Rapport du projet NoSQL : Scrapping du site Allociné

ESGI
école supérieure de génie informatique

Sayah MEDELLEL
Moussa CAMARA
Oussama ABDELHEDI

Le rapport final

ESGI école supérieure de génie informatique
Sayah MEDELLEL, Moussa CAMARA, Oussama ABDELHEDI

L’équipe du projet :

Sayah MEDELLEL, Moussa CAMARA, Oussama ABDELHEDI

Période :

Semestre 2 (2021)

Établissement :

ESGI, École Supérieure de Génie Informatique

Sujet :

Scraping du site Allociné

Professeur en charge :

Mr. Jean-Baptiste KOUEK

Sommaire

Ce rapport est composé :

- D'une partie présentation générale du projet, avec une introduction, des points non résolus, des difficultés rencontrées et d'un bilan global
- D'une annexe technique composée des dossiers concepteur et utilisateur
- D'une annexe informelle composée d'une bibliographie et d'un glossaire

Le dossier concepteur présente la conception du projet ainsi qu'une analyse de l'application regroupant plusieurs éléments :

- Une description de la structure du projet
- Une description des méthodes principales
- Une description des choix d'implémentation et d'autres détails techniques

Le dossier utilisateur regroupe l'ensemble des informations permettant de comprendre l'installation et l'utilisation de l'application.

La bibliographie regroupe les références des documents qui nous ont aidés pour réaliser le projet et le glossaire regroupe le vocabulaire spécifique et technique utilisé.

Introduction

Dans le cadre de la réalisation d'un projet en NoSql, nous avons réalisé une application qui permet de récupérer les données du site Allociné et de les analyser. Allociné est un site de référence en terme de séries TV, rassemblant plus de 10 millions de visiteurs par mois. Le site étant assez vaste, nous nous sommes focalisés sur un thème qui nous intéressait tous : les séries télévisées. En outre, un très grand nombre d'informations concernant les séries est disponible sur Allociné, nous avons donc limité les données à récupérer :

- Aux informations principales de la séries (titre, date de sortie …)
- Aux avis de la série (commentaires, notes …)

Le projet est scindé en trois parties :

- Data Scraping : automatisation d'un processus permettant de recueillir les données de plusieurs séries et de les regrouper dans plusieurs fichiers structurés
- Data Processing : mise en place de plusieurs processus permettant d'extraire et/ou de rassembler des données brutes des fichiers structurés et des données stockées dans une base de données en vue de produire des informations utilisables
- Data Visualisation : représentation graphique des données afin de communiquer les informations de manière claire et efficace

Pour réaliser ce projet, l'équipe est composée de trois membres :

- Sayah MEDELLEL
- Moussa CAMARA
- Oussama ABDELHEDI

Points non résolus

Globalement, les objectifs du projet ont été atteints, notamment pour les parties concernant l'extraction et le traitement des données. La visualisation des données a été un réel sujet de réflexion, les points pour aborder cette partie étant assez vastes. Nous avons tout d'abord pensé à utiliser Elasticsearch et Kibana afin de créer un moteur de recherche avec indexation afin de générer des graphiques spécifiques, cependant le temps nous a limité à nous pencher sur cette option.

Difficultés rencontrées

Nous avons rencontré plusieurs difficultés (humaines et techniques) pour la réalisation du projet. La crise sanitaire de la Covid-19 et le confinement imposé par le Gouvernement ont eu un vrai impact, empêchant les sorties et donc les rencontres à l'extérieur. Nous étions donc forcés de collaborer à distance, ce qui a provoqué quelques désagréments.

Tout d'abord, on peut évoquer le manque de communication, parfois nous ne recevions pas les messages en temps et en heure, ce qui est un inconvénient lorsqu'il faut modifier le travail. L'absence physique et les échanges à distance ont favorisé les pertes d'informations et ont pu créer une sorte de manque de cohésion. Les idées étaient plus difficiles à exprimer et il s'ensuivit des incompréhensions entre les membres de l'équipe.

Enfin, nous avons fait face à plusieurs problèmes techniques (notamment au niveau de la conception). Concernant les installations, nous avons également eu quelques problèmes à installer et configurer correctement certains outils. Il était nécessaire de comprendre tous les éléments qui permettent l'installation et la configuration de ces outils, dont le fait de trouver les dernières libs à jour (précompilés).

Afin de surmonter toutes ces difficultés nous avons dû nous entraider, communiquer plus régulièrement entre nous à travers les systèmes de messageries et nous aider des différents documents qui étaient à notre disposition (cours, sites web).

Bilan

La réalisation de ce projet a permis au groupe d'acquérir des compétences sur la programmation en python et notamment dans l'extraction de données mais également sur la capacité à donner un sens aux données (du traitement des données à leur visualisation). Ce projet nous a également permis de développer nos compétences sur plusieurs outils et technologies qui étaient encore inconnu pour nous.

Annexe technique

I) Dossier concepteur

Introduction

Ce document explique la phase d'avant-projet, phase qui va permettre de définir les objectifs clairs de l'application et notamment la manière dont celle-ci va être réalisée. Cette phase est nécessaire afin de comprendre au mieux le sujet posé et de l'aborder convenablement. Il est clair que cela va permettre de simplifier les étapes de réalisation par la suite. Globalement, elle va permettre de structurer, organiser et planifier le projet.

Spécification détaillée de la structure du système

Afin de comprendre la méthode de conception appliquée, nous avons effectué une description détaillée du système, réparti en un aspect technique et un aspect complémentaire.

A) Aspect technique

L'application est composée d'un ensemble de fonctions qui permettent le scraping d'Allociné. Globalement, les fonctions vont permettre à un utilisateur :

- De récupérer certaines données du site Allociné
- Les transformer et les insérer dans une base de données
- Les exporter pour porter dessus certaines analyses intelligentes

B) Aspect complémentaire

Globalement, le but d'une telle structure est de fournir un programme qui répond à plusieurs critères essentiels : flexibilité, durabilité et adaptabilité. D'ailleurs le programme conçu permet d'assurer complètement l'aspect de la dynamicité. En outre, le fonctionnement du programme et les résultats s'adaptent par rapport aux présentes dans le site. La date n'influe pas sur le comportement du programme, ce qui encore une fois est intéressant pour l'utilisateur.

C) Outils et technologies

1) Techniques

Le langage utilisé pour réaliser le programme de récupération des données du site Allociné est le Python. Des requêtes SQL ont également créées afin d'établir des tests sur les données présentes dans la base de données MongoDB.

2) Ressources immatérielles

Plusieurs ressources ont été utilisés, notamment des logiciels et applications permettant :

- Le développement du programme (IntelliJ, Visual Studio Code)
- Le développement du programme et l'analyse de données (Jupyter Notebook)
- L'organisation de la réalisation du projet (Gantt Project)
- La conception de la base de données (JMerise)
- Le stock des données (MongoDB)

D) Comportement

Nous avons vu précédemment que le programme a été conçu de manière à assurer l'aspect de la dynamicité. Pour assurer un tel aspect, le programme a été conçu avec une logique d'automatisation, limitant ainsi les interactions entre l'utilisateur et le programme. Nous allons détailler le fonctionnement du programme et des fonctions. Pour rappel, le projet est scindé en trois parties :

- Data Scraping
- Data Processing
- Data Visualisation

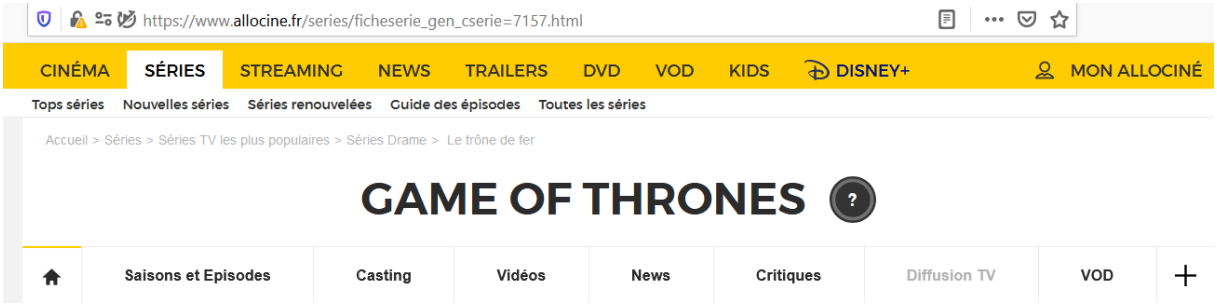
Chaque partie présente une méthodologie et un mode de fonctionnement différent.

1) Data Scraping

L'objectif dans cette partie est de récupérer les données concernant les séries et les avis des utilisateurs. Sur Allociné, chaque site possède sa propre fiche.

Figure 1 : Présentation de la fiche de la série Game Of Thrones

En-tête de la fiche :



Description de la fiche :

Notes de la fiche :

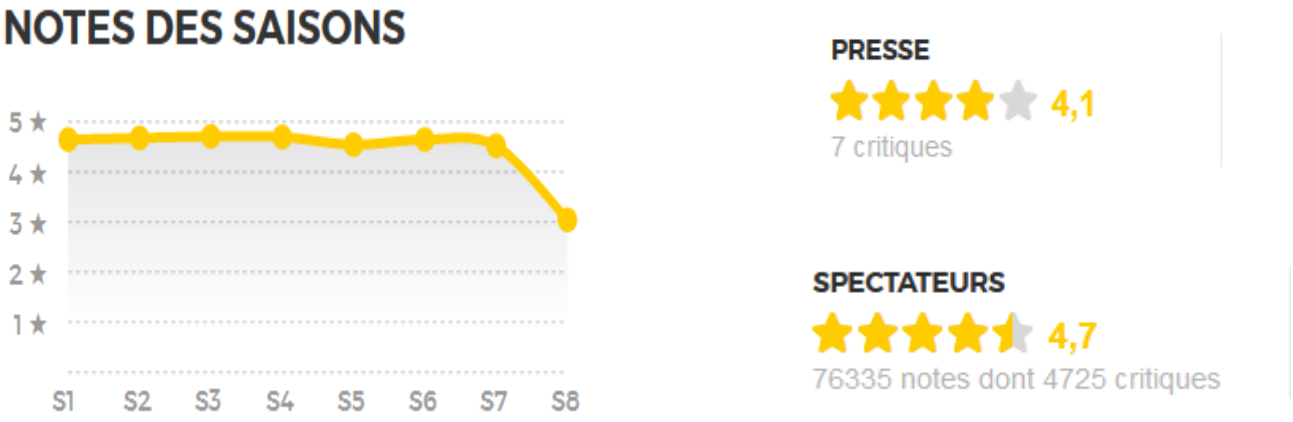


Figure 2 : Présentation de la fiche des avis de la série Game Of Thrones

En-tête de la fiche des avis :

Avis d'un spectateur :

HawkMan
Suivre son activité
67 abonnés
Lire ses 764 critiques

Critique de la saison 7

4,5 Publiée le 21 mars 2020

Avant dernière saison, plus courte, le combat final contre les Marcheurs Blancs approche et les clans se forment pour la défense du royaume.
Bref : un rythme plus lent, des dialogues qui s'étirent, des personnages qui doutent, superbe saison !

L'objectif est donc de récupérer certaines données des deux fiches. Les données des avis seront contenues dans une liste de dictionnaires qui sera elle-même contenu dans un fichier json.

Figure 3 : Tableau représentant la liste des données à récupérer selon les fiches

Fiche	Données à récupérer
Série	Url, titre, date, genres, auteurs, note des spectateurs / de la presse, acteurs, pays d'origine, nombre de saisons et d'épisodes, synopsis
Avis	Url, url de la série, texte, date, pseudo de l'auteur, note, label de la note

Dans ce dossier nous avons les scripts :

generate_series_urls.py : Récupère la liste des séries à scraper pour chaque page du site : https://www.allocine.fr/series-tv/.

-> Génère plusieurs fichiers dans le dossier urls_series_collected.

aggregate_urls.py : Agrège tous les fichiers obtenus avec le script generate_series_urls.py en un fichier json. -> Génère un fichier dans le dossier urls_series_aggregated.

filter_urls.py : Filtre les urls afin d'obtenir deux fichier json : un avec la liste des fiches decriptives à scraper et un autre avec la liste des pages avis à scaper.

-> Génère un fichier json dans le dossier urls_products_to_collect et un fichier dans le dossier urls_reviews_to_collect

products_collector.py : Contient l'ensemble des méthodes permettant de collecter les infos d'une fiche descriptive.

collect_product.py : Script de test qui collecte les infos d'une seule fiche descriptive. (Permet donc de tester les méthodes de products_collector.py).

-> Génère un fichier json dans le dossier *product*.

collect_products.py : Collecte les infos des fiches descriptives contenues dans la liste obtenue avec le script filter_urls.py.

-> Génère plusieurs fichiers json dans le dossier products.

reviews_collector.py : Contient l'ensemble des méthodes permettant de collecter les infos d'une page avis.

collect_review.py : Script de test qui collecte les infos d'une seule page avis. (Permet donc de tester les méthodes de reviews_collector.py)

-> Génère un fichier json dans le dossier review.

collect_reviews.py : Collecte les infos des pages avis contenues dans la liste obtenue avec le script filter_urls.py

-> Génère plusieurs fichiers json dans le dossier reviews.

aggregate_files.py : Agrège le contenu des dossiers products et reviews.

-> Génère un fichier dans le dossier merged_products et un fichier dans le dossier merged_reviews.

2) Data Processing

L'objectif de cette partie est d'appliquer un traitement sur les données récupérées avec le scraping. Les données sont ainsi stockées dans le SGBD MongoDB.

Figure 4 : Présentation du cluster

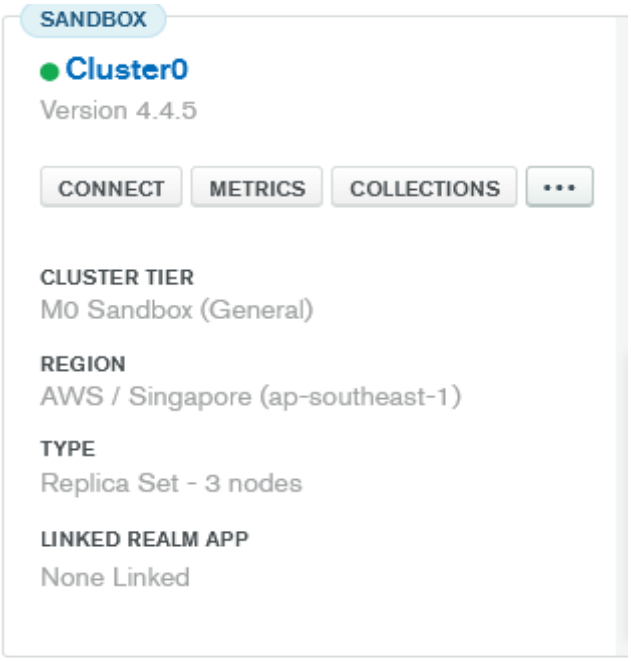
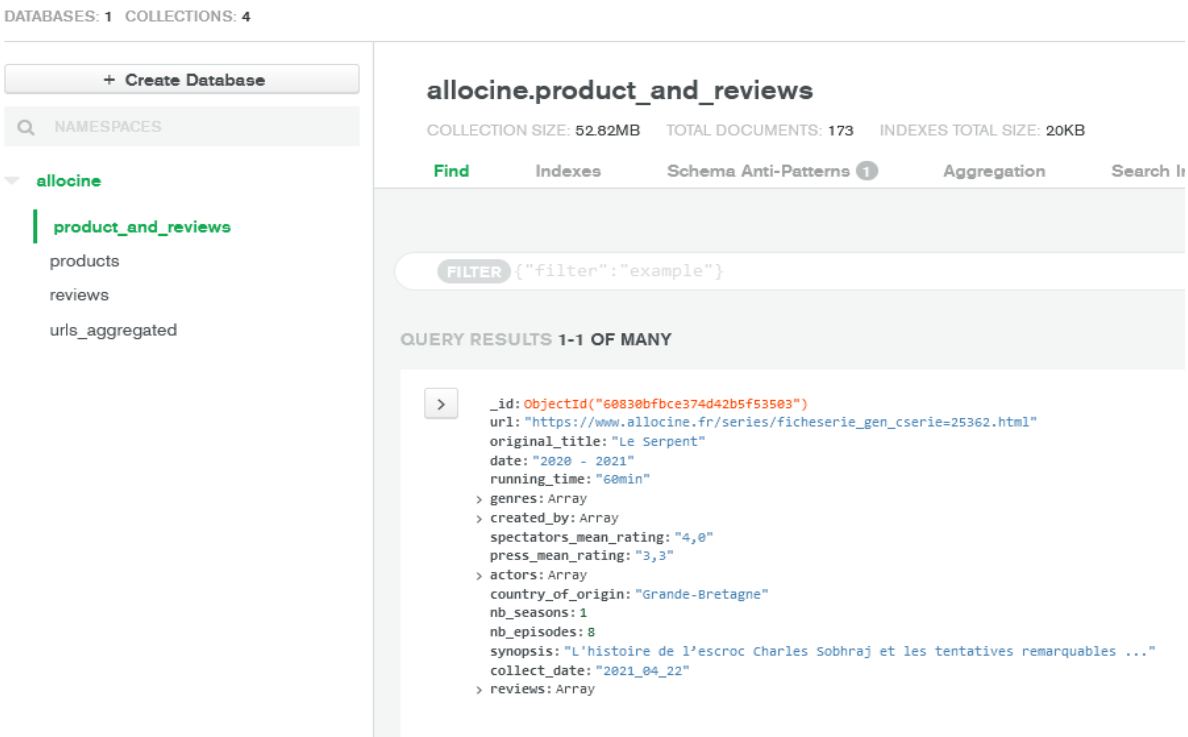


Figure 5 : Présentation de la base de données et des collections



L'accès au compte est disponible avec les informations suivantes :

Identifiant : marcmc928@gmail.com
Mot de passe : allocine123

Nous avons dans un second temps créer et configurer le cluster puis nous y avons ajouté notre base de données ainsi que 4 collections (*cf figure 5*).

- product_and_reviews : collection où nous avons les données sur les séries ainsi que les avis qui leurs sont associées
- products : collection des séries
- reviews : collection sur les avis des séries
- urls_aggregated : collection où l'on stock le nom de la série, son url ainsi que l'url des avis.

Nous avons dans le dossier **data-processing-allocine** trois notebook et 3 dossiers, chacun des dossiers contenant un fichier json obtenu lors de l'étape du scraping.

Concernant les trois notebooks :

Import_files_to_database.ipynb, comme son nom l'indique, importe les données dans la base de données. Tout d'abord nous faisons les import des libs puis l'import des chemins (les dossier products, reviews et urls_aggregated). Ensuite nous nous connectons à la base de données et on initialise la base de données et les collections de cette dernière. Enfin on ouvre chaque fichier json puis on insère les données dans les différentes collections.

aggregate_products_and_reviews.ipynb fusionne deux fichiers en un dans la base de données. De même, on insère les imports et les fichiers nécessaires pour enfin effectuer la jointure entre le dictionnaire des séries et celui des avis avant d'injecter les données dans la collection product_and_reviews.

Nous avons fait une copie des deux notebooks dans des fichiers python. Nous avons aussi le troisième notebook **analyze_reviews_with_nlp.ipynb** qui lui va permettre d'analyser les avis des séries et permettre de dire si un avis est positif ou négatif. Cette étape est effectué avec le Natural Language Processing (nlp).

3) Data Visualisation

L'objectif de cette partie est de modéliser les données disponibles dans la base de données. Globalement, les données stockées représentent les informations relatives aux séries et à leurs avis. Rappelons que plusieurs séries et plusieurs avis par série sont disponibles, un grand nombre d'analyses peut ainsi être effectué afin de récupérer les données pertinentes. Pour ce faire, nous avons décidé d'établir :

- un Dashboard regroupant plusieurs graphiques modélisant ces données
- un fichier contenant des requêtes à implémenter afin d'alimenter le Dashboard

Le Dashboard est implémenté avec Jupyter Notebook.

II) Dossier utilisateur

Introduction

Le dossier utilisateur permet à l'utilisateur de lui apporter tous les renseignements nécessaires et suffisants pour une bonne utilisation et compréhension de l'application. Il explique à l'utilisateur comment utiliser le programme, corriger les possibles erreurs et comprendre les résultats du programme.

Présentation de l'application

A) Fonctionnalités

Le produit final est une application qui permet de scraper le site Allociné, les formatter et enfin les analyser efficacement à l'aide d'outil de visualisation. Pour obtenir les informations sur les séries, l'utilisateur lance le programme qui affichera les résultats détaillés ou bien les possibles erreurs.

B) Mode d'utilisation

Les données concernant les séries et les avis sont récupérés grâce à l'exécution de deux scripts en simultané. Une fois l'exécution terminé, plusieurs fichiers sont créées. Un autre script permet « l'assemblage » de ces fichiers, formant ainsi un fichier pour l'ensemble des séries et un autre pour les avis.

Ainsi, pour utiliser le scraper il faut lancer les scripts dans l'ordre suivant :

- generate_series_urls.py
- aggregate_urls.py
- filter_urls.py
- collect_products.py
- collect_reviews.py
- aggregate_files.py

Pour tester le scraper il suffit de lancer les scripts :

- collect_product.py
- collect_review.py

Les parties concernant la data Processing et visualisation s'effectuent sur Jupyter Notebook et MongoDB. D'une part des requêtes ont été mises en place afin d'effectuer la jointure entre les séries et les avis, d'autre part l'exécution de partie de codes permettent la génération de graphiques.

L'ensemble des fichiers et des méthodes utilisés ont été commentés afin de garantir la compréhension et la bonne utilisation des programmes.

Annexe informelle

Bibliographie

- Cours de Mr. KOUEK
- https://stackoverflow.com
- https://github.com/TheophileBlard/french-sentiment-analysis-with-bert
- https://plotly.com/python/
- https://docs.mongodb.com/manual/
- **Repo git du modèle NLP :** https://github.com/TheophileBlard/french-sentiment-analysis-with-bert
- https://huggingface.co/tblard/tf-allocine

Glossaire

Libs : Library (Bibliothèque logicielle) : Collection de routines, qui peuvent être déjà compilées et prêtes à être utilisées par des programmes.

Package : Paquet : archive (fichier compressé) comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel.

NLP (traitement automatique des langages) : domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle, qui vise à créer des outils de traitement de la langue naturelle pour diverses applications.