
Machine Learning for Python

Project : Movie Recommender Systems

ENSAE 2020/2021

MASTÈRE SPÉCIALISÉ - DATA SCIENCE

GROUPE 4

NATHAN BRY - AMALE NOKRI

Table des matières

1	Etat de l'art	1
1.1	Les systèmes de recommandation	1
1.2	Les systèmes de recommandation de filtrage basé sur le contenu	1
1.2.1	Introduction	1
1.2.2	TF-IDF	2
1.2.3	Mesure de la similarité	2
1.3	Les systèmes de recommandation avec filtrage collaboratif	3
1.3.1	Introduction	3
1.3.2	Singular Value Decomposition	3
2	Implémentation	5
2.1	Les données	5
2.2	Exploratory Data Analysis	5
2.3	Implémentation du filtrage basé sur le contenu	7
2.3.1	Pré-traitement de texte	7
2.3.2	TF-IDF et similarité	7
2.3.3	Système de recommandation n°1 (movie-based)	8
2.3.4	Système de recommandation n°2 (keywords-based)	8
2.4	Implémentation de la méthode SVD	10
3	Conclusion	11

1 Etat de l'art

1.1 Les systèmes de recommandation

Un système de recommandation est un système / algorithme intelligent qui prédit l'évaluation et les préférences des utilisateurs pour certains produits. Il vise à fournir aux utilisateurs des recommandations personnalisées de produits, de contenus ou de services afin d'améliorer l'expérience client et maximiser leur engagement.

Mentionnés pour la première fois en 1990, ces systèmes sont aujourd'hui très en vogue si bien que nous y avons affaire au quotidien, que ce soit dans le e-commerce lorsqu'Amazon nous propose des produits à acheter, ou encore sur Netflix ou Spotify lorsque du contenu nous est suggéré (films / séries ou musique respectivement). Les réseaux sociaux tels YouTube, Instagram ou encore LinkedIn utilisent également des systèmes de recommandation. En plus d'aider et d'accompagner les consommateurs dans leurs choix, de réduire leur temps de recherche, et de leur faire découvrir des produits difficiles à trouver, ces systèmes ont également une valeur importante pour l'entreprise :

NETFLIX « Our recommender system influences choice for about 80% of hours streamed at Netflix. [...] We think the combined effect of personalization and recommendations save us more than \$1B per year » C.A. Gomez-Urbe et N.Hunt, 2015

amazon « 35% of what consumers purchase on Amazon comes from product recommendations ” Ian MacKenzie, Chris Meyer, and Steve Noble Open interactive popup 2013

Bien qu'il existe aujourd'hui une multitude de systèmes de recommandation différents, nous faisons le choix de nous intéresser à 2 des méthodes les plus populaires :

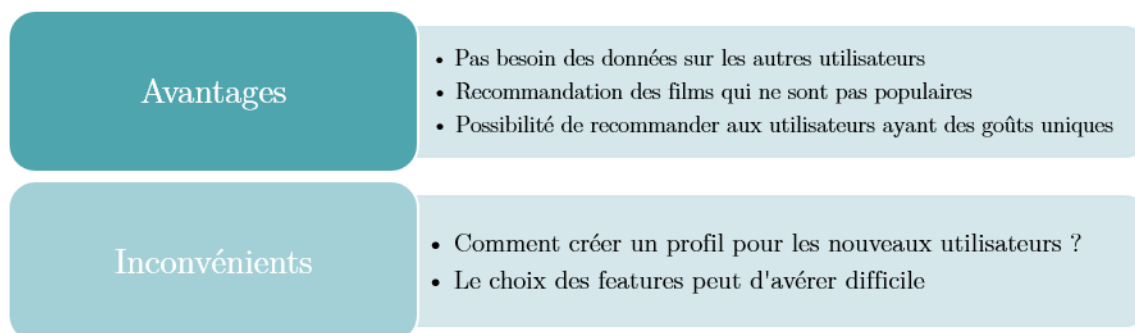
- Les systèmes de recommandation de filtrage basé sur le contenu (Content-Based Filtering) ;
- Les systèmes de recommandation avec filtrage collaboratif (Collaborative Filtering).

1.2 Les systèmes de recommandation de filtrage basé sur le contenu

1.2.1 Introduction

Les systèmes de filtrage à base de contenu recommandent des articles/documents/des films similaires à ceux que l'utilisateur a déjà apprécié. En effet, ce type de système de recommandation se base sur des profils sans prendre en compte les avis des autres utilisateurs (Peis et al, 2008). Ce type de filtrage utilise les préférences antérieures de l'utilisateur-ice pour recommander un produit. Par exemple, si un individu a aimé un film, le système recommandera des films qui appartiennent au même genre.

Le contenu peut représenter bon nombre de choses. Dans le cas d'un film, nous avons : le titre, le réalisateur, les acteurs, le résumé, le genre.



Il existe de nombreuses approches pour effectuer des comparaisons en se basant sur le contenu textuel. Ces approches peuvent être statistiques avec des approches telles que l'analyse démantique latente et l'analyse sémantique explicite, ou vectorielles avec des méthodes de Bi-clustering et de Vecteurs Sémantiques. Nous allons utiliser une approche faisant appel aux Vecteurs Sémantiques.

1.2.2 TF-IDF

Le contenu du profil dépend de la méthode utilisée. Le plus souvent, le profil est effectué sous forme d'un vecteur de mots-clés avec des poids. Ces mots sont, le plus souvent, extraits à l'aide de la mesure TF-IDF (Salton, 1989). Cette mesure est le produit de TF (Term Frequency) et IDF (Inverse Document Frequency), qu'on note :

$$TF_{w,d_i} = \frac{f_{w,d_i}}{\max_{w' \in d_i} f_{w',d_i}}$$

$$IDF_{w,d_i} = \log \frac{|\mathcal{I}|}{|\{i\} \in \mathcal{I} : w \in d_i\|}$$

où f_{w,d_i} est le nombre d'occurrences d'un mot w dans un document d_i .




	TF	IDF	TF-IDF
 <p>Isle of Dogs is a 2018 internationally co-produced stop-motion-animated science-fiction comedy drama film written, produced and directed by Wes Anderson.</p>	0	$\text{Log}(6/2)$	0
 <p>Avengers : Infinity War is a 2018 American superhero film based on the Marvel Comics superhero team the Avengers, produced by Marvel Studios and distributed by Walt Disney Studios Motion.</p>	2	$\text{Log}(6/2)$	0.95
 <p>Aquaman is a 2018 American superhero film based on the DC Comics character of the same name.</p>	1	$\text{Log}(6/2)$	0.47

FIGURE 1 – Exemple d'utilisation TF-IDF

1.2.3 Mesure de la similarité

Afin de mesurer la similarité entre des mots, il existe différentes métriques ; manhattan, euclidien, Person... Au cours de notre projet, nous allons utiliser la mesure cosinus, aussi appelée la similarité cosinus (Cosine Similarity).

La similarité cosinus permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux. Cette métrique est fréquemment utilisée en fouille de texte (baeza-yates and ribeiro neto, 1999).

Mathématiquement, on la définit ainsi :

$$sim_{cos}(d_1, d_2) = \frac{\vec{d_1} \cdot \vec{d_2}}{\|\vec{d_1}\| \|\vec{d_2}\|}$$

Sa valeur est comprise dans l'intervalle $[0,1]$. Plus elle est proche de un, plus les deux mots / textes (d_1 et d_2) sont similaires.

1.3 Les systèmes de recommandation avec filtrage collaboratif

1.3.1 Introduction

Les systèmes de recommandation basés sur le filtrage collaboratif aident les gens à faire des choix en se fondant sur les opinions d'autres personnes qui partagent des intérêts similaires. Un utilisateur recevra ainsi des recommandations d'articles appréciés par des utilisateurs similaires. Le filtrage collaboratif est basé sur l'hypothèse que les personnes qui étaient d'accord dans le passé le seront à l'avenir, et qu'elles aimeront les mêmes types d'articles qu'elles aimaient dans le passé.

Cette approche construit donc un modèle basé sur le comportement passé des utilisateurs. Le comportement de l'utilisateur peut inclure des vidéos regardées précédemment, des articles achetés, des évaluations d'articles ou de contenu, etc. De cette façon, le modèle établit un lien entre les utilisateurs et les articles. Le modèle est ensuite utilisé pour prédire ou un nouveau contenu/article (ou une note pour ce dernier) qui pourrait intéresser l'utilisateur.

L'un des principaux avantages de l'approche de filtrage collaboratif est qu'elle ne repose pas sur une analyse de l'élément en lui-même et qu'elle est donc capable de recommander avec précision des éléments complexes tels que des films ou des musiques, sans exiger une "compréhension" de l'élément lui-même.

La similarité entre les utilisateurs ou les articles peut être calculée par la similarité basée sur la corrélation de Pearson, la similarité basée sur la corrélation de Pearson contrainte, la similarité basée sur le coût ou les mesures ajustées basées sur le coût. Lors du calcul de la similarité entre les éléments à l'aide des mesures ci-dessus, seuls les utilisateurs qui ont évalué les deux éléments sont pris en compte. Cela peut influencer la précision de la similarité lorsque des éléments qui ont reçu un très petit nombre de notes expriment un niveau élevé de similarité avec d'autres éléments.

De nombreux algorithmes ont été utilisés pour mesurer la similarité des utilisateurs ou des éléments dans les systèmes de recommandation. Par exemple, l'approche du plus proche voisin k (k -NN) et la corrélation de Pearson telle qu'elle a été mise en œuvre pour la première fois par Allen.

Cependant nous nous concentrerons sur la décomposition en valeurs singulières (SVD - Singular Value Decomposition), une autre méthode très populaire utilisée comme une approche de filtrage collaboratif dans les systèmes de recommandation et popularisée par Simon Funk lors du Netflix Prize Challenge (2006-2009).

1.3.2 Singular Value Decomposition

La SVD est une méthode de l'algèbre linéaire qui a été généralement utilisée comme technique de réduction de la dimensionnalité en Machine Learning. C'est un des algorithmes d'apprentissage non supervisé les plus utilisés, qui est au centre de nombreux systèmes de recommandation et de réduction de la dimension qui sont au cœur de sociétés omniprésentes dans notre société telles que Google, Netflix, Facebook, Youtube, Amazon, etc. C'est une technique de factorisation matricielle qui réduit le nombre de caractéristiques d'un ensemble de données en réduisant la dimension spatiale de N -dimension à K -dimension (où $K < N$). Elle utilise une structure matricielle où chaque ligne représente un utilisateur et chaque colonne représente un contenu (vidéo, musique, article, etc), les éléments de cette matrice étant les notes qui sont attribuées aux différents contenus par les utilisateurs.

La factorisation de cette matrice se fait par la décomposition en valeurs singulières. Elle permet de trouver des facteurs de matrices à partir de la factorisation d'une matrice de haut niveau (les évaluations des articles par les utilisateurs). La décomposition en valeurs singulières est une méthode de décomposition d'une matrice en trois autres matrices, comme indiqué ci-dessous :

$$P_{m*n} = U_{m*m} \Sigma_{m*n} V_{n*n}^T$$

Où P est une matrice de dimension $m * n$,

U est une matrice orthogonale de dimension $m * m$,

Σ est une matrice diagonale de dimension $m * n$,
 V^T est la transposée de V une matrice singulière de dimension $n * n$.

Ici, la matrice P représente la relation entre les users et les contenus (ratings), la matrice U représente la relation entre les users et les variables latentes (les caractéristiques des contenus), la matrice Σ indique le poids de chaque variable latente, et la matrice V montre la similarité entre les contenus et les variables latentes.

Si on prend un exemple avec 4 users et 5 contenus (tels que des films), on obtient l'équation suivante :

$$P_{4*5} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & u_{44} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4 & 0 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} \\ v_{31} & v_{32} & v_{33} & v_{34} & v_{35} \\ v_{41} & v_{42} & v_{43} & v_{44} & v_{45} \\ v_{51} & v_{52} & v_{53} & v_{54} & v_{55} \end{bmatrix}$$

La préférence du user 1 pour le film 1 est alors donnée par :

$$p_{11} = \sigma_1 u_{11} v_{11} + \sigma_2 u_{12} v_{21} + \sigma_3 u_{13} v_{31} + \sigma_4 u_{14} v_{41}$$

2 Implémentation

Après avoir présenté 2 types de systèmes de recommandation très populaires, nous utiliserons Python pour mettre en pratique ces concepts et créer notre propre version de ces systèmes afin de recommander des films. Le code sera fait dans un Jupyter Notebook, et les résultats principaux seront également détaillés dans la suite de ce rapport.

2.1 Les données

Afin d'implémenter nos algorithmes nous utiliserons un dataset mis à disposition sur Kaggle. Le dataset consiste en une concaténation des 2 bases de données les plus connues du domaine cinématographique :

- MovieLens, un dataset (collecté par GroupLens) contenant 26 000 000 de notes données à 45 000 films par 270 000 utilisateurs.
- IMDb, une base de données (collectée par TMDb) contenant des informations sur les films, telles que les acteurs et réalisateurs, les budgets, les résumés, etc.

Nous commencerons par une brève analyse du dataset complet afin de découvrir et prendre en main nos données. Cependant, pour l'implémentation des algorithmes et par souci de ressources informatiques limitées, nous utiliserons une version réduite du dataset contenant 100 000 notes données à 9099 films par 671 utilisateurs.

2.2 Exploratory Data Analysis

Ratings

Comme nous pouvons le voir sur l'image ci-dessous, les notes (ratings) sont généralement bonnes. En effet, près de la moitié sont de 4+ sur 5.

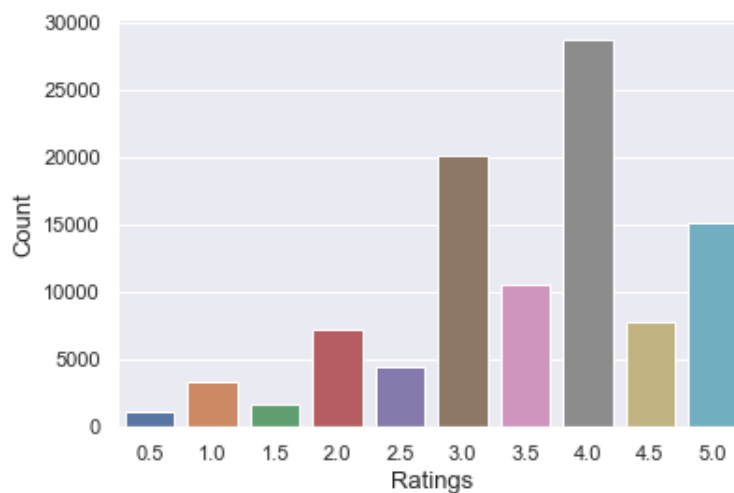


FIGURE 2 – Count des différentes notes

Genres

Les films sont parfois composés de plusieurs genres, par exemple Toy Story est un film comédie, de famille, et d'aventure. Au total, 20 genres sont répertoriés.

"live", "life" qui pourraient faire référence aux films où des héros cherchent quelque chose qui changera leur vie quotidienne. Nous voyons aussi des mots comme "friend", "famili" et "love" qui font probablement référence à des drames ou des comédies.

2.3 Implémentation du filtrage basé sur le contenu

2.3.1 Pré-traitement de texte

Nous allons mettre en place un système de recommandation basé sur les résumés des films (overviews). Avant d'élaborer ce système, nous avons dû effectuer un pré-traitement de texte. Cette étape essentielle se fait en appliquant diverses méthodes de NLP et permet de standardiser le texte afin de rendre son interprétation et analyse plus facile.

Nous avons suivi les étapes suivantes :



Après avoir mis tous les mots en minuscule, nous avons supprimé la ponctuation. Nous avons ensuite supprimé les mots communs en anglais (appelés les stop words, tel que « I », « it », « this », etc.). Enfin, nous avons réduit chaque mot à sa racine (cats → cat, banking → bank) grâce au Stemming. Par exemple, le résumé du film Toy Story avant et après le pré-traitement :

Avant	Après
Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene. Afraid of losing his place in Andy's heart, Woody plots against Buzz. But when circumstances separate Buzz and Woody from their owner, the duo eventually learns to put aside their differences.	lead woody andy toy live happily room andy birthday brings buzz lightyear onto scene afraid lose place andy heart woody plot buzz circumstance separate buzz woody owner duo eventually learn put aside difference

2.3.2 TF-IDF et similarité

Nous avons converti les mots des résumés en matrice de features TF-IDF.

Au total, il y a près de 21 000 mots différents sur les 9 099 résumés. Nous avons ensuite calculé le score de similarité en utilisant la Cosine Similarity.

Par exemple, pour le résumé 1 (Toy Story), le score de similarité du premier mot est de 1.

1	0.01847	...	0	0.00697
[0]	[1]		[32]	[33]

Essayons maintenant de construire des systèmes de recommandation de filtrage de contenu. Nous aurons 2 approches différentes (et pourtant assez similaires). Toutes deux utiliseront la TF-IDF et la Cosine Similarity.

2.3.3 Système de recommandation n°1 (movie-based)

Nous commencerons par un système qui, à partir d'un film en entrée, recommande des films similaires (dont les résumés sont similaires).

Nous construisons d'abord une matrice où chaque ligne est un film, chaque colonne un mot, et les valeurs sont les valeurs TF-IDF (chaque mot a une valeur pour chaque film).

Ensuite, nous avons construit une matrice où les lignes et les colonnes représentent toutes deux des films, et les valeurs sont les scores de similarité en cosinus entre les deux films impliqués.

Enfin, pour un film donné en input, nous renvoyons les films avec la plus grande Cosine Similarity.

Les différentes étapes étant disponibles et expliquées dans le Notebook HTML, nous nous concentrerons ici sur les résultats.

```
movies_recommender2("toy story",5)
```

	title	similarity_score	vote_average	vote_count
7545	Toy Story 3	0.524871	7.6	4710.0
2504	Toy Story 2	0.466925	7.3	3914.0
6201	The 40 Year Old Virgin	0.298939	6.2	2020.0
1702	Pretty in Pink	0.211899	6.6	313.0
6563	For Your Consideration	0.206966	5.9	44.0

Comme on peut le voir, notre système a recommandé Toy Story 2 & 3 comme étant les films les plus similaires à Toy Story, ce qui est certainement une bonne chose. Cependant, nous constatons également que certains films n'ayant pratiquement aucun lien ont été recommandés (en dehors de la similitude des résumés), ce qui est un peu décevant. En effet, nous doutons qu'un enfant voulant regarder un film de Walt Disney soit intéressé par "The 40 Year Old Virgin"... Essayons une autre approche pour le filtrage de contenu !

2.3.4 Système de recommandation n°2 (keywords-based)

Nous allons ici mettre en place un système de recommandation qui, au lieu de prendre un film en entrée, peut prendre des mots-clés. Par exemple, "J'aimerais regarder un film sur des princesses et l'amour, que me recommandez-vous?".

Ici aussi, nous utiliserons TD-IDF et Cosine Similarity. Ainsi, au lieu de calculer la Cosine Similarity entre les films (celui donné en input vs ceux que nous avons dans notre base de données), nous la calculerons entre une requête donnée en input (faite d'un ou plusieurs mots) et les films que nous avons dans notre base de données.

Pour ce faire, nous devons créer plusieurs index. Pour accélérer l'exécution du code, nous ne créerons ces index qu'une seule fois et les conserverons pour de futures recommandations.

C'est pourquoi le code utilisé pour créer ces index a été transformé en commentaires, tandis que certaines cellules utilisées pour charger les index ont été créées.

Les différentes étapes de création des index étant disponibles dans le Notebook HTML, nous nous concentrerons ici sur les résultats.

movies_recommender3('princess love')						
	title	genres	overview	score	vote_count	vote_average
0	Aladdin	Animation - Family - Comedy - Adventure - Fantasy - Romance	Princess Jasmine grows tired of being forced to remain in the palace and she sneaks out into the marketplace in disguise where she meets street-urchin Aladdin and the two fall in love, although she may only marry a prince. After being thrown in jail, Aladdin and becomes embroiled in a plot to f...	1.00	3,495.00	7.40
1	The Little Mermaid	Animation - Family	This colorful adventure tells the story of an impetuous mermaid princess named Ariel who falls in love with the very human Prince Eric and puts everything on the line for the chance to be with him. Memorable songs and characters -- including the villainous sea witch Ursula.	1.00	1,921.00	7.20
2	Royal Wedding	Comedy - Music - Romance	Fred Astaire (Tom) and Jane Powell (Ellen) are asked to perform as a dance team in England at the time of Princess Elizabeth's wedding. As brother and sister, each develops a British love interest, Ellen with Lord John Brindale (Peter Lawford) and Tom with dancer Anne Ashmond (Sarah Churchill--W...	1.00	17.00	6.30
3	Mulan II	Animation - Comedy - Family - Action	Fa Mulan gets the surprise of her young life when her love, Captain Li Shang asks for her hand in marriage. Before the two can have their happily ever after, the Emperor assigns them a secret mission, to escort three princesses to Chang'an, China. Mushu is determined to drive a wedge between the...	1.00	450.00	6.00
4	Antz	Adventure - Animation - Comedy - Family	In this animated hit, a neurotic worker ant in love with a rebellious princess rises to unlikely stardom when he switches places with a soldier. Signing up to march in a parade, he ends up under the command of a bloodthirsty general. But he's actually been enlisted to fight against a termite army.	1.00	1,320.00	6.00
5	Mirror Mirror	Adventure - Fantasy - Drama - Comedy - Science Fiction - Family	After she spends all her money, an evil enchantress queen schemes to marry a handsome, wealthy prince. There's just one problem - he's in love with a beautiful princess, Snow White. Now, joined by seven rebellious dwarves, Snow White launches an epic battle of good vs. evil...	1.00	1,148.00	5.50
6	The Magic Flute	Fantasy - Comedy - Music - Romance	The Queen of the Night enlists a handsome prince named Tamino to rescue her beautiful kidnapped daughter, Princess Pamina, in this screen adaptation of the beloved Mozart opera. Aided by the lovelorn bird hunter Papageno and a magical flute that holds the power to change the hearts of men, young...	1.00	15.00	7.10
7	Clash of the Titans	Adventure - Fantasy - Family	To win the right to marry his love, the beautiful princess Andromeda, and fulfil his destiny, Perseus must complete various tasks including taming Pegasus, capturing Medusa's head, and battling the Kraken monster.	1.00	208.00	6.80
8	Jack the Giant Slayer	Action - Family - Fantasy	The story of an ancient war that is reignited when a young farmhand unwittingly opens a gateway between our world and a fearsome race of giants. Unleashed on the Earth for the first time in centuries, the giants strive to reclaim the land they once lost, forcing the young man, Jack into the batt...	1.00	2,634.00	5.50
9	Enchanted	Comedy - Family - Fantasy - Romance	The beautiful princess Giselle is banished by an evil queen from her magical, musical animated land and finds herself in the gritty reality of the streets of modern-day Manhattan. Shocked by this strange new environment that doesn't operate on a "happily ever after" basis, Giselle is now adrift ...	1.00	1,512.00	6.60
10	The Swan Princess	Animation	The beautiful princess Odette is transformed into a swan by an evil sorcerer's spell. Held captive at an enchanted lake, she befriends Jean-Bob the frog, Speed the turtle and Puffin the bird. Despite their struggle to keep the princess safe, these good-natured creatures can do nothing about the ...	0.98	251.00	6.50
11	Sleeping Beauty	Fantasy - Animation - Romance - Family	A beautiful princess born in a faraway kingdom is destined by a terrible curse to prick her finger on the spindle of a spinning wheel and fall into a deep sleep that can only be awakened by true love's first kiss. Determined to protect her, her parents ask three fairies to raise her in hiding. B...	0.98	1,332.00	6.80
12	Frozen	Animation - Adventure - Family	Young princess Anna of Arendelle dreams about finding true love at her sister Elsa's coronation. Fate takes her on a dangerous journey in an attempt to end the eternal winter that has fallen over the kingdom. She's accompanied by ice delivery man Kristoff, his reindeer Sven, and snowman Olaf. On...	0.96	5,440.00	7.30

Nous pouvons constater que les résultats de ce système de recommandation ne sont pas mauvais du tout. En effet, un grand nombre de suggestions sont très sensées compte tenu de notre requête. Pour la demande "princess love" notre système recommande des films tels que Aladdin, La Petite Sirène, Raiponce, La Belle au Bois Dormant, etc.

De plus, cela permet à l'utilisateur d'entrer des mots clés d'intérêt ce qui est moins contraignant que le système précédant qui nécessite d'avoir un film connu en input.

2.4 Implémentation de la méthode SVD

Ici nous appliquons simplement les formules détaillées dans la partie I. Afin de ne pas surcharger le rapport nous nous concentrerons sur les résultats, les détails techniques (et les explications associées) étant disponibles dans le document HTML contenant le Notebook.

movieid	title		rating	genres	
3	293	Leon: The Professional	5.0	Thriller - Crime - Drama	
36	5952	The Lord of the Rings: The Two Towers	5.0	Adventure - Fantasy - Action	
38	7153	The Lord of the Rings: The Return of the King	5.0	Adventure - Fantasy - Action	
6	1204	Lawrence of Arabia	5.0	Adventure - Drama - History - War	
43	8874	Shaun of the Dead	4.5	Horror - Comedy	
8	1259	Stand by Me	4.5	Crime - Drama	
28	2761	The Iron Giant	4.5	Adventure - Animation - Family - Fantasy - Sci...	
10	1285	Heathers	4.5	Thriller - Comedy - Drama	
9	1276	Cool Hand Luke	4.5	Crime - Drama	
7	1250	The Bridge on the River Kwai	4.5	Drama - History - War	
18	2019	Seven Samurai	4.0	Action - Drama	
39	7361	Eternal Sunshine of the Spotless Mind	4.0	Science Fiction - Drama - Romance	
31	3114	Toy Story 2	4.0	Animation - Comedy - Family	
41	8636	Spider-Man 2	4.0	Action - Adventure - Fantasy	
26	2692	Run Lola Run	4.0	Action - Drama - Thriller	
23	2529	Planet of the Apes	4.0	Science Fiction - Adventure - Drama - Action	
20	2174	Beetlejuice	4.0	Fantasy - Comedy	
19	2072	The 'Burbs	4.0	Comedy - Horror - Thriller	
0	111	Taxi Driver	4.0	Crime - Drama	
4	596	Pinocchio	4.0	Animation - Family	

Films déjà vus par l'utilisateur

title		genres
229	Star Wars	Adventure - Action - Science Fiction
946	The Empire Strikes Back	Adventure - Action - Science Fiction
958	Return of the Jedi	Adventure - Action - Science Fiction
317	Forrest Gump	Comedy - Drama - Romance
948	Raiders of the Lost Ark	Adventure - Action
518	Terminator 2: Judgment Day	Action - Thriller - Science Fiction
423	Jurassic Park	Adventure - Science Fiction
100	Braveheart	Action - Drama - History - War
947	The Princess Bride	Adventure - Family - Fantasy - Comedy - Romance
1014	Back to the Future	Adventure - Comedy - Science Fiction - Family

Recommendations

L'image ci-dessous présente les résultats pour un utilisateur test (user n°6) pris au hasard. Le tableau de gauche montre les films déjà vus par cet utilisateur (et les notes et genres associés), tandis que le tableau de droite présente les 10 meilleures recommandations faites par notre système de recommandation.

Ce système de recommandation semble fonctionner assez bien, car nous pouvons voir que le genre de film préféré de l'utilisateur test (Aventure) est bien représenté parmi les films recommandés. De plus, les films recommandés sont assez populaires, ce qui rend très probable le fait que d'autres utilisateurs similaires les aient regardés et bien notés.

3 Conclusion

Nous avons mis en place trois systèmes de recommandation.

Les 2 premiers, qui se basent sur le contenu, utilisent soit des titres de films soit des mots clés, pour des résultats respectivement moyens et plutôt bons. Cependant ces 2 modèles sont plutôt entrée de gamme pourraient être améliorés en utilisant d'autres méthodes d'embedding tel que Word2vec. Word2vec repose sur des réseaux de neurones à deux couches et cherche à apprendre les représentations vectorielles des mots composant un texte, de telle sorte que les mots qui partagent des contextes similaires soient représentés par des vecteurs numériques proches. En bref, Word2vec transforme les verbatims en matrice et permet de prendre en compte le contexte, contrairement à la mesure TF-IDF qui est plus descriptive.

Le 3ème modèle, reposant sur une approche de filtrage collaboratif (en trouvant des users aux goûts similaires) et la Singular Value Decomposition, obtient également des résultats plutôt bons.

Cependant, les modèles utilisés dans l'industrie et auxquels nous sommes confrontés au quotidien sont en réalité bien plus complexes et souvent hybrides, dans la mesure où ils combinent les différentes méthodes que nous avons explorées.

Références

- [1] *A la découverte des systèmes de recommandation*. fr-FR. Mai 2019. URL : <https://www.invivoo.com/blog/systeme-de-recommandation/> (visité le 01/02/2021).
- [2] Nicolas BECHET. “E’tat de l’art sur les Systèmes de Recommandation”. fr. In : (), p. 23.
- [3] Andrea CAPITANELLI. *A mathematical introduction to word2vec model*. en. Sept. 2019. URL : <https://towardsdatascience.com/a-mathematical-introduction-to-word2vec-model-4cf0e8ba2b9> (visité le 02/02/2021).
- [4] *Décomposition en valeurs singulières*. fr. Page Version ID : 179058585. Jan. 2021. URL : https://fr.wikipedia.org/w/index.php?title=D%C3%A9composition_en_valeurs_singuli%C3%A8res&oldid=179058585 (visité le 06/02/2021).
- [5] Yoav GOLDBERG et Omer LEVY. “word2vec Explained : deriving Mikolov et al.’s negative-sampling word-embedding method”. In : *arXiv :1402.3722 [cs, stat]* (fév. 2014). arXiv : 1402.3722. URL : <http://arxiv.org/abs/1402.3722> (visité le 02/02/2021).
- [6] *How retailers can keep up with consumers — McKinsey*. URL : <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> (visité le 01/02/2021).
- [7] *Introduction aux algorithmes de recommandation : l’exemple des articles du blog Octo — OCTO Talks!* URL : <https://blog.octo.com/introduction-aux-algorithmes-de-recommandation-lexemple-des-articles-du-blog-octo/> (visité le 01/02/2021).
- [8] Dr Vaibhav KUMAR. *Singular Value Decomposition (SVD) In Recommender System*. en-US. Mar. 2020. URL : <https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/> (visité le 06/02/2021).
- [9] Tomas MIKOLOV et al. “Distributed Representations of Words and Phrases and their Compositionality”. In : *arXiv :1310.4546 [cs, stat]* (oct. 2013). arXiv : 1310.4546. URL : <http://arxiv.org/abs/1310.4546> (visité le 02/02/2021).
- [10] Tomas MIKOLOV et al. “Efficient Estimation of Word Representations in Vector Space”. In : *arXiv :1301.3781 [cs]* (sept. 2013). arXiv : 1301.3781. URL : <http://arxiv.org/abs/1301.3781> (visité le 02/02/2021).
- [11] Houda OUFIDA et Omar NOUALI. “Le filtrage collaboratif et le web 2.0”. fr. In : *Document numérique* Vol. 11.1 (2008). Publisher : Lavoisier, p. 13-35. ISSN : 1279-5127. URL : <https://www.cairn.info/revue-document-numerique-2008-1-page-13.htm> (visité le 01/02/2021).
- [12] E. PEIS, José MORALES-DEL-CASTILLO et J. DELGADO-LÓPEZ. “Semantic Recommender Systems. Analysis of the state of the topic”. In : *Hipertext.net; Núm. : 6 Edició en anglès* (jan. 2008).
- [13] Abhijit ROY. *Introduction To Recommender Systems- 1 : Content-Based Filtering And Collaborative Filtering*. en. Juil. 2020. URL : <https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421> (visité le 02/02/2021).
- [14] William SCOTT. *TF-IDF for Document Ranking from scratch in python on real world dataset*. en. Mai 2019. URL : <https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089> (visité le 06/02/2021).
- [15] *The backpropagation algorithm for Word2Vec*. URL : <http://www.claudiobellei.com/2018/01/06/backprop-word2vec/index.html> (visité le 02/02/2021).
- [16] *The Movies Dataset*. en. URL : <https://kaggle.com/rounakbanik/the-movies-dataset> (visité le 06/02/2021).