Système de recommandation de livres

Référence	Rapport_Final_LIVRIA
Projet	Projet informatique individuel – LIVRIA
Date remise	25/04/2019

Auteur
Younès Ghennam - 2A
Tuteur
M. Simonazzi Nicolas
Tuteurs soutenance
M. Simonazzi Nicolas
Mme. Clermont Edwige

TABLE DES MATIÈRES

I. Introduction

La technologie émergente de l'Intelligence Artificielle, souvent appelée par son acronyme IA, repose sur des algorithmes relativement complexes, qui ne visent pas seulement à reproduire, à imiter les processus cognitifs humains, à accomplir une tâche proprement humaine et de manière plus efficace, mais aussi de manière plus générale, à imiter tout type d'intelligence réelle. Existant depuis les années 1960, la recherche s'est développée récemment au point de multiplier les applications : voitures autonomes, diagnostics médicaux, assistants personnels, construction de smart city, finance algorithmique, robots industriels, jeux vidéo, ou encore les systèmes de recommandations, pour ne citer qu'une partie des nombreux domaines d'application.

De plus en plus de plateformes reposant sur un système de recommandation ont vu le jour. Elles visent principalement à suggérer ou présenter un contenu susceptible d'intéresser l'utilisateur comme le font Youtube, Spotify, Netflix, Amazon et bien d'autres encore. Néanmoins, nous avons pensé pertinent de concevoir un outil similaire pour les férus de littérature en lui offrant la même visibilité grâce à un système de recommandation de livres nommé LIVRIA, destiné aux petits et grands lecteurs.

I.1. Contexte du projet

Dans le cadre de notre formation au métier d'ingénieur cogniticien à l'ENSC (Ecole Nationale Supérieure de Cognitique), nous avons eu à réaliser un projet tutoré par groupes de 6 étudiants minimum, dont trois en seconde année et au moins le même nombre en première année du cycle d'ingénieur. Ce projet qui était à l'origine notre projet Transpromotion, j'ai décidé de le poursuivre, car je souhaitais bien intégrer les connaissances et les compétences que nous allions découvrir et approfondir en intelligence artificielle et en développement mobile lors de ce second semestre de deuxième année à l'ENSC.

L'idée de ce projet était à l'origine née du constat qu'il serait pertinent de mettre en place un site qui conseillerait des lectures en fonction du profil de l'utilisateur. Néanmoins, j'ai décidé de me tourner plus vers une implémentation d'application mobile pour l'interface utilisateur, car ces dernières sont de plus en plus répandues et je m'imaginais qu'il serait certainement plus pratique d'avoir accès à ce système de recommandation depuis son téléphone portable. Malheureusement, au moment où je rédige ce rapport, la partie interface développée en ReactNative n'est qu'à ses débuts, mais elle n'en restera pas là.

I.2. Pré-existant

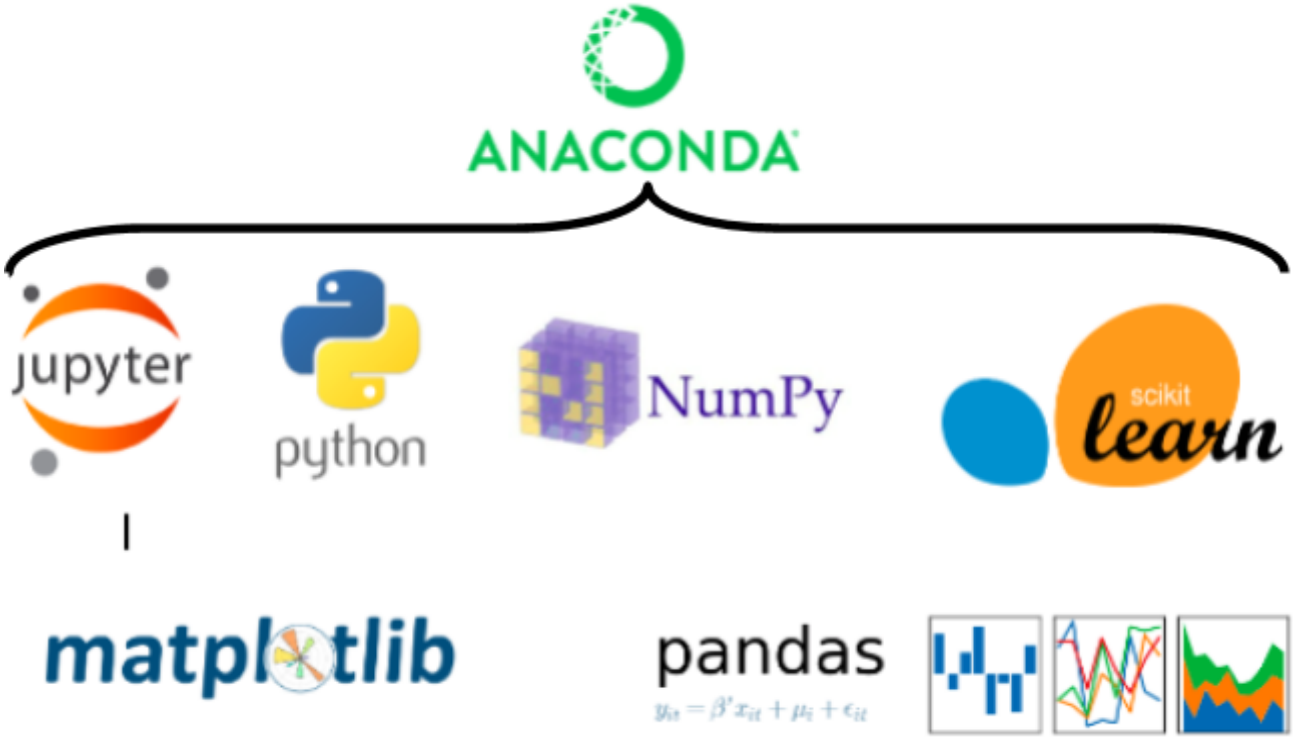
Il existe actuellement un nombre considérable d'applications de recommandation de livres. Elles sont toutes pertinentes dans la mesure où elles fournissent un service assez proche de ce que l'on cherche à offrir. Toutefois, les méthodes utilisées diffèrent de celle que nous comptons mettre en place. En effet, une large majorité de ces applications opte pour des techniques de développement telles que les Tags, les algorithmes basés sur les précédentes lectures de l'utilisateur, ceux basés sur des classements selon le style, le genre, l'auteur, le mouvement littéraire ou encore le thème abordé par le livre. Les classements sont établis en tenant compte de plusieurs critères à savoir la popularité donnée par le nombre de lectures, les notes attribuées par les lecteurs, le nombre d'"envie

de lire ce livre", etc. Certaines applications utilisent un moteur de recommandation externe, tel que celui proposé par Amazon. D'autres approches sont à noter, notamment le "Trending Now", technique qui consiste à donner plus de visibilité aux livres tendances et qui ont le plus de succès.

II. Rappel des spécificités techniques du projet

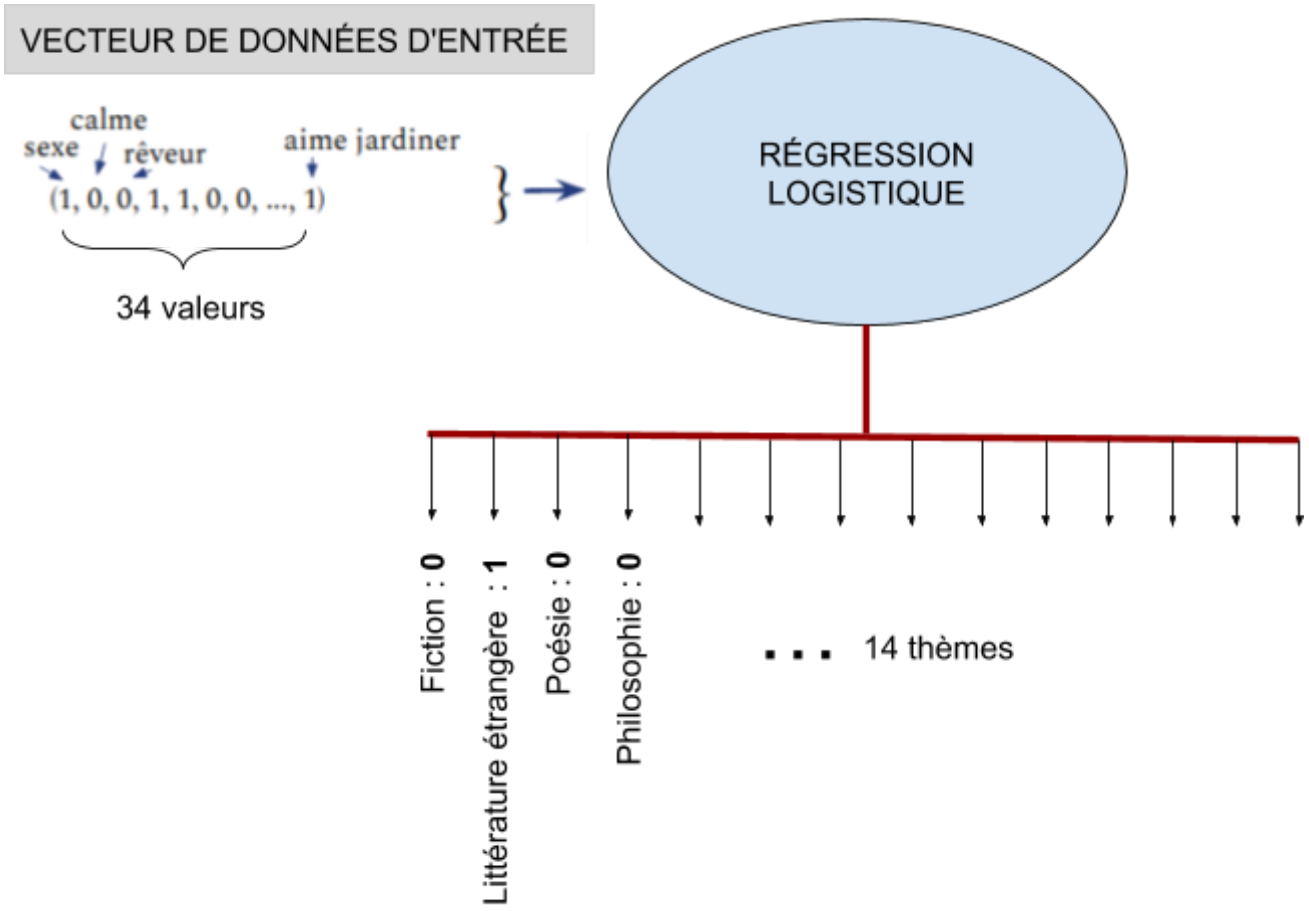
Initialement, l'objectif du projet était de produire une application mobile codée en **React Native** et s'appuyant sur un modèle de Machine Learning implémenté en **Python**.

Il s'avère que j'étais totalement étranger au langage Python, aux différents environnements de développement et aux outils mis à disposition pour développer des applications reposant sur ce qu'on appelle le "Machine Learning". C'est ainsi que j'ai découvert, par la lecture de *Hands on Machine Learning with Scikit-Learn and TensorFlow* - ouvrage que je considère d'une grande qualité, des outils qui deviennent rapidement indispensables pour toutes les personnes travaillant dans ce domaine et qui codent en Python. J'ai découvert **Jupyter** et appris à utiliser ses notebooks comme environnement de travail durant tout ce second semestre pour ce projet. J'ai appris à configurer grâce à **Anaconda** mes propres environnements en affectant à chacun des packages particuliers pour ne pas avoir à en désinstaller et réinstaller systématiquement. Je me suis frotté à l'utilisation des librairies que sont **matplotlib (et seaborn)** pour l'analyse visuelle des données, **numpy** et **pandas** pour la création de matrices, de tableaux et de DataFrame, pour ne citer que les plus récurrentes.



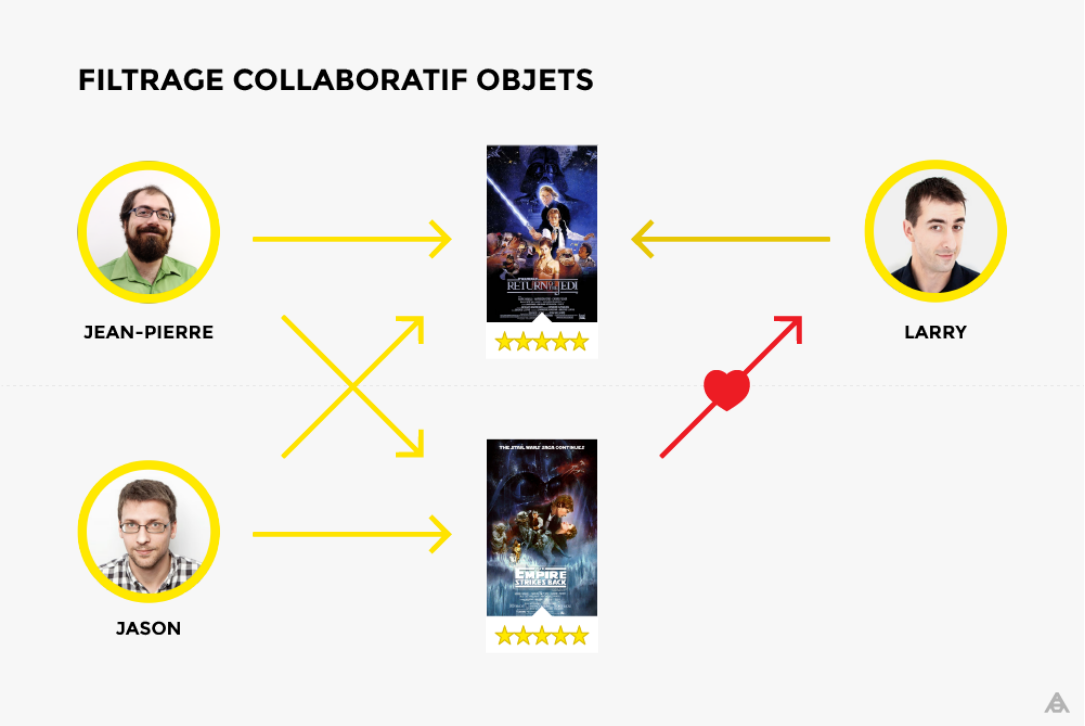
III. Fonctionnement de LIVRIA

Voilà globalement comment fonctionne LIVRIA, notre système de recommandation. Il s'agit d'abord de répondre à des question sur soit comme notre sexe, notre âge, notre tempérament, nos passes-temps favoris, les raisons qui nous font aimer un livre, etc. Via ces questions, nous créons un vecteurs de données comprenant 34 valeurs, vecteur d'entrée à partir duquel notre modèle de prédiction, qui est un modèle de régression logistique, nous donne en sortie un vecteur de 14 valeurs correspondant aux thèmes prédits comme étant ou non susceptibles de nous intéresser. Ensuite, pour faire le lien avec les livres, nous utilisons une base de données (Goodbooks-10k) dont les livres sont caractérisés par 39 tags qui correspondent à différents thèmes dont on retrouve les 14 pouvant être prédits et d'autres encore. Ainsi, nous pouvons vous suggérer des livres en nous basant sur les thèmes susceptibles de vous plaire, puis nous affinons les suggestions en prédisant non plus des thèmes mais des livres directement en utilisant ce qu'on appel le filtrage collaboratif. Nous nous appuyons sur les notes que vous attribuez aux livres pour trouver des utilisateurs similaires et donc des livres qu'ils ont lus et vous non, ou tout simplement en regardant la similarité entre les livres eux-mêmes.

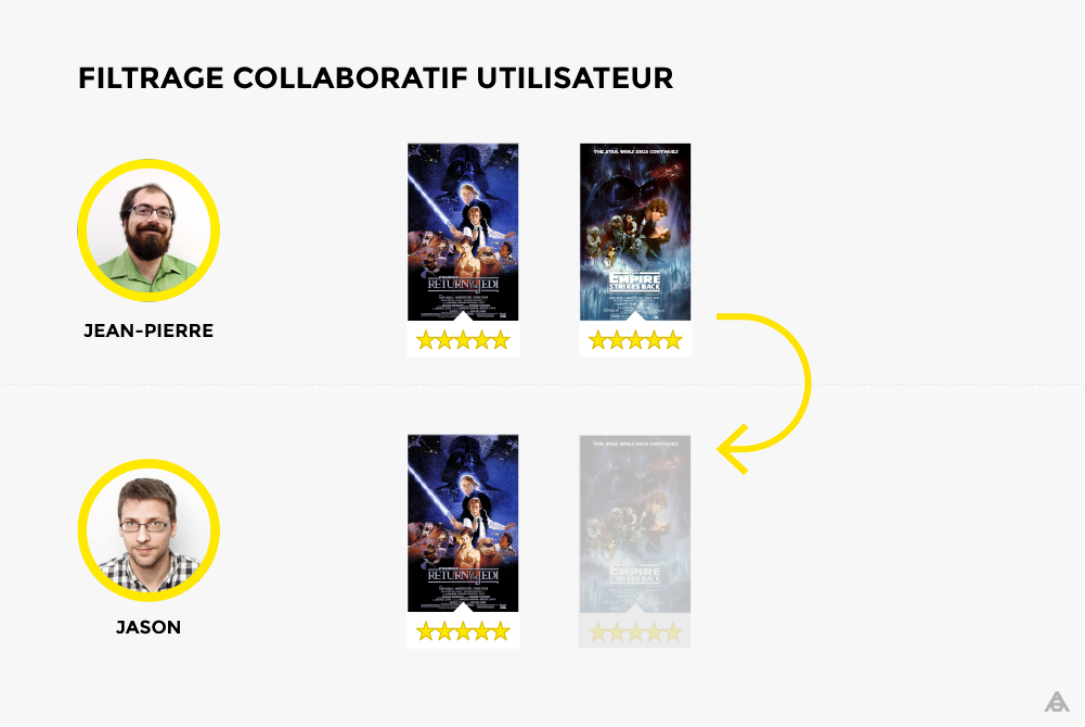


Fonctionnement de LIVRIA pour les prédictions de thèmes

Voici globalement comment fonctionne le filtrage collaboratif :



Filtrage collaboratif objet



Filtrage collaboratif utilisateur

Liste des livres qui pourraient vous plaire
==========

The Hunger Games de Suzanne Collins (2008). Note moyenne: 4.34/5

To Kill a Mockingbird de Harper Lee (1960). Note moyenne: 4.25/5

The Fault in Our Stars de John Green (2012). Note moyenne: 4.26/5

Mise en forme des suggestions

IV. Réalisations et lecture des notebooks

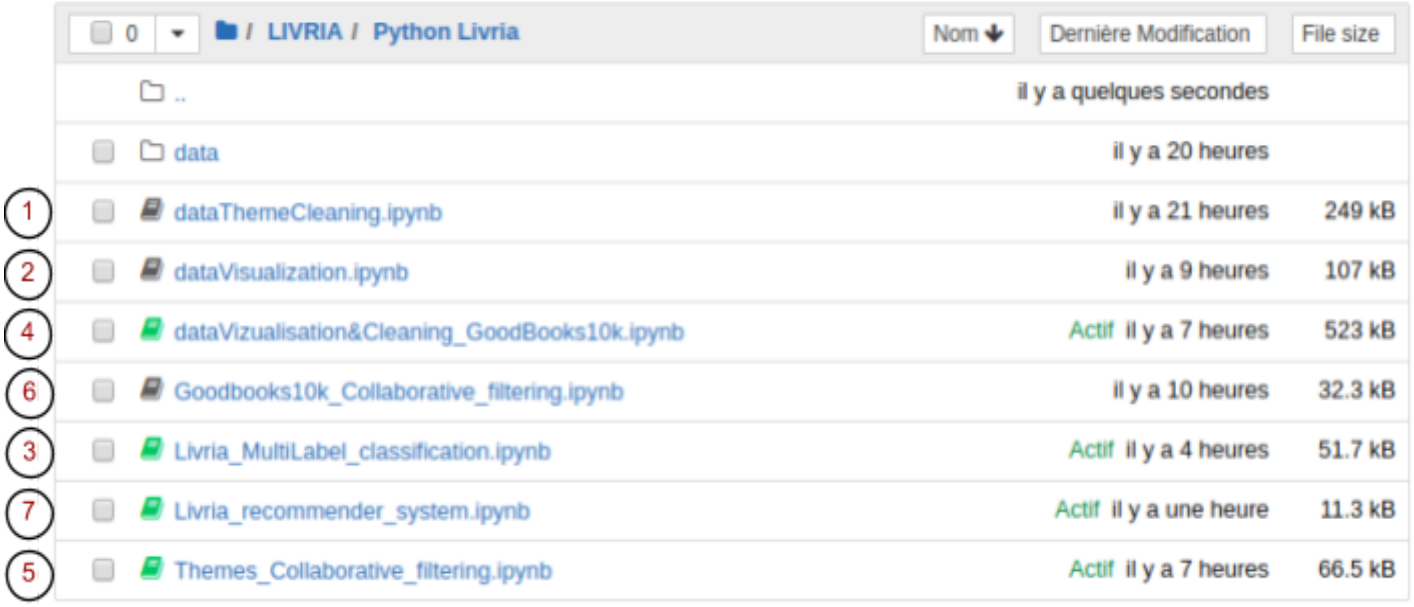
Rappelons que ce projet et a été commencé dans le cadre du projet transposition au premier semestre de cette année. Tous les livrables concernant la conception orientée utilisateur (CCU) ont été réalisés à ce moment, et ces derniers comprennent :

- Réalisation d'un questionnaire, recueil et analyse des données,
- Personna,
- Storyboard,
- Wireframe et tests utilisateurs sur ce dernier,
- Analyse basée sur les critères de Nielsen.

A savoir qu'il faudra certainement reprendre certaines de ces réalisation et les adapter aux nouvelles contraintes, car le rendu ne pourra pas être le même sur un téléphone que sur un écran de PC. Néanmoins, j'ai laissé cette partie CCU pour me focaliser sur le côté plus technique de l'application, la partie IA étant la pierre angulaire du projet.

Vous pourrez retrouver l'ensemble de mon travail dans le dépôt Git au lien suivant : https://github.com/Younzer/LIVRIA

Ce dernier est composé d'une partie interface (cf. dossier **Interface_ReactNative**) dont j'ai commencé ce qu'on pourrait considérer comme une petite ébauche, j'avais profité des enseignements en développement mobile pour commencer cette partie. D'autre part, vous retrouverez ce qui compose le plus gros de mon travail que j'ai prioritairement axé sur la partie intelligence artificielle (cf. dossier **Python Livria**). Il comprend l'ensemble des notebooks Jupyter qui permettent de prédire les thèmes susceptibles de plaire à un utilisateur en fonction de ses "critères", puis les livres en fonction des notes qu'il leur attribue. Ces notebooks sont je pense vraiment bien commentés, je cherche à vous guider comme moi même j'aurais souhaité l'être lorsque je faisais mes premiers pas dans leur implémentation. C'est pourquoi je ne m'attarderai pas sur leur contenu que je vous invite à découvrir (cf. lien du dépôt git en **Annexe ou ci-dessus**). Je vous indique cependant le "sens de lecture" de ces notebook, qui rend compte de la méthodologie que j'ai mise en place pour réaliser ce travail :



Répertoire des fichiers .ipynb

Je vous conseille de commencer par lire le fichier **dataThemeCleaning (1)**. C'est dans ce notebook que j'ai nettoyé et vectorisé les données récoltées via le questionnaire pour pouvoir permettre leur utilisation pour la prédiction des thèmes avec les différentes fonctions de Scikit-Learn. Après avoir nettoyé ces données, j'ai enregistré deux dataFrames au format csv dans le sous-dossier ./data :

- **df_entree**, qui correspond au vecteur de données des caractéristique d'entrée pour la prédiction des thèmes,
- et **df_sortie**, qui correspondent au vecteur de données sur les 14 thèmes de sortie.

À savoir que ces vecteurs ne sont composés que de **1** si cette valeur est en entrée ou en sortie, ou de **0** s'il elle n'y est pas. Ainsi, si l'utilisateur qui a indiqué qu'il était calme mais n'a pas indiqué qu'il était autoritaire aura son vecteur d'entrée avec un 1 pour la valeur calme et 0 pour la valeur autoritaire. Si le modèle de régression logistique que nous avons mis en place prédit que les romans et fictions pourraient lui plaire, ces thèmes seront caractérisés par un 1 en sortie et un 0 sinon.

Les vecteurs d'entrée sont composés des valeurs suivantes :

```
df_entree = {
       'Sexe': sexe, # sexe
       'Calme' : calme, # personnalité
       'Intellectuel' : intellectuel,
       'Aventurier' : aventurier,
       'Agite' : agite,
       'Sociable' : sociable,
       'Introverti' : introverti,
       'Altruiste' : altruiste,
       'Creatif' : creatif,
       'Reserve' : reserve,
       'Amusant' : amusant,
       'Ambitieux' : ambitieux,
       'Autoritaire' : autoritaire,
       'Jaloux' : jaloux,
       'Consciencieux' : consciencieux,
       'Curieux' : curieux,
       'Geek' : geek,
       'Sportif' : sportif,
       'Pantouflard' : pantouflard,
       'Esprit' : esprit, # passes-temps
       'Sport' : sport,
       'Dessin' : dessin,
       'Rien faire' : neRienFaire,
       'Jeux videos' : jeuxvideos,
       'Cuisine' : cuisine,
       'Theatre' : theatre,
       'Meditation' : meditation, # attentes du lecteur
       'Voyage' : faitvoyage,
       'FacileLire' : facilelire,
       'Reflechir' : reflechir,
       'Connaissance' : connaissance,
       'Personnage' : personnage,
       'Tout' : tout,
       'Style' : style
        }
```

Les vecteurs de sortie sont composés des valeurs suivantes :

```
df_sortie = {'RomanFiction' : romanFiction, # themes
       'BdComics' : bdComics,
       'ArtsCulture' : artsCulture,
       'DocMedia' : docMedia,
       'Erotisme' : erotisme,
       'Esoterisme' : esoterisme,
       'SanteBE' : santeBE,
       'HistGeo' : histGeo,
       'Jeunesse' : jeunesse,
       'LittEtrangere' : littEtrangere,
       'ScienceTechnique' : scienceTechnique,
       'LoisirVie' : loisirVie,
       'SHS' : shs,
       'Philosophie' : philo
        }
```

Ces nettoyage et vectorisation des données étaient nécessaires pour entraîner notre modèle de prédiction, mais ils ont aussi permis de mieux visualiser les données, et cela correspond à la deuxième étape dans notre méthodologie, et donc dans la lecture des notebooks. Je vous invite donc à lire le notebook. **dataVisualization (2)**.

Ce second notebook permet d'avoir une bien meilleure appréhension des données que nous avons récoltées. Nous y créons différents graphiques, des histogrammes, des boxplots et des figures interactives afin d'analyser visuellement et statistiquement les données. Une fois les données nettoyées et analysées, nous pouvons les exploiter pour la prédiction des thèmes. C'est ce que nous faison dans le notebook suivant : **Livria_MultiLabel_Classification (3)**. Nous y avons implémenté différents modèles de prédiction et mesuré les performances afin de les comparer et garder celui dont les prédictions sont les meilleures. Il s'avère que c'est le premier modèle que nous implémentons qui est le plus performant, il s'agit d'un modèle de régression logistique. Nous détaillons le choix de ce modèle et son utilisation dans la partie IV. Problématiques rencontrées.

Le notebook suivant, nommé **dataVizualisation&Cleaning_Goddbooks10k (4)**, correspond au nettoyage et à la visualisation des données du set de **Goodbooks-10k**. Ce set comprend des informations sur **10000 livres** notés par environ **50000 utilisateurs** pour un total de **6 millions de notes**. Il était nécessaire que nous choisissions uniquement une partie de ces livres car la base de données entière est trop importante pour être manipulée danq les différents calculs nécessaires à la prédiction sur Jupyter. Pour faire cette sélection, il fallait d'abord visualiser les données et s'assurer que l'échantillon prélevé soit représentatif du set entier pour ne pas avoir un effet de biais dû à la sélection. Une fois cette sélection faite, nous enregistrions les livres restants et les données relatives dont nous avions besoin dans un dataFrame nommé fd_notes au format csv dans le sous-dossier ./data.

Pour récapituler où nous en somme actuellement, nous avons nettoyé et vectorisé les données récoltées par le questionnaire, données permettant ce qui sera le point d'entrée de notre application : la prédiction des thèmes en fonction de vos "caractéristiques". Nous les avons visualisées et utilisées pour entraîner différents modèles de prédictions dont nous avons comparé les résultats pour ne garder que le plus performant, qui était ici le modèle de **régression logistique**. Nous sommes donc capables de prédire les thèmes mais il faut encore pouvoir afficher des livres correspondant à ces thèmes. C'est là que nous utilisons la base de données de Goodbooks-10k - que nous avons visualisée et dont on a prélevé un échantillon de **1000 livres** - pour faire le lien entre ces thèmes prédits et les livres marqués par des tags, car oui, Goodbooks-10k avait cet avantage important de réunir les données sur les tags assignés aux livres de sa base. Bien, la lecture des notebooks se poursuit dans l'ordre qui suit : **Themes_Collaborative_filtering (5)** et **Goodbooks10k_Collaborative_filtering (6)**, qui composent la deuxième partie des prédictions réalisées par notre IA.

Nous mettons en oeuvre dans ces deux avant-derniers notebooks différentes méthodes de filtrage collaboratif afin d'affiner la recommandation en se basant sur les notes que vous attribuées aux livres. Nous avons donc maintenant tous les outils nécessaires pour faire nos prédictions et rendre le résultat sous une forme visuelle plus agréable que de simple dataFrame. C'est ce que nous faisons dans **Livria_Recommender_system (7)** qui correspond à notre dernier notebook.

V. Problématiques rencontrées

Durant ce projet, j'ai rencontré de nombreuses difficultés qu'il m'a fallu résoudre ou contourner d'une manière ou d'une autre. L'idée d'origine était de **recommander des livres** en fonction des critères d'entrée que nous avions choisis de poser dans le questionnaire réalisé lors du transpromotion. Mais comment recommander des livres avec une base de données ne regroupant que si peu de livres et où chaque individu ayant répondu au questionnaire en recommande un différent par rapport au autres. Ce fût la **première problématique** rencontrée. Les avis ne se recoupent pas sur les livres mentionnés dans les réponses au questionnaire, et nous manquons de données pour prédire une liste de livres en particulier. L'idée était alors non pas prédire des livres précis en se basant sur notre base de données, mais de **prédire plutôt des thèmes de livre**, un thème étant plus global qu'un livre précis, cela favorise de meilleures prédictions.

Mais alors, comment faire le lien entre les thèmes prédits et des livres ayant ce ou ces thèmes ? Ce fût notre **seconde problématique**. L'idée était alors de récupérer des données de livres via une API et j'ai donc réalisé un important travail de recherche sur la manière d'extraire les données relatives aux ouvrages que l'on souhaite recommander. Malheureusement, il n'existe aucune API assez complète pour faire le lien entre les thèmes prédits et les livres, car aucune ne catégorisent les livres par thèmes. J'ai songé à utiliser des outils de Web Scraping pour récolter ces données car certains sites répertorient les livres par thème(s) mais ne proposaient pas d'API. Ce qui m'a fait sortir de l'impasse, c'est l'aide que m'a apportée M. Simonazzi, mon tuteur de projet, qui m'a conseillé de m'inspirer sur ce que fait Amazon en terme de recommandation et m'a envoyé un notebook sur le filtrage collaboratif effectué sur une base de données de films (cf. MovieLens). Il s'avère que le site sur lequel on peut retrouver la base de données MovieLens, nous pouvions en retrouver une autre avec des livres. Cette base de données s'appelle Book-Crossing, elle date de 2004, et en cherchant une plus récente, je suis tombé sur Goodbooks-10k qui elle contient notamment de nombreuses notes - 6 millions - attribuées par un peu plus de 50000 utilisateurs sur 10000 livres. Il était donc possible de faire du filtrage collaboratif sur cette base de données, qui procurait aussi une liste de tags permettant de faire le lien entre les thèmes prédits et les livres à suggérer.

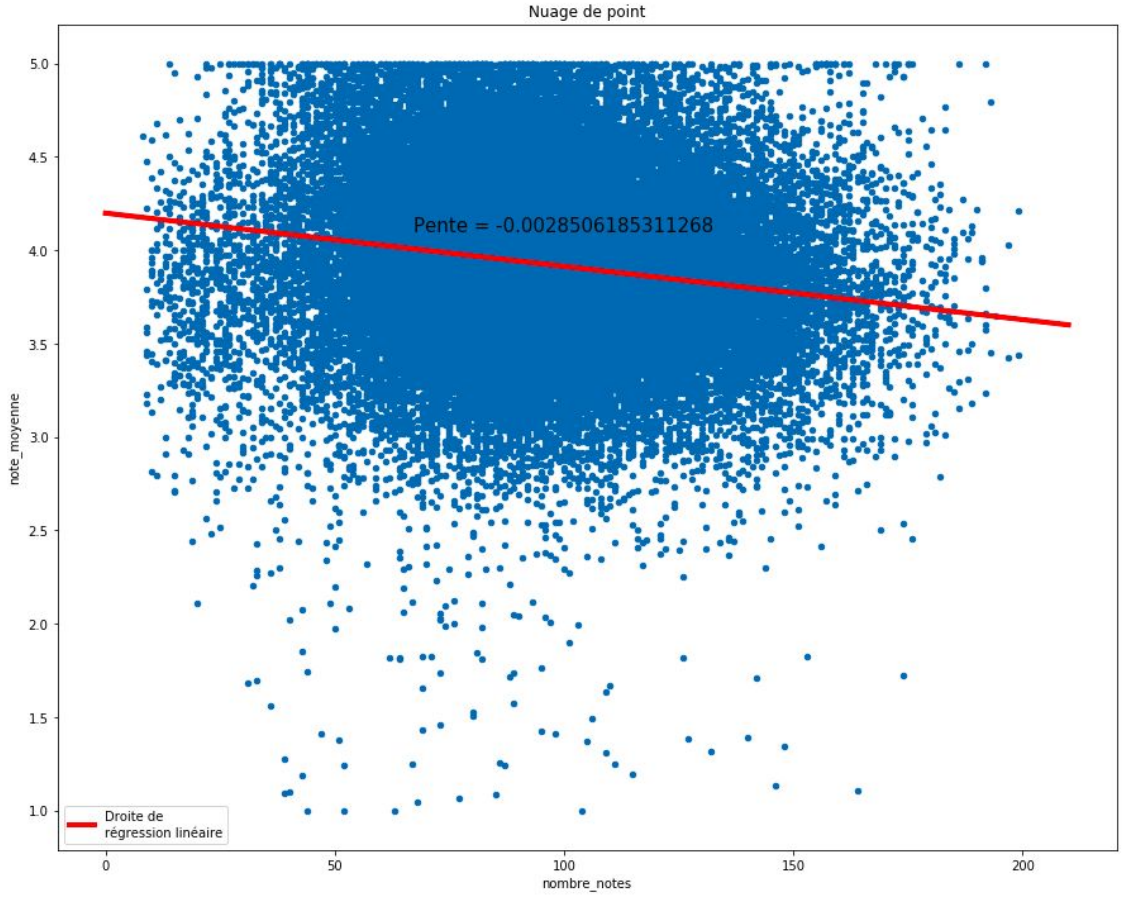
Néanmoins, de nombreux tags ne correspondaient pas à des thèmes et il y en avait de nombreux que nous ne souhaitions pas utiliser. C'est pourquoi il a fallu faire une sélection des tags à préserver. L'image qui suit vous montre le choix des tags correspondant à des thèmes que j'ai fait sur le set de données de Goodbooks-10k. Vous pouvez retrouver le reste des sélections de tag en **Annexe**. Les tags sont classés en fonction du nombre de fois où ils ont été utilisés, et ce dans l'ordre décroissant. J'ai ai mis en valeur dans les petits encadrés rouges les tags choisis. Vous pouvez les compter comme moi, ils sont au nombre de 39, ce qui permet de recouvrir bien plus de thèmes que ceux que nous proposions dans le questionnaire qui étaient uniquement au nombre de 14.

tag_name		tag_name		tag_name		tag_name	
biography-autobiography	447	life	412	women-writers	373	romance-contemporary	344
magical-realism	443	graphic	411	new-york	370	summer-reading	343
movie	442	to-read-fantasy	409	psychological	370	sub	338
standalone	441	stories	408	womens-fiction	368	book	337
culture	440	religious	406	1001-import	368	science-fiction-and-fantasy	336
steamy	439	finished-series	405	fantasy-series	367	brit-lit	336
cultural	439	read-as-a-kid	404	translation	364	erotic-romance	335
women-s-fiction	438	adult-non-fiction	403	children-young-adult	364	horror-thriller	335
on-my-kindle	438	chapter-books	403	completed-series	364	children-s-book	334
art	436	biographies-memoirs	402	fiction-general	363	ghosts	333
2013-books	433	crime-mystery-thriller	400	mystery-series	361	2013-read	332
home	430	abuse	396	demons	360	comics-and-graphic-novels	331
signed	429	1990s	396	mysteries-thrillers	360	2017-reading-challenge	329
2014-read	428	2017-books	396	personal-development	359	my-collection	329
book-boyfriend	425	m-f	394	to-read-owned	357	english-literature	329
comic	425	vamps	394	christian	357	fluff	328
sf-f	423	romantic	388	ya-paranormal	357	series-to-read	328
love-triangle	422	elementary	386	2011-reads	355	could-not-finish	327
my-childhood	422	2000s	386	not-read	355	fantasy-science-fiction	327
childrens-fiction	422	picture-books	386	mystery-detective	354	dark-fantasy	327
1001-books-to-read	421	adult-nonfiction	385	comics-graphic-novels	354	roman	327
to-reread	419	post-apocalyptic	385	satire	354	romantic-suspense	327
witches	418	already-read	383	short-story	353	loved-it	327
classic-lit	418	sociology	383	read-in-2008	351	books-read-in-2016	326
bio-memoir	416	young-adult-fantasy	381	graphic-novels-comics	349	comics-manga	325
erotic	416	favorite-author	380	read-alouds	349	lgbt	325
essays	415	business	377	2005	349	american-history	325
self-improvement	413	poetry	377	espionage	348	comic-books	325
trilogy	412	fiction-fantasy	375	family-relationships	345	women-authors	324

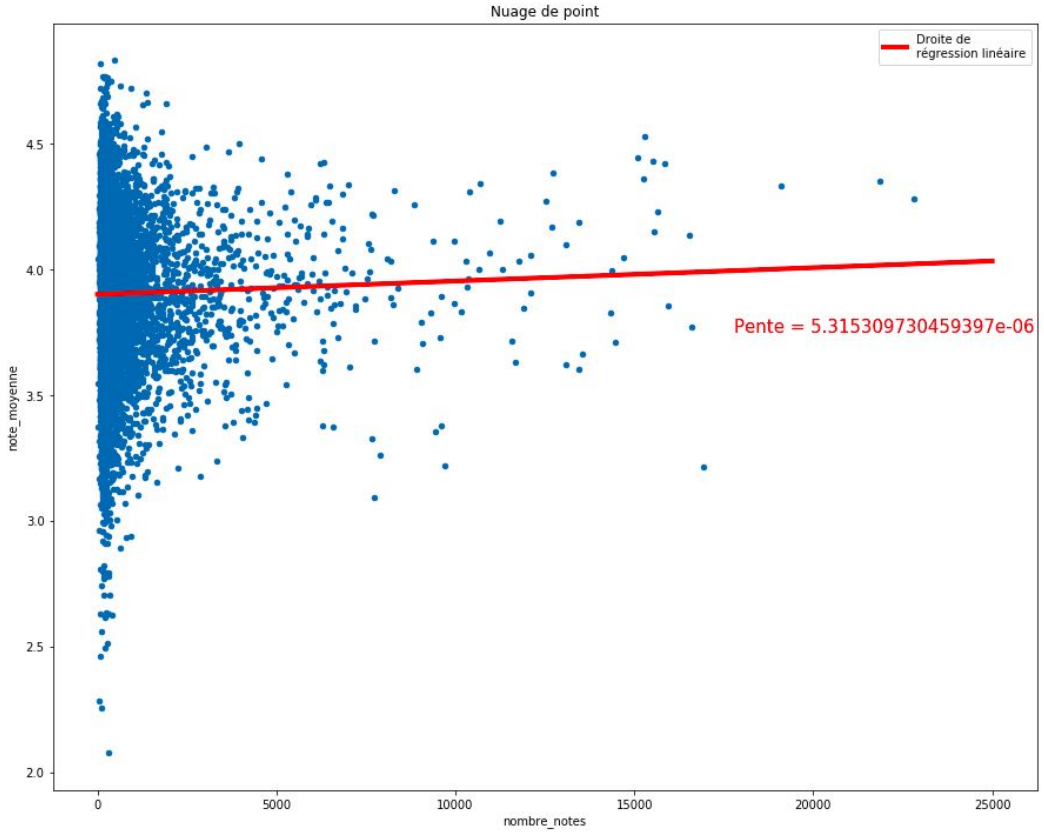
Sélection des tags de GoodBooks-10k

La problématique suivante tenait du fait qu'il était impossible de faire du filtrage collaboratif en se basant sur l'ensemble de cette base de données. Même après sélection des livres à partir des tags, seulement un livre avait été enlevé - ce qui était bon signe car nous avions choisi des thèmes assez différents pour recouvrir la quasie-totalité de ces derniers. Dans le notebook **Goodbooks10k_Collaborative_filtering (6)**, lorsque je tentais de créer une matrice avec en colonne chaque livre et en abscisse chaque utilisateur, avec à l'intérieur toutes les valeurs correspondant aux notes que chaque utilisateur a attribuées aux livres qu'il a lu, une erreur de type Mémoire m'indiquait qu'il était impossible de créer une telle matrice car la mémoire vive n'était pas assez importante. Il fallait donc garder seulement une partie des données pour faire le filtrage collaboratif. Mais comment choisir ces données ?

Il était possible de tout simplement choisir au hasard des livres parmis cette base, mais l'idée qui m'est venue en tête était la suivant : pour faire du filtrage collaboratif, plus on a des données qui se recoupent sur les mêmes livres, plus ce filtrage sera précis. Il fallait donc essayer d'avoir la matrice la plus complète possible, c'est-à-dire avec le plus de notes dedans, avec le moins de 0 possible. Il fallait donc choisir les livres non pas au hasard, non pas les mieux notés, mais les plus notés, et choisir les utilisateurs qui ont attribué le plus de notes. C'est ce que j'ai fait en m'assurant que l'échantillon restait représentatif en terme de notes attribuées, et c'est ce que nous pouvons voir dans les graphiques qui suivent :



Note moyenne en fonction du nombre de notes attribuées par chaque utilisateur



Note moyenne en fonction du nombre de notes attribuées à chaque livre

On peut voir que la corrélation entre le nombre de notes attribuées et la valeur moyenne de ces notes est quasiment nulle (cf. valeur des pentes des droites de régression linéaire). Ainsi, nous pouvions prendre les livres les plus notés et les utilisateurs ayant donné le plus de notes sans craindre que notre échantillon ne soit pas représentatif du set global en terme de notes attribuées.

La **dernière problématique** rencontrée fût celle concernant l'implémentation du modèle de prédiction des thèmes. Dans les différentes ressources que j'avais pu réunir, notamment le livre *Hands on Machine Learning with Scikit-learn and TensorFlow*, je n'arrivais pas à trouver un outil me permettant de prédire en sortie un vecteur de données composé de 14 valeurs, des 0 et des 1, pour dire pour chaque thème si oui ou non il est susceptible de plaire à l'utilisateur. J'étais au départ persuadé qu'il s'agissait d'un problème de régression logistique, mais en réalité la régression logistique ne prédit qu'une seule valeur de sortie, un seul 0 ou un seul 1. Mon problème était en réalité un problème de classification multi-label (plusieurs valeurs à prédire en sortie). Heureusement, une fois que je me suis rendu compte de cette nuance pour le moins importante, j'ai rapidement pu trouver les outils me permettant de faire de la prédiction sur une liste de valeurs que sont les thèmes, que l'on peut considérer comme des labels. Comble du sort, c'est une méthode permettant d'utiliser la **régression logistique** en transformant notre problème de classification multi-label en un problème de classification multi-class qui a fait ses preuves face aux autres modèles. Ses prédiction étaient les meilleures.

VI. Conclusion

Mon travail sur ce projet m'a fait prendre conscience de certaines des difficultés que l'on peut rencontrer lorsque l'on souhaite implémenter un modèle de prédiction, et plus en général, lorsque l'on souhaite réaliser travailler dans un domaine comme l'IA et le Machine Learning. D'une part, j'ai appris qu'il n'y a pas de solution qui soit la même pour tous les problèmes où l'on emploie des méthodes d'intelligence artificielle. Chaque problème a ses **spécificités**, repose sur des **données uniques** qu'il faudra savoir nettoyer, traiter et analyser de manière tout aussi spécifique.

J'ai eu la chance d'être confronté à des difficultés qui relèvent de ce qu'on appel les **problématiques métier**, c'est à dire des problématiques sur les stratégies à adopter pour chapoter de gros projets dans des logiques de développement et de production. Ainsi, si je n'en ai pas utilisé dans mon projet, je sais néanmoins qu'il existe des moyens pour entraîner un modèle de prédiction petit à petit lorsque les données sont trop importantes - en terme de taille pour la mémoire - pour être utilisées entièrement d'emblée. Il est aussi important d'entraîner différents modèles pour les comparer entre eux, car il n'existe pas un modèle meilleur que les autres systématiquement pour une type de problème donné.

Ce projet m'a appris qu'il est important d'adopter une **méthodologie** qui soit propre à notre problème, pour que la solution soit la plus adaptée et réponde le mieux à nos besoins. Cette méthodologie est essentielle pour sa propre compréhension du problème et donc des cheminements possibles vers une solution. C'est en associant régression logistique pour prédire les thèmes et filtrage collaboratif que nous parvenons à suggérer des livres. Cette solution peut être vue comme une combinaison qui aurait pu être différente dans notre cas mais qui a l'avantage d'être fonctionnelle et performante, et il est important de comprendre que cette solution ne serait peut être pas adaptée à d'autre systèmes de recommandation.

Pour ce qui est du futur, je compte continuer **LIVRIA** et faire la partie interface en React-Native puis certainement un site web en React pour avoir un projet complet que je pourrai mentionner dans mon CV. Evidemment je vois plein de pistes d'amélioration pour ce projet, mais si je ne parle que de la partie IA, je pense qu'il serait pertinent de refaire une sélection des livres en prenant en compte pas seulement le nombre de notes mais aussi les thèmes associés pour avoir un un échantillon assez représentatif en terme de thème de livre. Je n'ai pas vérifié si les livres choisis représentent toujours les 39 tags que j'avais sélectionnés. Je pourrai aussi tenter de mettre en place un modèle d'**apprentissage en ligne**, car il pourra continuer à être alimenté au fur et à mesure que les utilisateurs attribuent des notes aux livres. Il faudrait aussi que je récupère la description des livres et que je cherche une base de données correspondant plus aux lectures que nous faisons en France, car celle utilisée dans notre cas représente bien plus les lectures de nos amis anglophones et les titres des livres sont pour l'instant tous en anglais.

Annexe

Choix des tags gardés :

tag_name		tag_name		tag_name		tag_name		tag_name		tag_name	
to-read	9983	books	4818	classics	2785	calibre	2086	part-of-a-series	1576	science-fiction-fantasy	1257
favorites	9881	adult-fiction	4775	read-2016	2760	crime	2083	books-i-have	1568	science	1239
owned	9858	e-books	4616	read-2014	2756	didn-t-finish	2031	ya-fiction	1561	friendship	1238
books-i-own	9799	read-in-2013	4384	contemporary-fiction	2696	to-read-fiction	2003	mystery-suspense	1560	on-hold	1225
currently-reading	9776	book-club	4365	favorite-books	2683	fantasy-sci-fi	1991	kindle-books	1558	listened-to	1224
library	9415	audible	4346	dnf	2675	nook	1987	own-to-read	1543	children-s-books	1187
owned-books	9221	fantasy	4259	finished	2639	library-book	1977	read-2012	1543	2014-reads	1186
fiction	9097	romance	4251	read-in-2011	2639	chick-lit	1974	mysteries	1467	general	1182
to-buy	8692	audio-books	4220	read-in-2017	2603	20th-century	1964	must-read	1459	ya-books	1180
kindle	8316	abandoned	4168	5-stars	2599	paranormal	1941	need-to-buy	1457	my-favorites	1177
default	8239	novel	4084	historical-fiction	2590	school	1940	urban-fantasy	1427	scifi	1169
ebook	8054	re-read	3903	paperback	2566	classic	1939	childhood	1418	action-adventure	1145
my-books	7561	have	3846	historical	2544	magic	1898	read-in-english	1397	humour	1130
audiobook	7242	audio-book	3776	thriller	2522	mystery-thriller	1862	children	1383	biography	1109
ebooks	7203	mystery	3686	sci-fi-fantasy	2467	teen	1847	literary	1378	speculative-fiction	1106
wish-list	7192	borrowed	3674	american	2463	supernatural	1843	horror	1372	contemporary-romance	1105
my-library	7000	read-in-2012	3661	suspense	2419	recommended	1835	reread	1357	high-school	1101
audiobooks	6862	adventure	3661	reviewed	2399	nonfiction	1833	childrens	1354	to-read-non-fiction	1070
i-own	6670	young-adult	3630	4-stars	2383	favorite-authors	1824	young-adult-fiction	1354	women	1062
adult	6604	english	3625	unfinished	2361	realistic-fiction	1813	thrillers	1353	philosophy	1055
audio	6548	did-not-finish	3420	read-2013	2347	literary-fiction	1776	read-2017	1343	3-stars	1046
favourites	6422	favorite	3363	home-library	2315	bookclub	1766	2015-reads	1318	war	1044
novels	5665	maybe	3243	library-books	2299	want-to-read	1752	on-my-shelf	1309	fantasy-scifi	1041
own-it	5514	shelfari-favorites	3127	sci-fi	2227	read-in-2010	1731	british	1305	mystery-crime	1041
contemporary	5287	drama	3046	science-fiction	2222	tbr	1715	2016-reads	1269	usa	1036
read-in-2015	5239	literature	3005	action	2172	unread	1688	kids	1267	shelfari-wishlist	1033
series	5080	general-fiction	2974	humor	2161	funny	1675	children-s	1267	comedy	1031
e-book	4941	read-2015	2891	family	2160	bookshelf	1643	coming-of-age	1267	first-in-series	1031
read-in-2016	4940	ya	2842	history	2138	scifi-fantasy	1620	on-kindle	1262	juvenile	1031
read-in-2014	4902										

tag_name		tag_name		tag_name		tag_name		tag_name	
stand-alone	1016	american-literature	885	1	612	vampire	557	want-to-buy	493
childrens-books	1013	children-s-literature	882	in-my-library	611	2017-reads	557	graphic-novels	491
england	1013	children-s-lit	869	british-literature	610	read-more-than-once	557	19th-century	490
ya-fantasy	1012	want	865	reference	608	erotica	557	purchased	490
crime-mystery	1011	ya-lit	846	epic	603	children-books	556	books-owned	488
kids-books	1008	overdrive	841	5-star	602	bought	552	2015-read	487
detective	999	paranormal-romance	840	2014-books	599	to-read-own	550	movies	485
female-author	997	to-re-read	839	childhood-reads	599	loved	548	kid-books	485
book-club-books	981	murder	829	new-adult	597	american-lit	548	fantasy-fiction	484
personal-library	974	arc	824	collection	597	2006	543	graphic-novel	484
my-bookshelf	970	dark	824	books-to-buy	595	owned-to-read	539	read-aloud	482
2013-reads	969	psychology	810	memoir-biography	592	realistic	536	ya-romance	479
book-group	958	to-read-nonfiction	806	death	591	1001-books	535	bio	478
other	954	politics	793	alpha-male	588	chic-lit	528	spiritual	477
crime-fiction	954	europe	793	classics-to-read	587	classroom-library	525	scanned	475
youth	932	sff	777	2012-reads	586	1001	524	netgalley	474
faves	926	middle-school	773	school-books	582	mystery-suspense-thriller	524	classic-fiction	474
read-2011	924	sci-fi-and-fantasy	771	self-help	579	1001-books-to-read-before-you-die	519	modern-classics	473
modern-fiction	924	2016-books	769	book-boyfriends	575	adult-romance	519	comics	469
relationships	922	murder-mystery	767	animals	574	pnr	519	my-ebooks	469
childhood-books	920	high-fantasy	765	mythology	572	made-me-cry	513	political	468
couldn-t-finish	917	hardcover	762	children-s-fiction	568	childrens-literature	509	survival	464
read-in-2009	913	read-for-school	762	book-club-reads	567	modern	508	biographical	462
middle-grade	913	dystopian	755	on-my-bookshelf	566	2015-reading-challenge	507	meh	460
memoir	905	sf	755	chicklit	563	spirituality	503	travel	457
gave-up-on	900	biography-memoir	752	to-get	560	fiction-to-read	500	military	455
religion	893	classic-literature	752	2016-reading-challenge	559	translated	497	1001-books-you-must-read-before-you	453
mine	891	memoirs	744	2016-read	559	mystery-thriller-suspense	496	i-own-it	453
crime-thriller	890	teen-fiction	743	uk	558	gave-up	495	read-2010	451

Code source :

https://github.com/Younzer/LIVRIA

Ressources bibliographiques :

Hands-On Machine Learning with Scikit-Learn and TensorFlow
Concepts, Tools, and Techniques to Build Intelligent Systems
By Aurélien Géron

Liens :

Pour la maîtrise des librairies
http://python-simple.com/python-matplotlib/matplotlib-intro.php
Base de données Goodbooks-10k
http://fastml.com/goodbooks-10k-a-new-dataset-for-book-recommendations/
Résolution du problème de classification multi-label
https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff
Filtrage collaboratif
https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0