

ECOLE SIMPLON MICROSOFT



Recommandations de BD

Curbiere Loïc

Titre professionnel "Développeur en intelligence artificielle" de niveau 6 enregistré au RNCP sous le
n°34757 2 mars 2022

Table des matières

1 Contexte	3
2 D'où proviennent les données	3
3 Un planning qui a su évoluer	4
3.1 En tout début de projet	4
3.2 Mise en place du projet	6
4 Une base données analytiques	8
4.1 Analyse Exploratoire	9
4.1.1 Scrapping des données	9
4.1.2 Valeur nulle	9
4.1.3 Quel type de BD est-ce donc ?	9
4.1.4 De la moins bonne BD à la meilleure	10
4.1.5 Les mots qui sont les plus utilisés dans les titres	11
4.1.6 Les auteurs les plus prolifiques	11
4.2 Au niveau utilisateur	11
4.2.1 Nombre d'utilisateurs	11
4.2.2 Les mot qui reviennent le plus dans les commentaires	12
4.2.3 Les utilisateurs les plus critiques	12
5 Qui y a-t'il sous le capot	14
5.1 Petits rappels sur les moteurs de recommandations	14
5.2 Rappel rapide sur les Transformers	15
6 Le modèle	16
6.1 Mon modèle	16
6.2 Utilisation de BERT	17
6.3 Une aide pour remplir sa bédéthèque	17
6.4 Comment marche la cosine	18
6.5 Premiers résultats	18
6.6 Évaluation du modèle	19
7 Pour les besoins du projet : mise en place d'une base de données relationnelle	22
7.1 Explicitation de la BdD via un MCD, un MDD et un MCP	22
7.2 Précision du modèle	23
7.3 Évolution du planning	24
8 Mise en place d'un mode graphique	25
8.1 Ergonomie de l'application	25
8.1.1 Se logger	27
8.1.2 Gérer sa collection	27
8.1.3 Proposition automatique de nouvelle BD	28
8.1.4 Obtenir des Recommandations	28
8.2 Les Tests fonctionnels	29
8.3 Derrière, on trouve	31
8.3.1 Architecture du website	31
8.3.2 Tests Unitaires	31

8.4	Le monitoring et alerting	31
8.4.1	Monitoring	31
8.4.2	Alerting	32
9	Pour conclure et dans un futur proche	33
9.1	Conclusion	33
9.2	Axes d'amélioration	33

1 Contexte

Le contexte expliquant le choix de ce sujet est assez simple.

En effet, tout d'abord, faisant mon alternance au Crédit Agricole, une banque dont les données sont très sensibles ; et de plus, de par le haut niveau de sécurité, il n'est pas aisément de pouvoir mettre en place des solutions de Machine Learning.

J'ai donc opté pour un sujet personnel qui me permet librement de mettre en œuvre un projet de Machine Learning de bout en bout. J'aime beaucoup les BD, et je souhaite trouver un outil me permettant d'explorer de nouvelles pistes.

Le but de ce projet est de proposer des recommandations de bandes dessinées. Car en regardant de plus près ce qui existait déjà, j'ai pu constater que les recommandations de BD à l'inverse des recommandations de films¹, étaient les parents pauvres du web.

Mon projet a donc pour objectif, de changer la donne en permettant aux lecteurs de Bandes Dessinées de disposer d'outil de recommandation.

2 D'où proviennent les données

Un des éléments importants pour un système de recommandations efficace est d'avoir des données en grandes quantités et fiables. Pour ce faire, j'ai fait chercher des jeux de données sur :

data.gouv.fr

www.kaggle.com

www.bnfr

Sur divers sites liés à la data, et malheureusement, je n'ai pas trouvé de jeu de données qui correspondaient à mes attentes. Aucun ne mettait en valeur les genres- mots clefs de façon à répondre à mes attentes.

Pour répondre à mes besoins j'ai exploré plusieurs sites de bédéophile, il n'y a pas de dataset préexistant, je l'ai donc conçu en scrappant le site babelio² (qui est une référence dans le monde de la littérature). Tout dernièrement le site <https://www.bdphile.info/> répond aussi à mes attentes. L'avantage d'avoir scrappé les données est de n'avoir que les données, qui nous seront utiles et dans le format désiré.



FIGURE 1 – Site Web d'où sont extraites les données

1. IMDB par exemple <https://www.imdb.com/>
2. <http://www.babelio.fr>

Les scripts que j'ai utilisé pour scrapper les données sont très chronophages (plusieurs dizaines de jours) ; il a fallu ralentir les scripts et faire des pauses pour que le scrapping puisse s'effectuer sans trop de problème) Exemple de lignes de code utilisées l'intégralité des scripts se trouve en annexe.

```
...
for num,Note in enumerate(Lst_Notes): # On a une critique par lecteur
    curseur.execute("SELECT * from user where first_name=?",([Lst_Lecteurs[num]]))
    if len( curseur.fetchall() ) == 0:
        curseur.execute("INSERT INTO user (email,password,first_name) VALUES (?,?,?,?)",
        (str(Lst_Lecteurs[num])+"@BD.BD","sha256$vCGhaVEmTI7tvsmN$884cf52c0f1660296b69035c986a70730
        e951763c5a9f2d2bd3363b418d02",str(Lst_Lecteurs[num])))
    connecteur.commit()

    curseur.execute("SELECT id from user where first_name=?",([Lst_Lecteurs[num]]))
    id_user=curseur.fetchone()
    curseur.execute("SELECT titre FROM note where titre = ? AND auteur= ?",
    (Title,Lst_Lecteurs[num])) #Il faut une liste pour éviter l'erreur de Binding
...

```

Cette fonction ajoute un utilisateur s'il n'existe pas déjà

J'ai donc réussi à obtenir 11299 tomes, 12493 utilisateurs et 957231 critiques. Je scrappe³ donc la totalité du site, pour le moment où je ferai un moteur qui prend en compte toutes les critiques.

3 Un planning qui a su évoluer

3.1 En tout début de projet

En tout début de projet, j'ai établi ce « planning » :

Recommandations simples par genre

Recommandations simples par Description (words embedding)

Et enfin, recommandations par critique (analyse de sentiments)

Le planning a évolué au cours du développement de KoiLire, et vous trouverez en annexe quelques images montrant cette évolution. En complément vous trouverez aussi les sprints backlog Voici une représentation simple de mon projet à son début :

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Analyse des sites de critiques littéraires Scraping des données Constitution de la BDD analytique	Nettoyage des données Analyse des données Visualisation	Entrainement du/des modèles Optimisation du modèle	ORM Base de données Session Utilisateur	Retour des résultats sur l'interface graphique Monitoring

FIGURE 2 – Planning prévisionnel

Pour ne pas perdre de temps dans la mise en place de mon projet, j'ai adopté un déploiement en mode Agile. Normalement, dans des projets en mode Agile, plusieurs personnes sont impliquées. Étant

3. Script annexe p5 à 9

le seul intervenant, même si les formateurs ont été mis à contribution de temps en temps, j'ai porté toutes les casquettes.

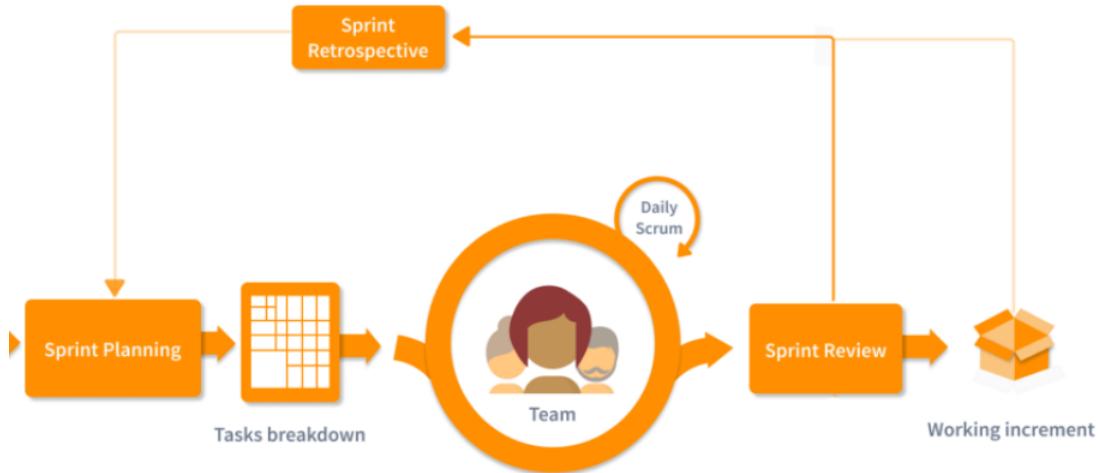


FIGURE 3 – Planning SCRUM prévisionnel

Pour être certain de bien faire toutes les tâches et dans un ordre logique j'ai mis en place des tableaux kanban ; un par sprint (tous les tableaux kanban se trouvent en annexe).

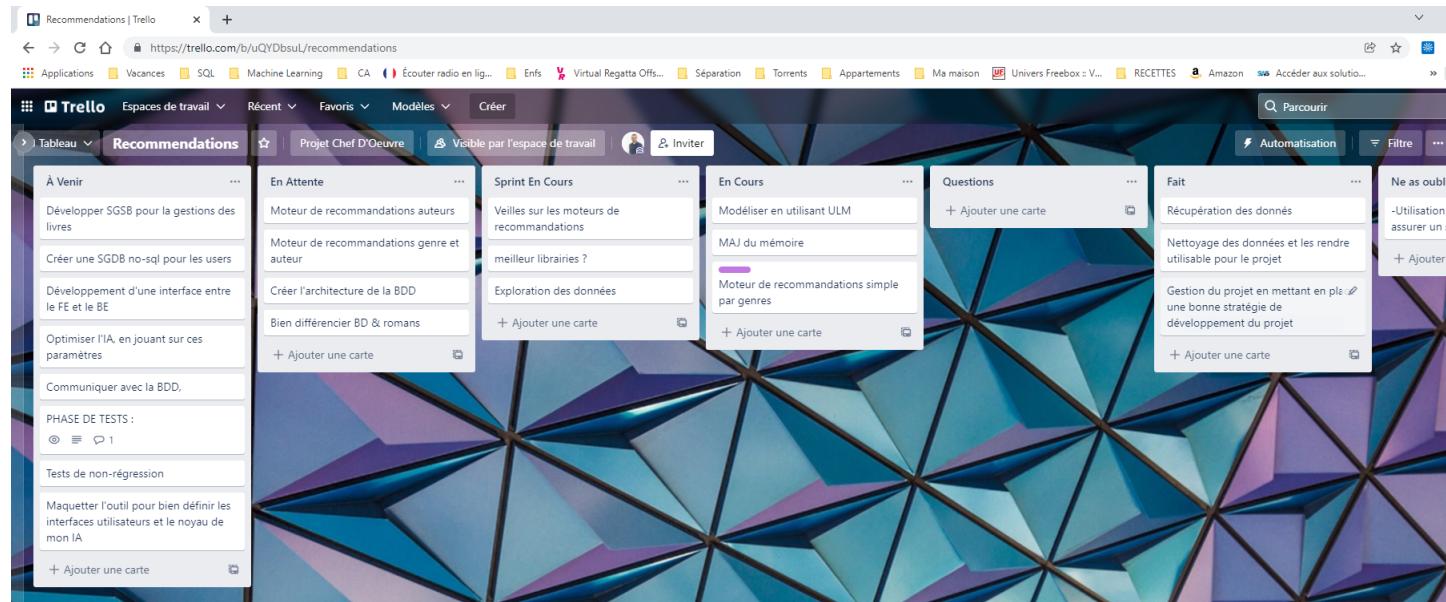


FIGURE 4 – Tableau kanban du 1er sprint

3.2 Mise en place du projet

Cas d'usage Le projet représenté via UML a ce design :

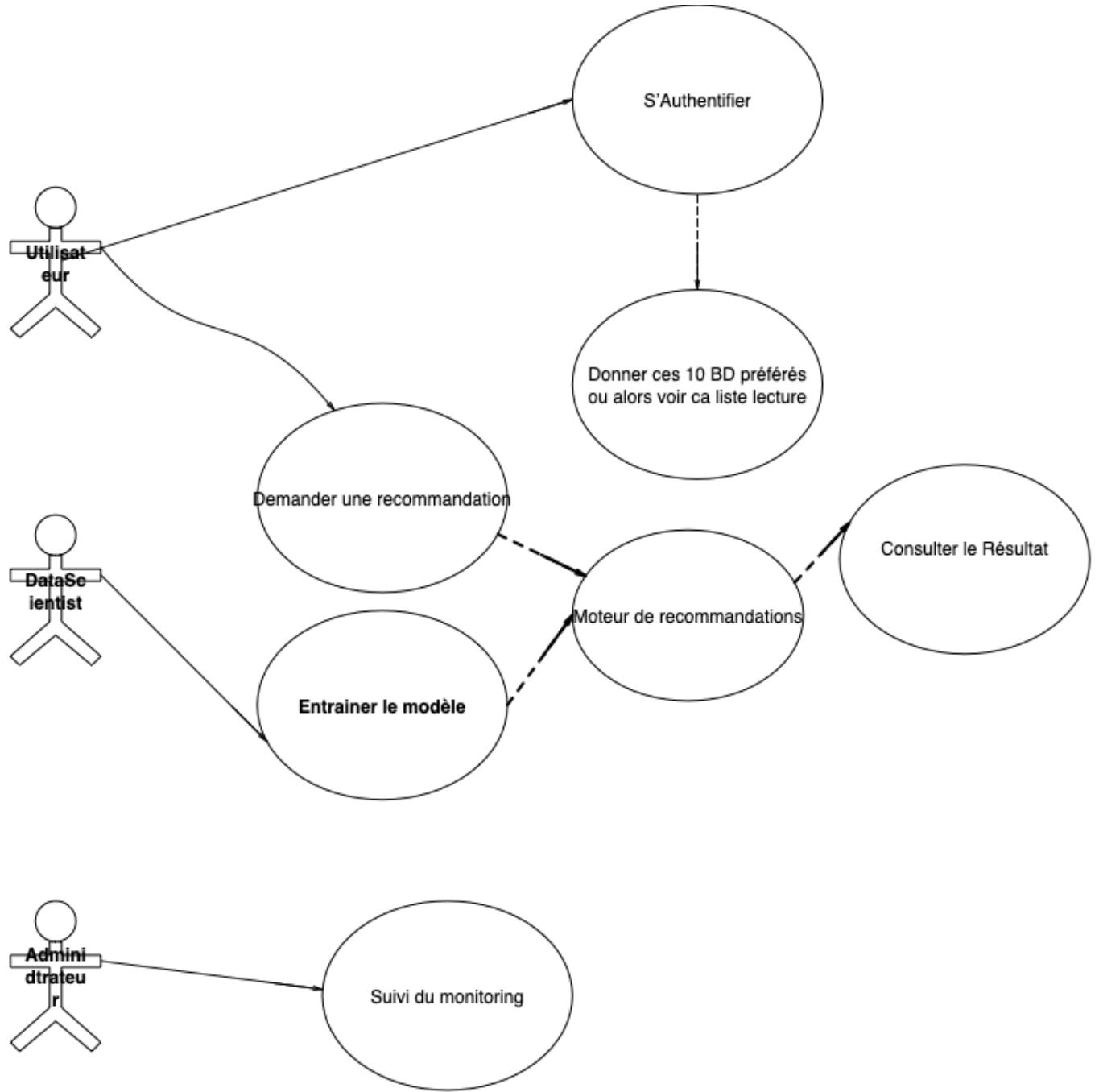


FIGURE 5 – Découpage fonctionnel

Chacun des différents utilisateurs est totalement indépendant des autres, que nous pouvons séparer en 3

phases distinctes. La première liée à l'utilisateur, dont les pré-requis sont une appétence pour la Bande Dessinée, est de pouvoir se connecter à l'outil.

La seconde, correspondant plus à notre rôle, est d'entraîner un modèle qui sera performant sans être gourmand en ressource (évolution vers une application mobile?)

La dernière est liée à la gestion de l'outil, avec le monitoring et aussi le webdesign. Chacun des rôles peut être défini par la User Story suivante :

Au niveau utilisateur :

En tant qu'utilisateur	Je peux créer mon compte
En tant qu'utilisateur	Je peux me connecter
En tant qu'utilisateur	Je peux gérer ma collection c'est à dire ajouter ou supprimer une BD
En tant qu'utilisateur	Je peux voir toutes les critiques que j'ai faites
En tant qu'utilisateur	Je peux faire des critiques
En tant qu'utilisateur	Je peux gérer mes critiques

Au niveau administrateur :

En tant qu'administrateur	Je peux créer me connecter
En tant qu'administrateur	Je peux créer et accéder à flask_monitoring_app
En tant qu'administrateur	Je peux gérer toutes les critiques

Au niveau du Data Scientist :

En tant que Data Scientist	Je peux mettre à jour le fichier de cosine
----------------------------	--

4 Une base données analytiques

Les données sont directement stockées dans une base de données SQL, pour la phase d'analyse, j'utilise la librairie *pandas* pour effectuer mon analyse. Pour l'analyse des données, et leur explicitation je travaillerai sur les 3 tables principales, que sont la table user (qui contient tous les utilisateurs), la table lesBD (qui contient toutes les BD référencées par le système) et la table note (qui contient toutes les critiques faites par les utilisateurs)!!!!;

```
connecteur=sqlite3.connect('database.db')
df=pd.read_sql_query("SELECT titreC,genres,auteur,rating FROM lesBD",connecteur) # DataFrame listant les BD
df=df.rename(columns={'titrec':'titre'})
```

	titre	genres	auteur	rating
0	Les Aventures de Tintin, tome 03 : Tintin en A...	[' journaliste ', ' Reporters ', ' ligne cla...	Hergé	3.55
1	Les Aventures de Tintin, tome 11 : Le Secret d...	[' ligne claire ']	Hergé	4.16
2	Les Aventures de Tintin, tome 17 : On a marché...	[' exploration ', ' journaliste ', ' belge ']	Hergé	4.13
3	Les aventures de Tintin, Tome 6 : L'Oreille ca...	[' mystère ']	Hergé	3.85
4	Les aventures de Tintin, tome 16 : Objectif lune	[' science ']	Hergé	4.08
...

FIGURE 6 – df des BD listées

```
dfUser=pd.read_sql_query("SELECT * FROM User",connecteur)
dfUser
```

	id	email	password	first_name
0	1	XanderOne@BD.BD	sha256vCGhaVEmTI7tvsmN884cf52c0f1660296b6ab...	XanderOne
1	2	Mahpee@BD.BD	sha256vCGhaVEmTI7tvsmN884cf52c0f1660296b6ab...	Mahpee
2	3	GeorgesSmiley@BD.BD	sha256vCGhaVEmTI7tvsmN884cf52c0f1660296b6ab...	GeorgesSmiley
3	4	Crazynath@BD.BD	sha256vCGhaVEmTI7tvsmN884cf52c0f1660296b6ab...	Crazynath
4	5	dedanso@BD.BD	sha256vCGhaVEmTI7tvsmN884cf52c0f1660296b6ab...	dedanso
...

FIGURE 7 – df des utilisateurs du site babelio

```
# Liste des critiques
dfCri=pd.read_sql_query("SELECT Titre,Note,Auteur FROM note",connecteur) # DataFrame listant les critiques
dfCri=dfCri.drop_duplicates()
dfCri=dfCri.reset_index()
dfCri
connecteur.close
```

	index	titre	data	Note	auteur
0	0	Les Aventures de Tintin, tome 03 : Tintin en A...	Cet album ressemble beaucoup aux deux premiers...	4	XanderOne
1	1	Les Aventures de Tintin, tome 03 : Tintin en A...	Durant son aventure au Congo, Tintin a été con...	4	Mahpee
2	2	Les Aventures de Tintin, tome 03 : Tintin en A...	Tintin en Amérique est, selon les formules amé...	4	GeorgesSmiley
3	3	Les Aventures de Tintin, tome 03 : Tintin en A...	Apres l'Afrique, et le Congo plus précisément,...	3	Crazynath
4	4	Les Aventures de Tintin, tome 03 : Tintin en A...	J'aime bien Tintin et, même si cet album-ci n'...	4	dedanso
...

FIGURE 8 – df des critiques de BD

En partant de ces tables, j'ai effectué l'analyse exploratoire suivante

4.1 Analyse Exploratoire

4.1.1 Scrapping des données

Lors du scrapping, je ne récupère qu'une partie des données, celles que je vais utiliser, celles que j'ai prévu d'utiliser(les champs sont prévus en base de données mais ne sont pas exploitées pour l'instant(comme les auteurs secondaires) ainsi que des données que pour l'instant je ne traite pas (comme l'éditeur par exemple).

4.1.2 Valeur nulle

Comme c'est moi qui récupère les données via du scrapping, il n'y a pas ici de donnée manquante, si trop de données sont manquantes, je ne les intègre pas (dans la multitude de données, elles sont négligeables)⁴

```
def missing_values_table(df):
    mis_val = df.isnull().sum()
    mis_val_percent = 100 * df.isnull().sum() / len(df)
    mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)
    mis_val_table.columns = mis_val_table.rename(
        columns = {0 : 'Missing Values', 1 : '% of Total Values'})
    mis_val_table['Missing Values'] = mis_val_table['Missing Values'].sort_values(ascending=False)
    print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n")
    print ("There are " + str(mis_val_table['Missing Values'].shape[0]) +
          " columns that have missing values.")
    return mis_val_table['Missing Values']
missing_values_table(df)
```

4.1.3 Quel type de BD est-ce donc ?

```
df['genres']=df['genres'].apply(ast.literal_eval)
len(df['genres'])
1237
```

4. Les scripts utilisé pour l'analyse exploratoire Annexe p9 à 16

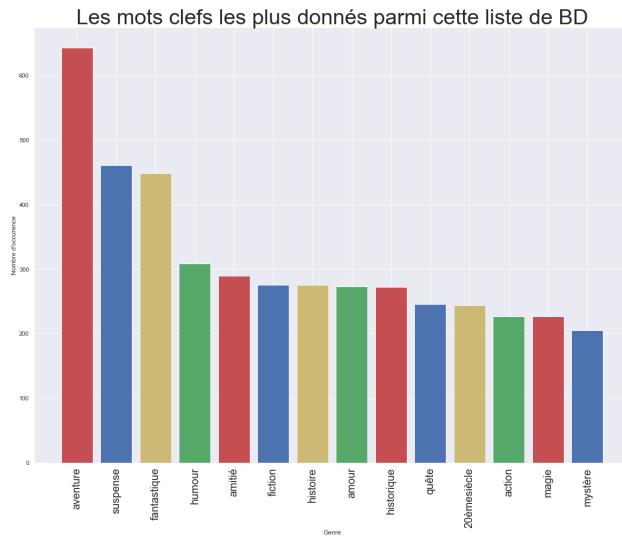


FIGURE 9 – Les mots clés les plus représentés

⁵ Au départ, je pensais faire une recommandation via les genres/mots clefs mais il y en a beaucoup trop. Je vais devoir repenser mon moteur de recommandations

Une autre donnée importante est la note de la BD ; cette note représente la moyenne des notes données par tous les utilisateurs.

4.1.4 De la moins bonne BD à la meilleure

En regardant, les données de plus près, on voit que certaines notes sont erronées (il arrive d'avoir des notes d'1M, certainement dû au scrapping), donc on ne les prend pas en compte. Les notes s'échelonnent de 0 à 5

Note	
0.500000	2117
1.000000	12763
1.500000	4734
1.666667	3
1.900000	3
...	
4.722222	3
4.777778	3
4.857143	3
5.000000	220518
5.500000	13
...	

FIGURE 10 – Notes des BD

5. Le détail de ce script se trouve dans l'annexe Script analyse exploratoire

4.1.5 Les mots qui sont les plus utilisés dans les titres

Pour aider dans le choix des BD de la bédéthèque, je fais un état des lieux des mots le plus présent dans les titres.



FIGURE 11 – Les mots clés les plus présent dans les titres

4.1.6 Les auteurs les plus prolifiques

L'étude permet aussi de voir, quels auteurs sont les plus prolifiques, vous trouverez ci dessous un schéma mettant en avant les 10 auteurs les plus prolifiques. Cette donnée sera prise en compte dans la prochaine version.

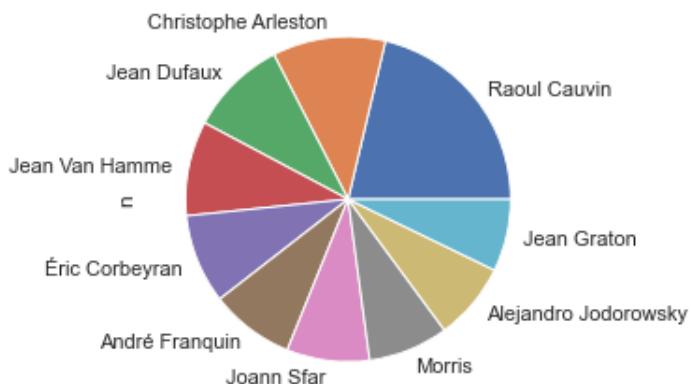


FIGURE 12 – Les auteurs qui produisent beaucoup

4.2 Au niveau utilisateur

4.2.1 Nombre d'utilisateurs

Il n'est pas possible d'accéder aux recommandations, si l'on est pas un utilisateur connu du système. Donc regardons de plus près la base de données "database.db" et plus précisément la table users. Je n'ai pas encore mis en place, un filtrage collaboratif mais celui-ci sera présent dans les prochaines versions.

```
SELECT COUNT(*) FROM lesBD
```

```
SELECT COUNT(*) FROM note  
957231 critiques
```

4.2.2 Les mots qui reviennent le plus dans les commentaires



FIGURE 13 – Les mots clés les plus présent dans les critiques

4.2.3 Les utilisateurs les plus critiques

```
def compte_mot(phrase):
    nb_mot=1;lettre_pre=' '
    if phrase!=None and type(phrase)!=int:
        for lettre in phrase:
            if lettre==' ' and lettre_pre!=' ':
                nb_mot+=1
            else :
                lettre_pre=lettre
    return nb_mot
dfC['TailleCritique']=dfC['data'].apply(compte_mot)
dfC.sort_values(by=['TailleCritique'],ascending=False)
```

Out[55]:		data	date		titre	Note	user_id	auteur	TailleCritique
168450	Donjon est une série finie mais pas vraiment t...	2016-01-29 00:00:00		Donjon Zénith, tome 1 : Coeur de canard	5.0	1019	jyrille	3286	
229869	Faisons le point sur juin 2021. Il a été pour ...	2021-07-11 00:00:00		Watchmen - Intégrale	4.0	220	LeScriboUILARD	3140	
310321	Humanité poignante-Ce tome contient une histoi...	2021-10-15 00:00:00		Rusty Brown	5.0	1401	Presence	2914	
316554	RETOUR AUX FONDAMENTAUXRetour au Sommet des Di...	2017-12-21 00:00:00		Le Sommet des dieux, tome 4	5.0	5818	nebalfr	2530	
410257	Ce tome fait suite à Walking Dead, tome 25 : S...	2018-01-19 00:00:00		Walking Dead, tome 26 : L'appel aux armes	5.0	1401	Presence	2467	
204761	Ce tome comprend une histoire complète et indé...	2017-05-24 00:00:00		A La Recherche de Peter Pan, tome 1	5.0	1401	Presence	2396	
340245	Ce tome fait suite à La frontière invisible. I...	2014-12-03 00:00:00		Les cités obscures : La théorie du grain de sable	5.0	1401	Presence	2394	
442793	Ce tome fait suite à Requiem, Chevalier Vampire...	2019-01-17 00:00:00		Requiem, Chevalier Vampire, tome 7 : Le couven...	5.0	1401	Presence	2392	
240758	Ce tome fait suite à Walking Dead, tome 1 : Pa...	2017-01-05 00:00:00		Walking Dead, Tome 4 : Amour et mort	5.0	1401	Presence	2346	
385978	Ce tome fait suite à Walking Dead, tome 26 : L...	2018-02-05 00:00:00		Walking Dead, tome 27 : Les Chuchoteurs	5.0	1401	Presence	2346	

FIGURE 14 – Les plus grandes critiques

Par exemple pour jyrille on a toutes ces critiques :

On note malgré tout des doublons, ceux-ci sont repérables car les colonnes "data", "note", "date" sont exactement les mêmes. On ne les traitent donc pas, on les supprime. Ceux ci sont dûs au scrapping de données. Même si j'ai des protections (avec des tests de présence SQL)

```

curseur.execute("Select titre from lesbd where titre = ?",[Title])
#Il faut une liste pour éviter l'erreur de Binding
if len( curseur.fetchall() ) == 0:

```

Du fait du scrapping (problèmes de connections par exemple), j'ai dû éliminer des doublons.

```
dfC=dfC.drop_duplicates()
```

		data	date		titre	Note	user_id	auteur	TailleCritique
168397	Donjon est une série finie mais pas vraiment t...	2016-01-29 00:00:00		Donjon Zénith, tome 1 : Coeur de canard	5.0	1019	jyrille		3286
11998	Et Franquin crée Lagaffe... le titre n'est pas...	2015-04-15 00:00:00		Gaston, Tome 9 : Le cas Lagaffe	5.0	1019	jyrille		1569
376026	Patience est le dernier roman graphique de Dan...	2018-06-08 00:00:00			Patience	4.0	1019	jyrille	1538
401687	Cet article est une introduction aux mondes de...	2015-04-15 00:00:00			Le garage hermétique	3.0	1019	jyrille	1355
227215	On ne présente plus V pour Vendetta mais on va...	2015-07-31 00:00:00			V pour Vendetta : Intégrale	5.0	1019	jyrille	1327
200733	S.O.S. Bonheur est une série finie qui comport...	2021-06-24 00:00:00			S.O.S Bonheur : Intégrale	5.0	1019	jyrille	1144
24365	Blast est une série finie en quatre tomes. Ell...	2015-06-13 00:00:00			Blast, Tome 1 : Grasse carcasse	4.0	1019	jyrille	1121
391454	Le bus est une BD au format à l'italienne. Ell...	2021-02-03 00:00:00			Le Bus	4.5	1019	jyrille	1018
273724	Le roman graphique du jour est un one-shot de ...	2019-06-18 00:00:00			Ici même	5.0	1019	jyrille	1009
87568	Idées noires est une série en deux tomes (dont...	2015-06-13 00:00:00			Idées noires : L'Intégrale	5.0	1019	jyrille	840
4945	Le retour à la terre est une série en cours en...	2015-06-11 00:00:00			Le retour à la terre, Tome 1 : La Vraie Vie	5.0	1019	jyrille	831
345895	Désidément, Franquin est inusable à tous les n...	2013-09-27 00:00:00			Souvenirs de l'empire de l'atome	5.0	1019	jyrille	471
245362	De nombreux points communs existent entre Mon ...	2013-09-26 00:00:00			Mon ami Dahmer	5.0	1019	jyrille	429
19953		None	2015-04-15 00:00:00		Gaston, Tome 9 : Le cas Lagaffe	5.0	1019	jyrille	1
347649		None	2013-09-27 00:00:00		Souvenirs de l'empire de l'atome	5.0	1019	jyrille	1
349127		None	2013-09-27 00:00:00		Souvenirs de l'empire de l'atome	5.0	1019	jyrille	1

FIGURE 15 – Les critiques de jyrille

Quand None est indiqué cela veut simplement dire qu'il a évalué une BD sans laisser de commentaire.

Cette exploration des données m'a permis de voir, que l'orientation du projet, que je préconisait au départ devait évoluer

5 Qui y a-t'il sous le capot

5.1 Petits rappels sur les moteurs de recommandations

Pour exploiter ce qui se fait de mieux dans la recommandation, j'ai effectué une veille sur les différents modèles permettant la mise en place de recommandations de films (qui est très documentée sur le Net). Cette veille m'a aussi permis de mettre en avant certains processus qui me paraissaient applicables aux bandes dessinées.

(cette présentation vient du site [jeda.co](#) et présente très bien le sujet) Au delà des exemples de Netflix et Amazon, les systèmes de recommandations sont beaucoup plus répandus que l'on pense. On distingue plusieurs classes de ces usages pour des buts distincts :

- Augmenter d'interaction entre utilisateurs : les réseaux sociaux vont vous suggérer des amis (Facebook, Linkedin pour ne citer qu'eux)
- Augmenter l'engagement des utilisateurs : valable pour les sites de contenus tels que LeMonde.fr, Youtube, Netflix, Spotify
- Augmenter le chiffre d'affaires : c'est une raison plus évidente, avec notamment les sites de e-commerce, des entreprises du retail, par exemple la Fnac, Amazon, ou Carrefour lorsque vous faites vos courses en ligne.

Sérendipité : Ce mot relativement impressionnant désigne le fait de réaliser une découverte ou invention de manière inattendue. Un exemple des plus connus ? La découverte de l'Amérique. Cela s'applique au cas des moteurs de recommandations dans la mesure où par les suggestions d'amis, de contenus ou d'articles, l'utilisateur ou client va être guidé vers quelque chose qu'il n'aurait pas trouvé par lui-même et qui serait, en plus, susceptible de lui plaire.

Mais ce concept est flou. Finalement, comment être sûr qu'un moteur de recommandations fonctionne pour les utilisateurs ? La question se pose de l'évaluation d'un bon moteur de recommandations. Plusieurs possibilités pour évaluer la qualité de la recommandation.

- Mesurer la satisfaction de la recommandation : via des notes laissées à des contenus suggérés ou les likes. Pour récupérer ces données, l'API faisant la recommandation doit s'assurer que tous les utilisateurs aient un compte.
- Des mesures « indirectes » à la recommandation peuvent être faites. Pour faire de meilleurs comparatifs dans l'analyse de ces mesures, les A/B tests sont très souvent utilisés.

Comment construire un moteur de recommandations ? Il y a 3 grands modèles qui peuvent être utilisés pour construire un moteur de recommandations : une approche model based et deux autres basés sur du calcul de similarité

— L'approche « model-based » : c'est une approche qui est entièrement basée sur des algorithmes de Machine Learning. En prenant l'exemple de Netflix, il va s'agir avec cette approche de prédire les préférences de films d'un utilisateur en fonction des notes données sur ce film par d'autres utilisateurs. C'est l'approche mathématiques dont on entend le plus parler, mais celle qui est la moins mise en production.

2 problèmes à ceci : Les données peuvent manquer, et le temps de calcul peut être très long, étant donné la complexité de ces modèles. La recommandation en temps réel n'est alors pas possible.

— L'approche « Item – based » : l'approche produit permet de se référer aux caractéristiques du produit, du film, de la musique pour construire des sous-groupes de produits qui sont similaires. Si un utilisateur a vu des films de super-héros, il va lui être proposé par la suite d'autres films de super-héros.

— L'approche « User – based » : dans cette approche, on va se reposer sur les informations données sur le compte de utilisateur, ses préférences, ses goûts. Cette fois, ce sont les utilisateurs qui vont être groupés. En prenant l'exemple des films, le moteur va proposer les mêmes films aux utilisateurs de mêmes caractéristiques. Ces deux dernières stratégies sont préférées car elles permettent de faire des recommandations en temps réel. Quant aux bases de données utilisées, celles-ci doivent être adaptées, on peut utiliser des bases de données en graphes, elles permettent d'expliquer les relations entre chaque élément (utilisateur BD par exemple).

5.2 Rappel rapide sur les Transformers

Pour simplifier, il s'agit d'outils vous permettant de rapidement déterminer si les gens parlent en bien ou en mal d'un sujet donné. A l'aide de modèles d'apprentissage profond qui imitent les fonctions cognitives du cerveau, cette technologie peut comprendre des structures linguistiques complexes et même traiter des formes basiques d'ironie et de sarcasme, donc les tournures de phrases. J'ai décidé de mettre en avant les transformers :CAMENBert développés par Huggin face. Je l'utilise car il "maîtrise" les subtilités de la langue française.

BERT (présenté ici <https://arxiv.org/abs/1810.04805>) signifie Bidirectional Encoder Representations from Transformers. BERT en deux mots :

1. Bidirectionnel - pour comprendre le texte que vous lisez, il vous faut prendre en compte le mot précédent et le mot suivant.
2. Transformers - The Attention Is All You Need paper a présenté le modèle Transformer. Le Transformer lit des séquences entières de jetons à la fois. Dans un sens, le modèle est non directionnel, tandis que les LSTM lisent séquentiel (de gauche à droite ou de droite à gauche). Le mécanisme d'attention permet d'apprendre les relations contextuelles entre les mots.
3. Intégrations de mots contextualisés (pré-entraînés) - Le document ELMO a introduit un moyen d'encoder les mots en fonction de leur signification / contexte. Canard a plusieurs significations -l'oiseau qu'on trouve près de nos cours d'eaux, ou bien le sucre qu'on trempe dans le café. L'utilisation des transformes a permis au moteur de recommandations, de prendre en compte les jeux de mots, les tournures de mots ainsi que l'ironie

6 Le modèle

6.1 Mon modèle

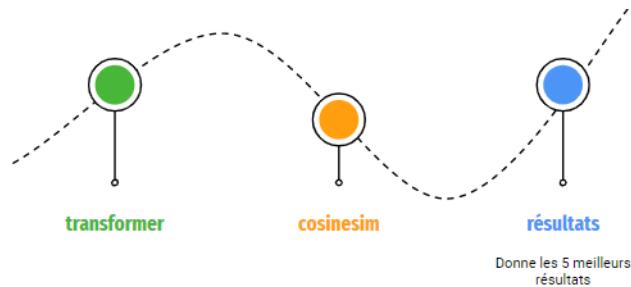


FIGURE 16 – Représentation de mon modèle

Je scrappe les données du site babelio, durant l'analyse exploratoire je me suis rendu compte que les recommandations par mots clefs étaient peu performantes ; j'ai donc opté pour une autre approche. En regardant plus attentivement le jeu de données, je me suis dit qu'utiliser la description serait une bonne idée, car mes recommandations ne s'enfermeront pas dans un genre unique. Après une étude approfondie, je me suis décidé à utiliser les transformers. En entrée de mon modèle, on a

	titre	resume
0	Les Aventures de Tintin, tome 03 : Tintin en A...	[Dans "Tintin en Amérique" (1932), le héros c...]
1	Les Aventures de Tintin, tome 11 : Le Secret d...	["Dans cette aventure, Tintin accompagne le Ca...
2	Les Aventures de Tintin, tome 17 : On a marché...	['La fusée a décollé de Syldavie, emportant Ti...
3	Les aventures de Tintin, Tome 6 : L'Oreille ca...	["Afin de récupérer un fétiche volé, Tintin s'...
4	Les aventures de Tintin, tome 16 : Objectif lune	['De retour de voyage, Tintin et le capitaine ...
5	Les Aventures de Tintin, tome 04 : Les Cigares...	["Alors qu'il est en croisière sur un paquebot...
6	Les Aventures de Tintin, tome 05 : Le Lotus bleu	["Poursuivant jusqu'en Chine les trafiquants r...
7	Les Aventures de Tintin, tome 12 : Le Trésor d...	['Dans Le Secret de la Licorne (1943) et Le Tr...
8	Les Aventures de Tintin, tome 07 : L'Île Noire	['Un mystérieux avion met Tintin, pour la prem...
9	Les aventures de Tintin, tome 22 : Vol 714 pou...	["Dans cette nouvelle aventure, nous retrouvons...

FIGURE 17 – Le jeu de données avant l'utilisation de BERT

On note que l'information est encore textuelle, et après BERT

```

array([[-0.40069798,  0.8627006 , -0.27910104, ..., -0.4157493 ,
       0.6491381 ,  0.08155891],
      [-0.7647829 ,  0.81523013,  0.51436025, ..., -0.56890714,
       0.26642147,  0.7483811 ],
      [-0.33948687,  1.1335102 ,  0.5711372 , ..., -0.368568 ,
       0.578115 ,  0.6559747 ],
      ...,
      [-0.33987653,  0.8547068 ,  0.460571 , ..., -0.06200056,
       0.39029774,  0.8161199 ],
      [-0.56684387,  0.57430196,  0.55268896, ..., -0.5056164 ,
       -0.00258925,  1.0193282 ],
      [-0.65133756,  0.71998787,  0.33641106, ..., -0.5049894 ,
       0.47493964,  0.4751004 ]], dtype=float32)

```

FIGURE 18 – Les données vectorisées

6.2 Utilisation de BERT

Pour mon modèle j'ai utilisé un modèle pré-entraîné(bert-base-nli-mean-tokens) BERT. Le transformer gère aussi toute la partie preprocessing (ce qui explique que la prédiction prenne du temps). Du coup ces recommandations permettent à l'utilisateur de rester dans des thèmes qui lui sont chers tout en explorant de nouvelles voies auxquelles il n'aurait pas penser lui même. Désormais les données sont vectorisées et organisées par similitude.

6.3 Une aide pour remplir sa bédéthèque

J'ai aussi mis en place un autre moteur de recommandations qui propose une BD suivant la BD que l'on vient de sélectionner ; ce moteur quand à lui est basé sur la similarité des titres Voici à quoi ressemble la logique de la proposition

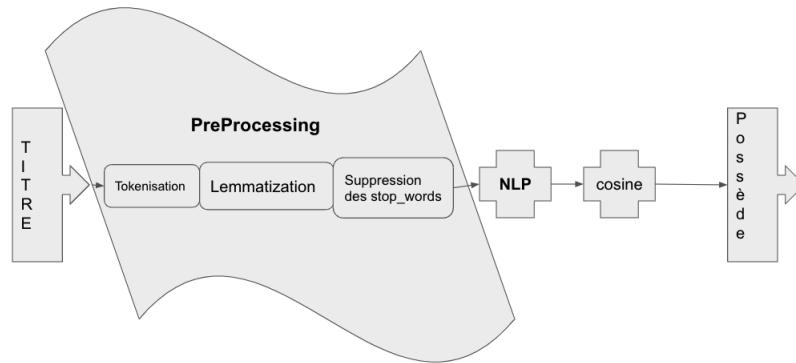


FIGURE 19 – Aide pour remplir les étagères

Le code se trouve en annexe⁶

6. Annexe page 40

6.4 Comment marche la cosme

Pour mes recommandations j'utilise la cosme, elle représente la similarité de deux vecteurs à n dimensions en déterminant le cosinus de leur angle. Ce score est fréquemment utilisée en TAL.

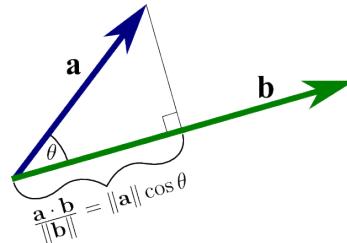


FIGURE 20 – Représentation de la cosme

C'est à dire plus l'angle est petit plus les textes se ressemblent

6.5 Premiers résultats

Pour la 1ère étape, tout se fait en mode console, on entre manuellement la BD et le moteur nous retourne les 5 meilleures recommandations

```
Entrer le titre que vous aimez : Astérix, tome 6 : Astérix et Cléopâtre
Les 5 recommandations que nous pouvons vous faire pour Astérix, tome 6 : Astérix et Cléopâtre
Weegee : Serial photographer
Astérix, tome 17 : Le Domaine des dieux
Astérix, tome 18 : Les Lauriers de César
Astérix, tome 20 : Astérix en Corse
Astérix, tome 3 : Astérix et les Goths
Entrer '0' pour sortir et ce que vous voulez pour continuer 0
Bye
```

FIGURE 21 – Les recommandations en mode console

Les Aventures de Tintin, tome 11 : Le Secret de La Licorne :

- Les petits Mythos, tome 2 : Le grand Icare
- Donjon Parade, tome 6 : Garderie pour petiots
- Agrippine, tome 1 : Agrippine
- Agrippine, tome 3 : Les combats d'Agrippine
- grippine, tome 4 : Agrippine et les inclus

⁷

```
def get_title(index):
    return df[df.index == index]["titre"].values[0]
def get_index(title):
    return df[df.titre == title]["index"].values[0]
```

7. La recommandation en ligne de code Annexes p23 à 27

```

df['index'] = [i for i in range(0, len(df))]

bert = SentenceTransformer('bert-base-nli-mean-tokens')
precosine=bert.encode(df['resume'].tolist(), batch_size=256)
similarity =cosine_similarity(precosine)

# Test en mode console
notOver = True
while(notOver):
    user_bd = input("Entrer le titre que vous aimez : ")

# Generate Recommendations
recommendations = sorted(list(enumerate(similarity[get_index(user_bd)])), key = lambda x:x[1],reverse=True)
print("Les 5 recommandations que nous pouvons vous faire pour " + user_bd + " sont: ")
print(get_title(recommendations[4][0]), get_title(recommendations[5][0]),sep = "\n")
decision = input("Entrer '0' pour sortir et ce que vous voulez pour continuer")
if(int(decision) ==0):
    print("Bye")
    notOver = False

```

6.6 Évaluation du modèle

Il n'y a pas, à proprement parler de métrique d'évaluation ; nous sommes dans le cas de modèle non supervisé ; seul le A/B testing peut apporter une réponse. Pour vérifier mes recommandations j'ai effectué des tests que j'ai vérifiés manuellement : Premièrement, j'ai effectué un A/B testing , vous trouverez ici un exemple de ce que j'ai testé ; j'ai aussi demandé à des connaissances (qui apprécient la BD) d'évaluer mon moteur :

Nous vous recommandons pour la BD Les Aventures de Tintin, tome 11 : Le Secret de La Licorne :

1. Les petits Mythos, tome 2 : Le grand Icare
2. Donjon Parade, tome 6 : Garderie pour petiots
3. Agrippine, tome 1 : Agrippine
4. Agrippine, tome 3 : Les combats d'Agrippine
5. Agrippine, tome 4 : Agrippine et les inclus

Nous vous recommandons pour la BD Les aventures de Tintin, tome 22 : Vol 714 pour Sydney :

1. Blake et Mortimer, tome 15 : L'Etrange rendez-vou
2. L'héritage des dieux. 1, Likonda
3. Seuls, Tome 3 : Le clan du requin
4. Star Wars - Chevalier errant, Tome 1 : Ignition
5. Keos, tome 1 : Osiris

Nous vous recommandons pour la BD Astérix, tome 1 : Astérix le gaulois :

1. Spirou et Fantasio - Intégrale, tome 8 : Aventures humoristiques
2. Astérix, tome 26 : L'odyssée d'Astérix

3. Astérix, tome 25 : Le Grand Fossé
4. Astérix, tome 27 : Le Fils d'Astérix
5. Astérix, tome 28 : Astérix chez Rahâzade

Nous vous recommandons pour la BD XIII, Tome 2 : Là où va l'Indien... :

1. Tome 3 : Toutes les Larmes de l'Enfer
2. t dans les Saules, Tome 4 : Foutoir au manoir
3. aux vents, tome 1 : La Folle et l'Assassin
4. Alisik, tome 1 : Automne
5. Ratafia, tome 2 : Un zèle imbécile

Nous vous recommandons pour la BD Lanfeust de Troy, tome 6 : Cixi impératrice :

1. Lanfeust de Troy, Tome 4 : Le paladin d'Eckmül
2. Lanfeust de Troy, tome 3 : Castel Or-Azur
3. Lanfeust de Troy, anfeust de Troy, tome 7 : Les Pétaures se cachent pour mourir
4. Lanfeust de Troy, tome 8 : La Bête fabuleuse
5. Lanfeust de Troy, tome 5 : Le frisson de l'Haruspice

A part, pour le premier (le secret de la licorne), les résultats semblent cohérents, pour étayer ceci j'ai mis en place (sous les conseils d'un formateur) une métrique qui se présente ceci :

Le postulat pour cette métrique, est que si on aime la BD entrée, on aime les propositions ! Où l'inverse si on les déteste. Pour simuler ceci, et ainsi obtenir une métrique j'ai effectué ceci :

- Le lecteur choisi une BD
- Le moteur lui fournit des recommandations
- On prend tous les lecteurs qui ont noté la BD choisie
- On fait (la différence entre la note de la BD choisi - la note de la cosme si elle existe)² qu'on appellera a^2
- On fait cela pour les critiques/note de la BD choisie - Du coup, on peut évaluer le nombre de critiques lui qu'on nommera b^2
- On fait ces savants calculs pour chaque recommandation et on aura a^2/b^2
- une petite racine carrée et on obtient : NOTRE ÉVALUATION

On remarque, que plus il y a de critiques, plus la recommandation est bonne.

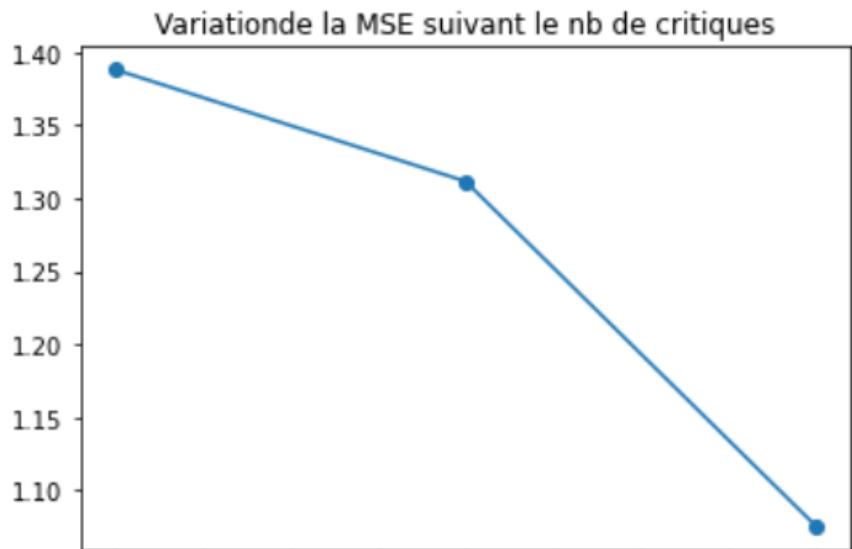


FIGURE 22 – Plus on a de critiques, meilleur est le modèle

7 Pour les besoins du projet : mise en place d'une base de données relationnelle

La base de données analytique n'a été utilisée que pour l'analyse des données, elle a une importance dans la première partie du projet mais j'ai très rapidement mis en place ma Base de Données Relationnelle.⁸ Pour un contenu plus détaillé, merci de vous référer aux annexes.

Pour ce POC j'utilise sqlite, mais pour une exploitation de mon application en production , la mise en place d'une solution telle que MongoDB (bdd non relationnelle) ou MySQL sera nécessaire.

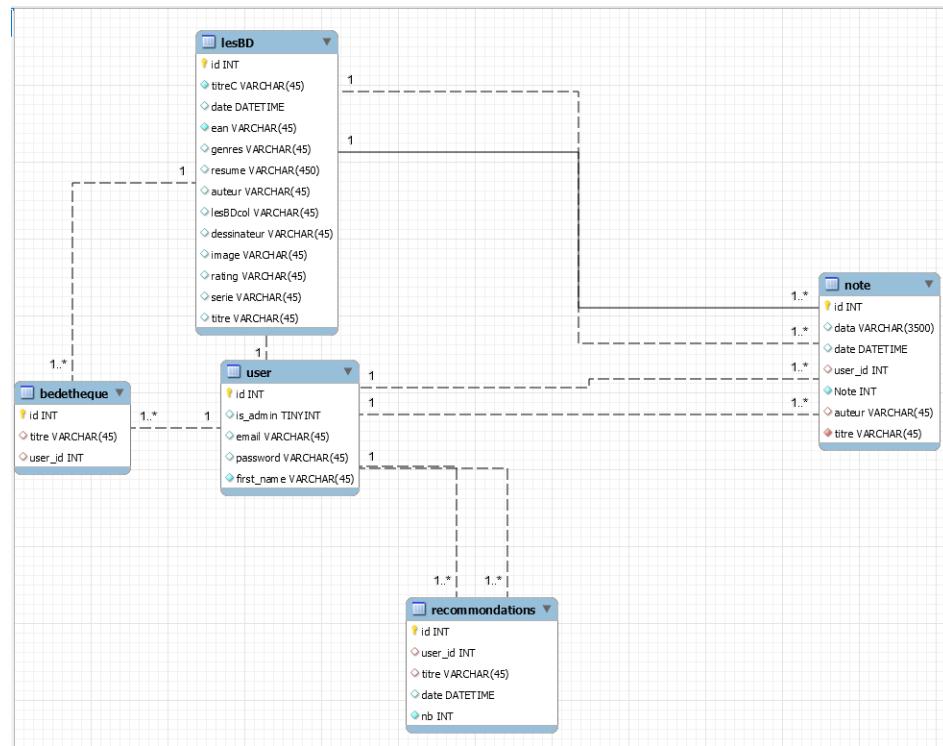
Pour pouvoir faire des recommandations, et sauver ce résultat, tout en gardant l'historique du lecteur, je me suis appuyé sur le moteur de base de donné sqlite et pour la partie graphique je me suis appuyé sur l'ORM sql-Alchemy. J'utilise donc 5 tables directement en lien avec le lecteur et d'autres liées au fonctionnement de l'API

1. user : table où tous les users/lecteur sont sauvés
2. note : toutes les critiques
3. lesBD : la liste (non exhaustive) des Bandes dessinées
4. bedetheque : collection de chaque lecteur
5. recommandations : tables où sont sauvées les recommandations faites par moteur

Les tables liées au fonctionnement de l'API

1. mse2 : table où les cosine sont sauveées
2. notations : table de note de 0 à 5

7.1 Explication de la BdD via un MCD, un MDD et un MCP



8. La base sqlite annexe p17 à 22

FIGURE 23 – MCD by MySqlWorkbench

Lecture des associations

- Un et un seul utilisateur(table user) peut avoir plusieurs recommandations (table recommandations)
- Un et un seul utilisateur (table user) peut avoir fait plusieurs critiques (table note)
- Un et un seul utilisateur (table user) peut avoir plusieurs BD dans sa collection (table bédéthèque)
- Une BD (table lesBD) peut être possédée par plusieurs utilisateurs (table users)

Les clefs étrangères (FK : Foreign Key)

- table note : FK user_id et auteur (respectivement id et first_name de la table user)
- table bedetheque : FK titre (titreC de la table lesBD) user_id (id de la table user)
- table recommandations : FK titre (titreC de la table lesBD) user_id (id de la table user)

Pour les besoins du projet, toutes ces tables ont été créées. La table principale qui référence toutes les BD. Toutes ces tables se trouvent en annexes

Toutes les interactions sont gérées par des requêtes SQL ou bien des commandes via SQL alchemy. Même les requêtes SQL sont gérées par l'ORM, ce qui explique la nomenclature :

par exemple :

Cette ligne de commande permet d'insérer une nouvelle BD

```
curseur.execute("INSERT INTO lesbd  
→ (serie,titre,titrec,ean,genres,resume,auteur,image,date,ranking) VALUES (?,?,?,?,?,  
→ ?, ?, ?, ?, ?, ?, ?)",(Serie,Title,TitreC, EAN, str(Genres), Description, Auteur1,  
→ img_src, Date, Rating))
```

et celle ci permet de savoir si la personne l'a déjà critiquée

```
curseur.execute("SELECT titre FROM note where titre = ? AND auteur=  
→ ?",(Title,Lst_Lecteurs[num]))
```

ou directement via des commandes SQL Alchmy :

Cette commande permet de savoir si le lecteur possède cette BD dans sa collection

```
bedetheque.query.filter((bedetheque.user_id==current_user.id)&(bedetheque.titre==request.form['titre']))
```

et celle-ci permet d'incrémenter le nombre de fois où une BD est recommandée.

```
db.session.query(recommandations).filter(recommandations.user_id==current_user.id or  
→ recommandations.titre==get_title(dfCo,tt_tmp)).update({recommandations.nb:recommandations.nb+1})
```

L'ensemble des requêtes se trouvent en annexe (page 54) C'est à ce moment, après avoir constaté la multitude de plot-keywords (11297) que j'ai décidé de m'orienter sur des recommandations via la description.

7.2 Précision du modèle

On note que plus on a de critiques plus la précision est bonne, même si la différence est minime ; mais plus la base de donnée est importante plus les temps d'exécution sont longs.

7.3 Évolution du planning

Comme je travaille dans l'esprit Agile, le planning a évolué, en effet il a fallu prendre en compte l'évolution du moteur de recommandations.

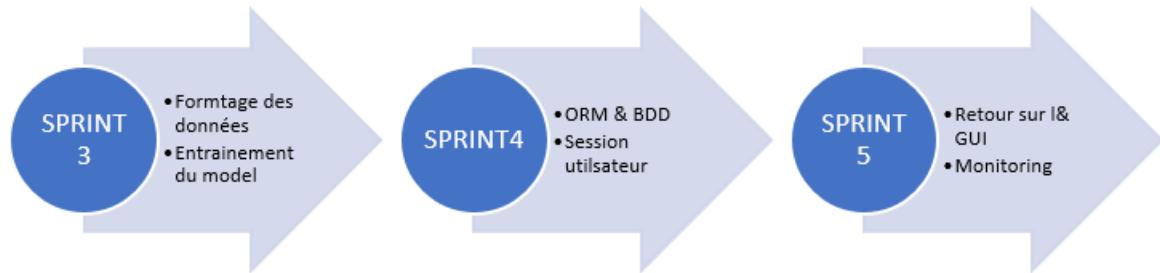


FIGURE 24 – Ce qui reste à faire à mi-parcours

8 Mise en place d'un mode graphique

Ce premier test primordial, consiste à vérifier qu'en passant du mode console au mode graphique, l'application délivre toujours la même réponse.

Voir nos propositions Nous vous recommandons pour la BD Astérix, tome 6 : Astérix et Cléopâtre:				
Astérix, tome 17 : Le Domaine des dieux	Astérix, tome 18 : Les Lauriers de César	Astérix, tome 20 : Astérix en Corse	Astérix, tome 3 : Astérix et les Goths	Astérix, tome 2 : La Serpe d'or

FIGURE 25 – Réponse en mode graphique

On note une légère différence entre les 2 résultats, c'est dû à une BDD de plus en plus importante ; et avec notre regard d'humain on peut dire que la recommandation s'affine.

8.1 Ergonomie de l'application

Pour passer à l'étape graphique, j'ai utilisé le moteur flask. Par rapport à d'autre Django, Flask est beaucoup plus simple à mettre en place. Je n'ai pas utilisé Streamlit car ce moteur ne prend pas en charge les sessions, qui dans mon cas sont importantes. J'ai essayé de mettre en place une charte graphique assez simple, car ce qui importe ici ce sont les recommandations, plus que le design. Pour mener à bien cette tâche, j'ai séparé (logiquement) mon API en 2 parties distinctes. Le back-end avec "l'intelligence" de l'API et le front-end avec le visuel.

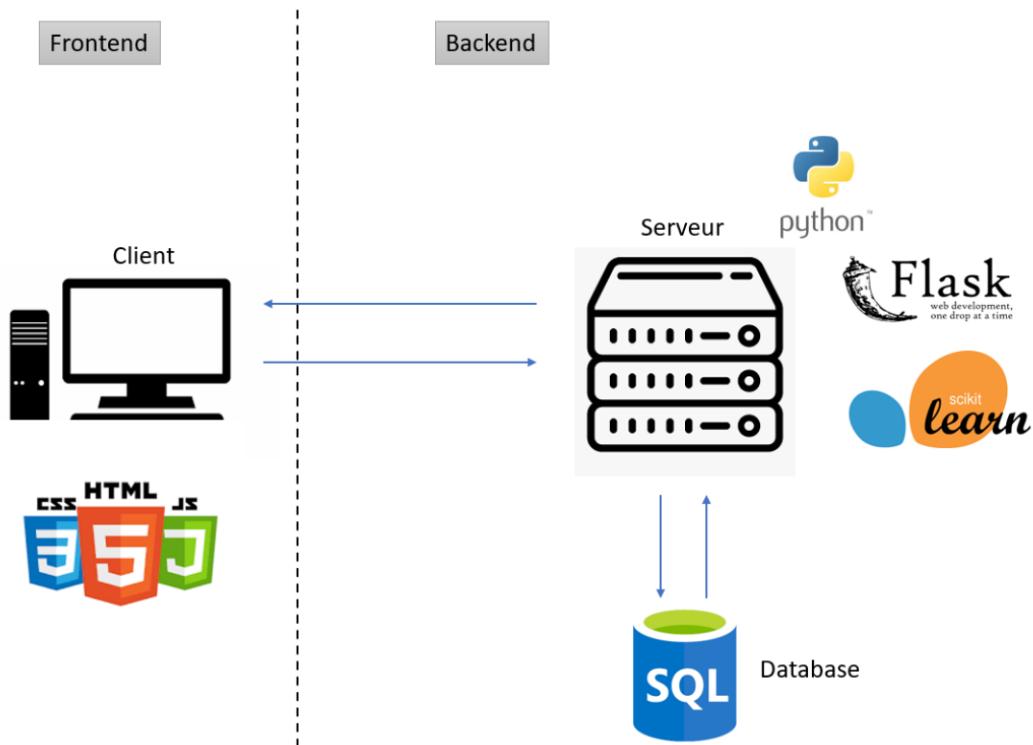


FIGURE 26 – Vue d'ensemble de l'API

La partie front-end (qui est vue/perçue par l'utilisateur final) utilise bootstrap pour avoir un site responsive, et qui sera des plus simple à faire évoluer vers une version mobile. Le back-end quant à lui utilise flask ; cette utilisation permet aux scripts python, donc aux recommandations d'être effectuées.

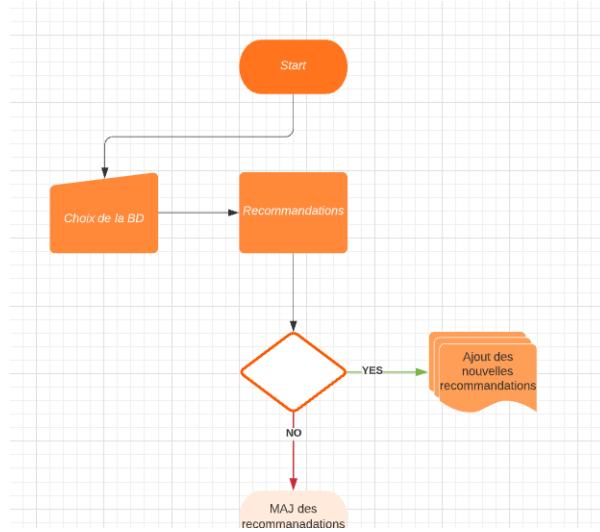


FIGURE 27 – Expérience utilisateur

Pour bien déterminer si l'application correspond à nos attentes (à celles des utilisateurs finaux), nous effectuons un certains nombre de tests, dits fonctionnels. Premièrement j'ai vérifié les routes de mon API

Route	Usage	méthode	nécessité d'être logué : oui / non
/ et / home	racine de l'application	GET	oui
/login	gestion du login	GET et POST	non
/signup	gestion du sign up	GET et POST	non
² /logout	gestion du log out	GET	oui
/faire_reco	Gestion des recommandations	GET et POST	oui
/tableau_bord	Gestion des critiques	GET et POST	oui

Les templates sont un peu différents, mais leur fonctionnement reste très proche des routes qui y sont liées

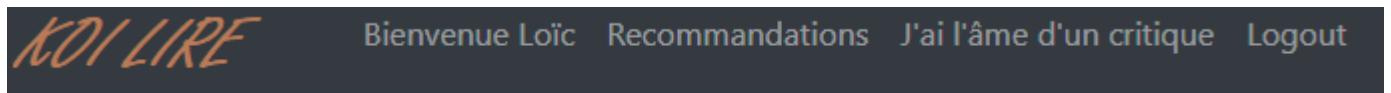


FIGURE 28 – Menu utilisateur

En résumé l'utilisateur peut :

8.1.1 Se logger

Sign Up

Email Address
curbierel@gmail.com

First Name
Loïc

Password

Password (Confirm)

Submit

FIGURE 29 – Page login

Les sécurités standards ont été mises, :

- nombre de caractères pour le MdP
- adéquation du MdP et de sa confirmation
- vérifier que l'adresse mail contient bien @
- unicité de cet email.

Le HTML et le python de l'API Page annexe 28 à 44

8.1.2 Gérer sa collection

Vous avez peut-être ce titre: Astérix, tome 22 : La Grande Traversée

Voulez vous une proposition (en temps réel donc long)

Oui Non Je possède cette BD



FIGURE 30 – Page principale de l'utilisateur

Cette page permet à l'utilisateur de sauvegarder sa collection. J'ai mis en place une sécurité, le lecteur ne peut pas posséder plusieurs fois le même tome (totalement identique), si l'édition est différente pour l'API, la BD est différente.

8.1.3 Proposition automatique de nouvelle BD

Une nouvelle option que l'on active ou non (qui nécessite du temps de calcul), propose un titre que le lecteur pourrait avoir dans sa bibliothèque (cette recommandation se fait sur le titre). A l'inverse de la fonction de recommandation, elle fonctionne en temps réel. Elle utilise le NLP (les lemmas pour traiter ensemble tous les termes de la même famille et les stop_word ou mot ne donnant pas plus de sens à une phrase) alors que mon autre système de recommandations fonctionnent avec les transformers.

8.1.4 Obtenir des Recommandations

Bienvenue cicou45 Recommandations J'ai l'âme d'un critique Logout About

Recommandations par la description

J'aime un titre ressemblant à

Voir nos propositions

Nous vous recommandons pour la BD Les Aventures de Tintin, tome 03 : Tintin en Amérique:

Lenny Valentino, tome 1 : Poison Town	Mafia Story, Tome 6 : La chute de Lucky Luciano	Agatha Christie, Tome 8 : Le Meurtre de Roger Ackroyd	Il était une fois en France, tome 5 : Le Petit Juge de Melun	Une Aventure des Brigades du Tigre : Ni Dieu, Ni Maître
---------------------------------------	---	---	--	---

Les recommandations qui m'ont été faites du 01/12/2021 au 28/03/2022 Afficher les recommandations pour la période choisie

Trolls de Troy, tome 4 : Le Feu occulte
Trolls de Troy, tome 3 : Comme un vol de pétaures
Trolls de Troy, tome 1 : Histoires trolles
Tilda sur les toits, tome 1 : Le masque et la fée
Le Pays des elfes - Elfquest, tome 1 : Les Rescapés du feu
Trolls de Troy, Tome 16 : Poils de trolls

FIGURE 31 – Recommandations

C'est à partir de cette page, que l'on a accès aux recommandations ; ce moteur utilise les fonctions vues précédemment, et sauvegarde toutes les prédictions. A chaque fois qu'une BD est recommandée, un compteur interne est incrémenté ; donc une BD peut-être recommandée plusieurs fois. Les 5 premières recommandations (celles qui sont le plus recommandées) sont mises en valeur.

8.2 Les Tests fonctionnels

1. En tant d'utilisateur du service
 - Création d'un compte

The screenshot shows a web browser window with the title 'Sign Up'. The URL is 'https://127.0.0.1:5000/sign-up'. The page has a dark header with the KOI-LIRE logo and navigation links for 'Login', 'Sign Up', and 'About'. The main content area is titled 'Sign Up' and contains four input fields: 'Email Address' (with value 'test@test.fr'), 'First Name' (with value 'test'), 'Password' (with placeholder '*****'), and 'Password (Confirm)' (with placeholder '*****'). Below the fields is a blue 'Submit' button.

FIGURE 32 – Inscription
— se connecter à sa session et gérer sa bédéthèque

The screenshot shows a web browser window with the title 'Chez moi'. The URL is 'https://127.0.0.1:5000/chez-moi'. The page has a KOI-LIRE logo with the tagline 'QUELLE BD JE VAIS LIER MAINTENANT'. Below the logo is a 'Login' form with fields for 'Email Address' (value 'test@test.fr') and 'Password' (placeholder '*****'). A blue 'Login' button is at the bottom.

FIGURE 33 – Page lors de la connection

The screenshot shows a web browser window with the title 'Chez moi'. The URL is 'https://127.0.0.1:5000/chez-moi'. A banner at the top says 'McAfee | True Key™ Votre identifiant 127.0.0.1 a été enregistré.' On the right, there are buttons for 'Ne jamais enregistrer' and 'Ne pas enregistrer maintenant'. A green status bar message says 'Vous avez peut-être ce titre: Les aventures de Tintin, tome 20 : Tintin au Tibet'. Below this, a message in red text says 'Si l'utilisateur désire une suggestion, elle apparaît ici'. There is a checkbox 'Oui Non' and a button 'Je possède cette BD'. To the right is a thumbnail image of a comic book cover titled 'TINTIN AMÉRIQUE'.

FIGURE 34 – La collection de l'utilisateur

- avoir une recommandation

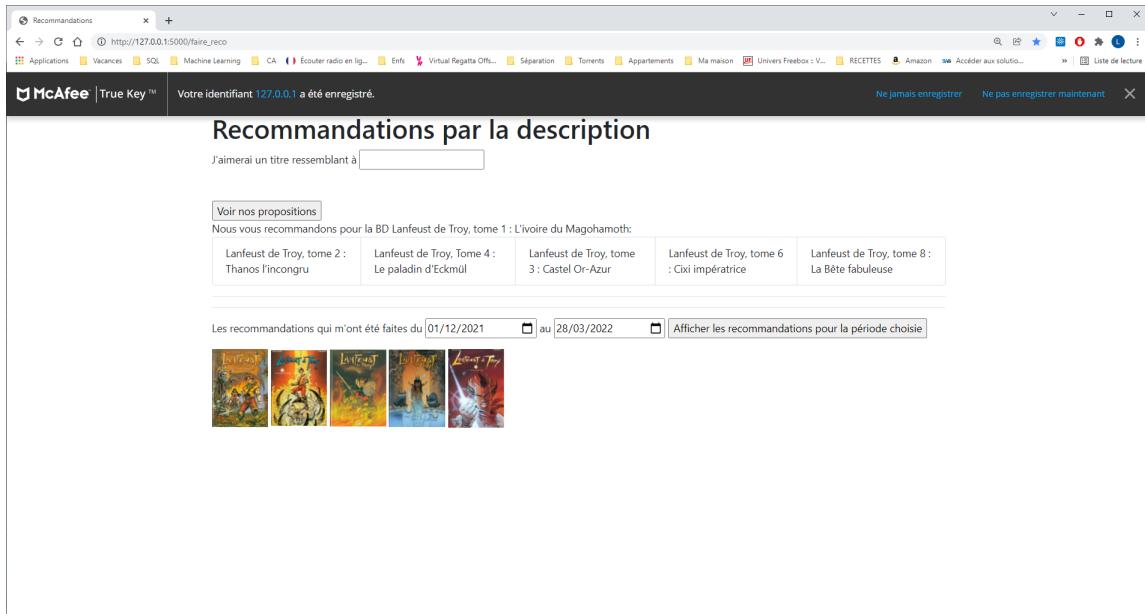


FIGURE 35 – Que pourrais-je lire

- faire une critique



FIGURE 36 – Donner son avis

2. En tant d'administrateur
 - Suppression de n'importe quel compte
 - accès au monitoring de l'application

8.3 Derrière, on trouve

8.3.1 Architecture du website

```
| database.db  
+ flask_monitoringdashboard.db  
+ main.py  
+ website/  
|   + static/  
|   |   + index.css  
|   |   + index.js  
|   |   + KOILIRE.png  
|   |   + logo.PNG  
|   |   + scrolltop.png  
|   + templates/  
|   |   + about.html  
|   |   + auth.html  
|   |   + faire_reco.html ..... Page des recommandations  
|   |   + home.html ..... Gestion de la bibliothèque du lecteur  
|   |   + models.html  
|   |   + monitoring.html ..... Gestion des critiques  
|   |   + tableau_bord.html  
|   + about.py  
|   + auth.py  
|   + faire_reco.py  
|   + fonction.py .2 home.py  
|   + models.py  
|   + monitoring.py  
|   + tableau_bord.py
```

8.3.2 Tests Unitaires

Avant de mettre mon API en production, j'ai effectué un certain nombre de tests dits unitaires. Pour cela j'ai utilisé la bibliothèque pytest. J'ai donc testé la création d'utilisateur, leur suppression ; j'ai aussi testé la création et la suppression de recommandations⁹

8.4 Le monitoring et alerting

8.4.1 Monitoring

Montrer un site est de plus important, et l'usage d'applications dédiées et performantes est devenu une nécessité ces dernières années ; c'est pourquoi j'ai choisi d'utiliser flask dasboard monitoring. Ce dashboard permet de surveiller les performances et l'utilisation du site, de collecter les demandes/requêtes aberrantes. Le tableau de bord est très simple et facile à prendre en main. Le niveau de logs s'échelonne de 0 à 4 (0 :Pas de log et 4 on loggue tout) Il est possible d'avoir plusieurs types de graphique et d'utiliser le plus pertinent pour nous, par exemple, dans mon cas j'ai choisi "API performance" qui permet voir le temps de réponse pour chaque graphique. Cela permettra d'optimiser le code de l'API pour avoir un temps de réponse correct.

9. Test unitaire des script Annexe p.44 et 45

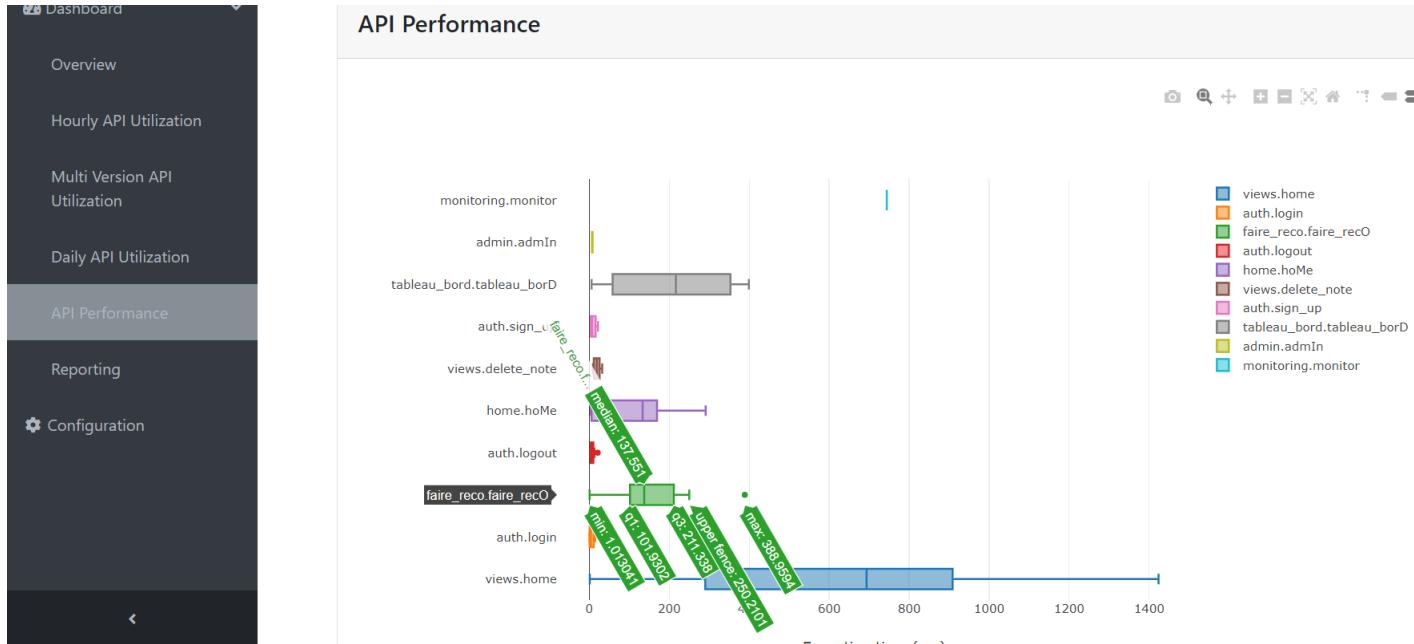


FIGURE 37 – Monitoring de l'API

8.4.2 Alerting

Pour l'instant cette API est encore à l'état de POC. Mais quand elle sera mise en production, son système d'alerting sera déjà prêt (On notera tout de même que c'est erreur serveur celle-ci ne sera pas remontée) Pour l'instant l'alerting a été mis en place comme cela :

1. utilisation du module loggin (pour la gestion/session des utilisateurs)
2. Nous utilisons aussi un SMTPHandler qui va nous permettre de créer un server e-mail qui ne sera actif au niveau Error minimum.
3. Puis unFileHandler, configuré également sur le même niveau de surveillance, cet objet écrira dans un errorlog.txt les logs en cas d'erreur du back-end Flask.

Cette suite d'implémentation de modules permettra à l'administrateur du site d'être averti en cas de gros/graves incidents.

```
from flask_mail import Mail, Message
...
mail= Mail(app)
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 993
app.config['MAIL_USERNAME'] = 'curbierel@gmail.com'
app.config['MAIL_PASSWORD'] = 'XXXXXXXXXX'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
@app.route("/mail")
def index():
    msg = Message('Server Error', sender = 'curbierel@gmail.com', recipients = ['lcurbiere@gmail.com'])
    msg.body = "Somethong went wrong - please check logs"
```

```
mail.send(msg)
return "Sent"
```

9 Pour conclure et dans un futur proche

9.1 Conclusion

Pour conclure ce projet, j'ai utilisé la majorité de mon temps à la recherche de sources fiables. Une fois ceci fait j'ai pu travailler sur mes recommandations. Un autre point qui me semblait important, était la textitsérendipité et voulais mettre ce point en avant, pour ne pas faire des recommandations trop stéréotypées.

9.2 Axes d'amélioration

Cette API est un MVP, pour avoir un outil plus complet, il faudra :

1. Remplacer la base de données sqlite par mySQL par exemple
2. Contacter babelio (site d'où viennent les données) pour un éventuel partenariat
3. Automatiser la mise à jour de la BdD (prévu mais pas encore fait)
4. Mise en place de recommandations collaboratives avec les critiques
5. Modification du monitoring avec l'utilisation de Pawpatrol à la place de flask dashboard manager.

Table des figures

1	Site Web d'où sont extraites les données	3
2	Planning prévisionnel	4
3	Planning SCRUM prévisionnel	5
4	Tableau kanban du 1er sprint	5
5	Découpage fonctionnel	6
6	df des BD listées	8
7	df des utilisateurs du site babelio	8
8	df des critiques de BD	9
9	Les mots clés les plus représentés	10
10	Notes des BD	10
11	Les mots clés les plus présent dans les titres	11
12	Les auteurs qui produisent beaucoup	11
13	Les mots clés les plus présent dans les critiques	12
14	Les plus grandes critiques	12
15	Les critiques de jyrille	13
16	Représentation de mon modèle	16
17	Le jeu de données avant l'utilisation de BERT	16
18	Les données vectorisées	17
19	Aide pour remplir les étagères	17
20	Représentation de la cosine	18
21	Les recommandations en mode console	18
22	Plus on a de critiques, meilleur est le modèle	21
23	MCD by MsqliWorkbench	22
24	Ce qui reste à faire à mi-parcours	24
25	Réponse en mode graphique	25
26	Vue d'ensemble de l'API	25
27	Expérience utilisateur	26
28	Menu utilisateur	26
29	Page login	27
30	Page principale de l'utilisateur	27
31	Recommandations	28
32	Inscription	29
33	Page lors de la connection	29
34	La collection de l'utilisateur	29
35	Que pourrais-je lire	30
36	Donner son avis	30
37	Monitoring de l'API	32