X (1998)	5,59	Crime, Drama
Inception (2010)	5,52	Action, Crime, Drama, Mystery, Sci-Fi, Thriller, IMAX
Usual Suspects, The (1995)	5,41	Crime, Mystery, Thriller

Les recommandations collaboratives basées sur le deep learning semblent donner de bons résultats avec la similarité cosinus. On note une meilleure sérendipité pour les recommandations du film “Toy Story (1995)”. Le genre semble également être fortement pris en compte par le modèle puisque le film “Saving Private Ryan (1998)” a 8 recommandations du genre “Action”, le film “Toy Story (1995)” a 6 recommandations du genre “Children” et le film “Social Network, The (2010)” a 8 recommandations du genre “Drama” (le film d’entrée peut se retrouver dans les résultats). L’année de release du film

semble également fortement influencer le modèle puisque les recommandations du film “Social Network, The (2010)” sont toutes postérieures à 2001, pour “Saving Private Ryan (1998)” antérieures à 2001 et pour “Toy Story (1995)” antérieures à 1997.

Les recommandations basées sur le produit semblent donner de moins bons résultats. Le modèle a tendance à recommander les mêmes films populaires pour chaque film donné en entrée. On retrouve ainsi pour ces films les recommandations suivantes : “Shawshank Redemption, The (1994)”, “Forrest Gump (1994)”, “Pulp Fiction (1994)” et “Schindler's List (1993)”. Ces films étaient de plus tous dans la liste des films populaires que nous avions dressée dans l’exploration des données. L'intérêt de ces recommandations semblent donc en conclusion relativement faible.

> NB : Les recommandations pour nos deux profils utilisateur de base n’ont pas donné de résultats concluants. Il semble que le nombre choisi d’interactions antérieures soit trop faible, ce qui nous amène probablement à un problème de démarrage à froid. Plutôt que d’observer des résultats peu pertinents, j’ai décidé de créer un troisième profil utilisateur avec la concaténation des interactions des deux profils de base. Nous aurons donc un historique de 20 interactions avec une répartition des genres plus grande, présentant néanmoins un profil cohérent.

Recommandations Pour l'utilisateur C

ratings, regularization_coeff=0.5, gravity_coeff=1., embedding_dim=35, init_stddev=.05, num_iterations=2000, learning_rate=5, train_error_observed=1.720610, test_error_observed=2.578339, observed_loss=1.720610, regularization_loss=1.519472, gravity_loss=0.83660258

Films précédemments notés

Titre	***Note***	***Genres***
Star Wars: Episode V - The Empire Strikes Back...	4	Action, Adventure, Sci-Fi
Saving Private Ryan (1998)	4,5	Action, Drama, War
Gladiator (2000)	5	Action, Adventure, Drama
Interstellar (2014)	4,5	Sci-Fi, IMAX
Lord of the Rings: The Return of the King, The...	4	Action, Adventure, Drama
Skyfall (2012)	4,5	Action, Adventure, Thriller
Gone Girl (2014)	4,5	Drama, Thriller
The Martian (2015)	4,5	Adventure, Drama, Sci-Fi
Godfather: Part II, The (1974)	4,5	Crime, Drama
Social Network, The (2010)	5	Drama
Toy Story (1995)	5	Adventure, Animation, Children
Aladdin (1992)	4,5	Adventure, Animation, Children
Monsters, Inc. (2001)	4	Adventure, Animation, Children
Up (2009)	4,5	Adventure, Animation, Children
Ratatouille (2007)	4,5	Animation, Children, Drama
WALL·E (2008)	4,5	Adventure, Animation, Children
Zootopia (2016)	4,5	Action, Adventure, Animation
Toy Story 3 (2010)	4,5	Adventure, Animation, Children
Incredibles 2 (2018)	4,5	Action, Adventure, Animation
Cinderella (2015)	4,5	Children, Drama, Fantasy

Recommandations

Titre	***Produit***	***Genres***
Harry Potter and the Goblet of Fire (2005)	7,3E-09	Adventure, Fantasy, Thriller
Star Wars: Episode III - Revenge of the Sith (2005)	6,3E-09	Action, Adventure, Sci-Fi
Bourne Identity, The (2002)	6,2E-09	Action, Mystery, Thriller

X2: X-Men United (2003)	6,0E-09	Action, Adventure, Sci-Fi
Harry Potter and the Order of the Phoenix (2007)	5,8E-09	Adventure, Drama, Fantasy
X-Men: The Last Stand (2006)	5,7E-09	Action, Sci-Fi, Thriller
Chronicles of Narnia: The Lion, the Witch and the Wardrobe, The (2005)	5,7E-09	Adventure, Children, Fantasy
X-Men (2000)	5,6E-09	Action, Adventure, Sci-Fi
Harry Potter and the Prisoner of Azkaban (2004)	5,5E-09	Adventure, Fantasy, IMAX
Star Wars: Episode VI - Return of the Jedi (1983)	5,2E-09	Action, Adventure, Sci-Fi
Star Trek: First Contact (1996)	5,2E-09	Action, Adventure, Sci-Fi
Avatar (2009)	5,1E-09	Action, Adventure, Sci-Fi
Harry Potter and the Sorcerer's Stone (a,k,a, Harry Potter and the Philosopher's Stone) (2001)	5,0E-09	Adventure, Children, Fantasy
Dumb & Dumber (Dumb and Dumber) (1994)	5,0E-09	Adventure, Comedy
Pirates of the Caribbean: Dead Man's Chest (2006)	5,0E-09	Action, Adventure, Fantasy

Recommandations

Titre	***Similarité cosinus***	***Genres***
Hear My Song (1991)	41,50%	Comedy
Iron Man (1931)	41,33%	Drama
Two Family House (2000)	38,40%	Drama
Sex Ed (2014)	37,96%	Comedy, Romance
Ginger Snaps Back: The Beginning (2004)	37,25%	Fantasy, Horror
Heartbreaker (L'Arnacoeur) (2010)	37,24%	Comedy, Romance
U,S, vs, John Lennon, The (2006)	36,15%	Documentary
Dragons: Gift of the Night Fury (2011)	35,37%	Adventure, Animation, Comedy
Being Julia (2004)	33,29%	Comedy, Drama
Elevator to the Gallows (a,k,a, Frantic) (Ascenseur pour l'échafaud) (1958)	31,42%	Crime, Drama, Thriller
Baxter (1989)	30,09%	Drama, Horror
Inkwell, The (1994)	29,80%	Comedy, Drama
Bugsy (1991)	29,70%	Crime, Drama
Frances (1982)	29,69%	Drama
Eros (2004)	29,67%	Drama

NB : Il est important de noter que les résultats donnés par le modèle changent d'un entraînement à l'autre. Les métriques d'erreur restent relativement stables entre chaque entraînement mais il semble que le modèle capture à chaque fois les caractéristiques de manière légèrement différentes. Afin de ne pas présenter des recommandations trop fluctuantes suite à un nouvel entraînement qui peut potentiellement être fait régulièrement, ajouter une étape de classement semble être nécessaire.

Grâce à ce passage à 20 interactions, les recommandations semblent s'être sensiblement améliorées. Etonnamment, c'est le produit qui donne désormais un grand nombre de films populaires alors que la similarité cosinus donne un niveau de sérendipité assez élevé. Une concaténation des deux types de résultats pourrait donner un meilleur équilibre entre les recommandations de films populaires et les découvertes heureuses.

Visualisation de l'embedding des films

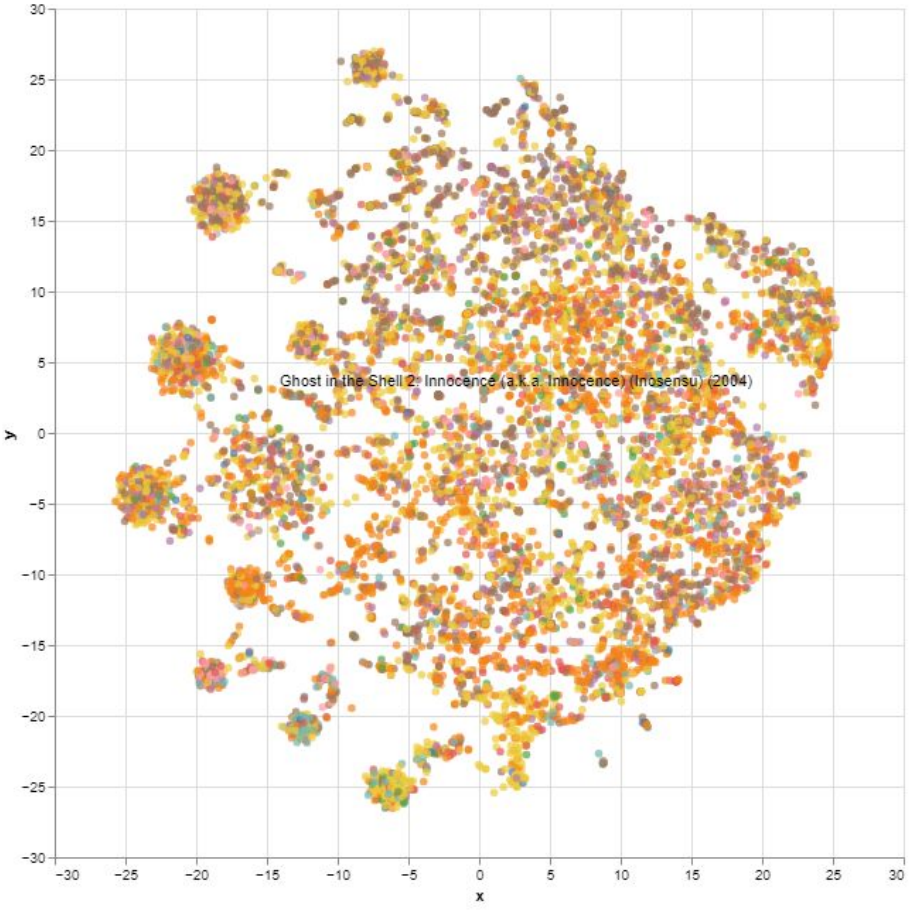

Figure 38. Visualisation 2D de l'embedding des films grâce à l'algorithme t-SNE.

Grâce à l'algorithme t-SNE qui permet une réduction de dimensionnalité en représentant un ensemble de données de grandes dimensions dans un espace de deux ou trois dimensions, il est possible de visualiser le contenu d'un embedding.

La figure 38 représente la visualisation via t-SNE de l'embedding des films de notre modèle utilisé pour nos recommandations avec le deep learning. Chaque point représente un film dans son contexte par rapport aux autres films. On peut constater que des clusters de films se sont formés. Avec les outils digitaux adéquats, il est possible de consulter les films liés à ces points. Cette visualisation personnelle n'étant pas disponible en ligne, il est toujours possible d'interagir avec un exemple de visualisation d'embedding entraîné sur le site : *https://projector.tensorflow.org/*. Les modèles qui y sont présentés concernent un vocabulaire de mots anglais.

14.6. Discussions de l'expérimentation

Les recommandations basées sur le deep learning semblent avoir le plus grand potentiel parmi toutes les techniques explorées. Les méthodes de filtrage collaboratif par modèle nous ont permis de faire nos premières recommandations pertinentes sans connaissances en apprentissage automatique. Ces méthodes ont néanmoins des limites qu'il est difficile de contourner telles que l'ajout de caractéristiques supplémentaires (genres, âge, etc). Là où les techniques traditionnelles se cognent rapidement à des limites, l'apprentissage automatique ouvre de nouvelles opportunités.

Durant mes recherches, j'ai également pu constater les limites imposées par la disponibilité des jeux de données pouvant être utilisés pour l'étude des systèmes de recommandations. Ce sont souvent les mêmes jeux de données que l'on rencontre dans de nombreux papiers de recherche. Ces données sont presque toujours des données explicites alors que les meilleurs systèmes de recommandations utilisent en majorité les données implicites. Pour effectuer des recherches sur les données implicites, certains chercheurs doivent se contenter de transformer artificiellement les données explicites en données

implicites faute de mieux. On peut facilement imaginer l'énorme avantage que possède les ingénieurs ayant accès aux données de leur entreprise.

15. Futur des systèmes de recommandations

15.1. Lorsque la recommandation devient le contenu

Le machine learning s'est invité dans de nombreux domaines, y compris les recommandations. Nous avons découvert comment Spotify parvient à combiner des techniques innovantes pour améliorer ses recommandations. Si nous nous mettons un instant à leur place, comment pourrions-nous voir le monde de demain en termes de recommandations ? Que pourrions-nous inventer de plus ? Imaginons que les recommandations se matérialisent directement dans le contenu lui-même. Les techniques d'apprentissage automatique sont à l'heure actuelle déjà capables de générer des morceaux de musique censés être plaisants à l'oreille humaine. En améliorant ces techniques et en y intégrant les goûts des utilisateurs, il serait possible de créer des musiques personnalisées pour chacun. Les utilisateurs continueraient sans doute à écouter leurs artistes préférés, mais ils écouteraient aussi les chansons créées spécialement pour eux. Les meilleures plateformes musicales tiendraient alors en partie de leur capacité à créer les meilleures chansons.

15.1.1. Techniques prometteuses

Figure 39. L'oeuvre "Portrait d'Edmond de Belamy" générée par un réseau adverse génératif et vendue pour 432 500 $ en octobre. Les GANs ont analysé et synthétisé plus de 15 000 portraits produits depuis le Moyen Âge pour connaître les règles du portrait (Is artificial intelligence set to become art's next medium?, Christies 2018).

Les nouvelles techniques de création de contenu par deep learning semblent prometteuses. Parmi les dernières avancées, nous pouvons citer les réseaux adverses génératifs (generative adversarial networks en anglais ou GANs) qui permettent d'imaginer une création sur base de sources qui leur sont fournies. Ce genre de réseau est représenté par un modèle génératif où deux réseaux sont placés en compétition dans un scénario de théorie des jeux. Le premier réseau est le générateur qui génère un échantillon (ex. une image), tandis que son adversaire, le discriminateur essaie de détecter si un échantillon est réel ou bien s'il est le résultat du générateur. Zalando utilise déjà ce genre de réseaux pour suggérer à ses équipes de design de nouvelles pièces de collection imaginées à partir de pièces existantes (*Generative Fashion Design,* Zalando research). Ce qui serait une réelle avancée pour le consommateur serait d'avoir la possibilité de pouvoir matérialiser rapidement ses créations. On pourrait s'attendre à voir bientôt se

former des synergies avec des technologies telles que l'impression 3D ou des des machines aux capacités semblables.

15.2. Les nouveaux coachs au quotidien

La prochaine décennie annonce l'émergence des objets connectés grâce à l'avènement de la 5G et l'amélioration des techniques embarquées. Ces nouveaux objets sont autant d'opportunités de récupérer des données pouvant servir à la génération de recommandations, voire à leur embarcation auprès de l'utilisateur.

Pour nous rendre compte des possibilités offertes par ces avancées prochaines, rentrons dans la peau d'un sportif. Imaginons que ce sportif soit un cycliste qui ait besoin d'une nouvelle tenue. Celui-ci se rend dans une boutique où il peut rencontrer un miroir intelligent qui est en mesure de l'identifier et de lui recommander des articles sur base de ses données. Miroir qui lui permet également de se visualiser directement avec les vêtements. Plus que de recommander des vêtements existants, le miroir lui propose des créations imaginées sur base de ses goûts. Une fois rentré chez lui avec sa nouvelle tenue, il consulte son programme sportif qui se personnalise en continu sur base de ses performances précédemment récoltées par les capteurs de son vélo, ainsi que de son bracelet intelligent qui capte sa forme du moment. Une fois le vélo enfourché, le cycliste se laisse guider par son système de navigation qui le guide selon les paramètres choisis (difficulté des chemins, qualité des paysages, forme actuelle, etc). Au cours du parcours, l'application se rend compte qu'un ami roule dans les parages avec un profil semblable. L'application recommande alors de fusionner les parcours tout en en créant un nouveau sur base des paramètres des deux cyclistes.

Cet exemple nous le montre, les recommandations pourraient accompagner notre quotidien bien plus que nous pourrions le penser. La 5G permettra le support d'un grand nombre d'objets connectés sur un même réseau et rendra les communications pratiquement temps réel. Grâce à ces objets, il sera possible d'embarquer les recommandations et d'avoir un meilleur contexte sur l'environnement de l'utilisateur. Les systèmes de recommandations deviendraient alors de véritables coachs au quotidien.

16. Conclusion

Les systèmes de recommandations se révèlent comme toutes les sciences, une opportunité pouvant être bénéfique pour le bien de chacun comme pouvant être nuisible. Les systèmes de recommandations apportent des bénéfices indéniables aux utilisateurs qui peuvent consulter du contenu personnalisé selon leurs goûts et envies du moment, où du moins en théorie… Car le maître mot des systèmes actuels semble être l'engagement à tout prix, quitte à parfois diminuer la qualité des recommandations. Les travers de certaines applications actuelles ne se limitent généralement pas aux systèmes de recommandations, mais à leur approche du business. Si nous pouvons regretter ce constat, nous pouvons aussi y voir une opportunité pour créer quelque chose de meilleur. Nous vivons dans un monde de plus en plus disruptif où l'accès à la connaissance n'a jamais été aussi facile. En agissant intelligemment, nous pouvons créer le changement qui fera le monde de demain et le succès de nos entreprises.

17. Bibliographie

IMDb, IMDb.com, help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV#weighted.

Generative Fashion Design, research.zalando.com/welcome/mission/research-projects/generative-fashion-design/.

"5G." *Wikipedia*, Wikimedia Foundation, 16 Oct. 2020, fr.wikipedia.org/wiki/5G.

Baer, Josh. "The Evolution of Big Data at Spotify." *SlideShare*, 4 June 2015, www.slideshare.net/JoshBaer/the-evolution-of-big-data-at-spotify.

Baltrunas, Linas. "Context Aware Recommendations at Netflix." *SlideShare*, fr.slideshare.net/linasbaltrunas9/context-aware-recommendations-at-netflix.

Baltrunas, Linas. "Contextualization at Netflix." *SlideShare*, 26 Sept. 2019, www.slideshare.net/linasbaltrunas9/contextualization-at-netflix.

"Capacité à Informer Des Algorithmes De Recommandation : Une Expérience Sur Le Service YouTube - 2019 - CSA - Conseil Supérieur De L'audiovisuel." *Accueil Du Conseil Supérieur De L'Audiovisuel*, www.csa.fr/Informer/Collections-du-CSA/Focus-Toutes-les-etudes-et-les-comptes-rendus-synthetiques-proposant-un-zoom-sur-un-sujet-d-actualite/Capacite-a-informer-des-algorithmes-de-recommandation-une-experience-sur-le-service-YouTube-2019.

Chen, Yibo, et al. "Solving the Sparsity Problem in Recommender Systems Using Association Retrieval." *Journal of Computers*, vol. 6, no. 9, 2011,

doi:10.4304/jcp.6.9.1896-1902.

"Collaborative Filtering." *Collaborative Filtering - Spark 2.2.0 Documentation*, spark.apache.org/docs/2.2.0/ml-collaborative-filtering.html.

"Collaborative Filtering." *Wikipedia*, Wikimedia Foundation, 14 Oct. 2020, en.wikipedia.org/wiki/Collaborative_filtering.

"Company Info." *Spotify*, newsroom.spotify.com/company-info/.

"Corrélation (Statistiques)." *Wikipedia*, Wikimedia Foundation, 12 Aug. 2020, fr.wikipedia.org/wiki/Corrélation_(statistiques).

Covington, Paul, et al. "Deep Neural Networks for YouTube Recommendations." *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, doi:10.1145/2959100.2959190.

Duzen, Zafer, and Mehmet S. Aktas. "An Approach to Hybrid Personalized Recommender Systems." *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2016, doi:10.1109/inista.2016.7571865.

Engineering, Pinterest. "PinSage: A New Graph Convolutional Neural Network for Web-Scale Recommender Systems." *Medium*, Pinterest Engineering Blog, 30 Nov. 2018, medium.com/pinterest-engineering/pinsage-a-new-graph-convolutional-neural-network-for-web-scale-recommender-systems-88795a107f48.

"Factorisation De Matrices Pour Les Systèmes De Recommandation." *Wikipedia*, Wikimedia Foundation, 6 May 2020, fr.wikipedia.org/wiki/Factorisation_de_matrices_pour_les_systèmes_de_recommandation.

Gomez-Uribe, Carlos A., and Neil Hunt. “The Netflix Recommender System.” *ACM Transactions on Management Information Systems*, vol. 6, no. 4, 2016, pp. 1–19., doi:10.1145/2843948.

Grbovic, Mihajlo. “Listing Embeddings in Search Ranking.” *Medium*, Airbnb Engineering & Data Science, 4 May 2018, medium.com/airbnb-engineering/listing-embeddings-for-similar-listing-recommendations-and-real-time-personalization-in-search-601172f7603e.

Hardesty, Larry. “The History of Amazon's Recommendation Algorithm.” *Amazon Science*, Amazon Science, 20 Aug. 2020, www.amazon.science/the-history-of-amazons-recommendation-algorithm.

Hu, Yifan, et al. “Collaborative Filtering for Implicit Feedback Datasets.” *2008 Eighth IEEE International Conference on Data Mining*, 2008, doi:10.1109/icdm.2008.22.

Johnson, Chris. “From Idea to Execution: Spotify's Discover Weekly.” *SlideShare*, 16 Nov. 2015, fr.slideshare.net/MrChrisJohnson/from-idea-to-execution-spotifys-discover-weekly.

“Marketing on Pinterest: Pinterest Business.” *Pinterest*, business.pinterest.com/en/.

“Matrix Factorization | Recommendation Systems | Google Developers.” *Google*, Google, developers.google.com/machine-learning/recommendation/collaborative/matrix.

“MovieLens.” *GroupLens*, 21 May 2020, grouplens.org/datasets/movielens/.

“Méthode Des k plus Proches Voisins.” *Wikipedia*, Wikimedia Foundation, 9 Aug. 2020, fr.wikipedia.org/wiki/Méthode_des_k_plus_proches_voisins.

Person. “Is Artificial Intelligence Set to Become Art's next Medium?: Christie's.” *The First Piece of AI-Generated Art to Come to Auction | Christie's*, Christies, 12 Dec.

2018,
www.christies.com/features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx.

Schrage, Michael. “Great Digital Companies Build Great Recommendation Engines.” *Harvard Business Review*, 1 Aug. 2017, hbr.org/2017/08/great-digital-companies-build-great-recommendation-engines.

“Similarité Cosinus.” *Wikipedia*, Wikimedia Foundation, 20 Feb. 2020, fr.wikipedia.org/wiki/Similarité_cosinus.

Smith, Brent, and Greg Linden. “Two Decades of Recommender Systems at Amazon.com.” *IEEE Internet Computing*, vol. 21, no. 3, 2017, pp. 12–18., doi:10.1109/mic.2017.72.

“TF-IDF.” *Wikipedia*, Wikimedia Foundation, 2 Sept. 2020, fr.wikipedia.org/wiki/TF-IDF.

UPEM, Publié par Master SHS. “Bulles De Filtre Et Démocratie.” *Les Mondes Numériques*, 28 Jan. 2017, lesmondesnumeriques.wordpress.com/2017/01/28/bulles-de-filtre-et-democratie/.

“Word Embedding.” *Wikipedia*, Wikimedia Foundation, 14 Feb. 2020, fr.wikipedia.org/wiki/Word_embedding.