École Nationale Supérieure d'Informatique et d'Analyse des Systèmes

Mai 2024

Data Applications

Pipeline d'Analyse des Commentaires YouTube : De la Collecte des Données au Dashboarding

---

Elaboré par :
BEKKAI ChamssDoha, BI&A
BENALI Amin, BI&A
GHAZOUAN Oumaima, BI&A
YOUB Omar, BI&A

Encadré par :
Mrs.Nourddine KERZAZI

- Année scolaire : 2023/2024 -

RÉSUMÉ

Le projet vise à mettre en place un pipeline complet pour l'analyse des commentaires YouTube. Il commence par la collecte des données à partir de diverses chaînes en utilisant l'API YouTube, combinant des techniques de web scraping et des requêtes API. Ensuite, les données collectées sont nettoyées pour éliminer les valeurs manquantes, les caractères spéciaux et autres incohérences, garantissant ainsi leur qualité pour les étapes suivantes. Une fois nettoyés, les commentaires sont analysés pour extraire des informations pertinentes telles que les tendances, les sentiments et les sujets discutés. Enfin, un tableau de bord interactif est construit à l'aide de Dash, un framework Python, permettant une exploration facile et dynamique des résultats de l'analyse.

ABSTRACT

This project aims to establish a comprehensive pipeline for analyzing YouTube comments. It encompasses multiple phases, ranging from data collection to result visualization. Initially, YouTube comments are extracted from various channels using the YouTube API, leveraging a combination of web scraping techniques and API queries. Subsequently, the collected data undergoes thorough cleaning and preparation to eliminate missing values, special characters, and other inconsistencies, thereby ensuring its quality for subsequent analysis. Following data preprocessing, the cleaned comments are subjected to analysis to extract relevant information such as trends, sentiments, and discussed topics. Finally, an interactive dashboard is developed using Dash, a Python framework, facilitating easy and dynamic exploration of the analysis results.

TABLE DES MATIÈRES

INTRODUCTION GÉNERALE

Contexte et problématique

L'analyse des commentaires YouTube offre une opportunité précieuse de comprendre les opinions et les sentiments des utilisateurs vis-à-vis des contenus publiés sur cette plateforme. En exploitant les vastes quantités de données générées par les utilisateurs, nous pouvons extraire des insights significatifs pour les créateurs de contenu, les spécialistes du marketing, et les chercheurs. Ce projet vise à mettre en place une pipeline d'analyse des commentaires YouTube, allant de la collecte des données à la visualisation des résultats dans un tableau de bord interactif.

Objectifs du projet

1. **Collecte des Données :** Utiliser l'API YouTube pour extraire les commentaires des vidéos de plusieurs chaînes populaires.
2. **Nettoyage et Préparation des Données :** Appliquer des techniques de nettoyage pour garantir la qualité des données et les préparer pour l'analyse.
3. **Intégration des Données :** Centraliser les commentaires extraits dans un fichier unique pour faciliter les analyses ultérieures.
4. **Analyse et Résumé des Commentaires :** Utiliser des techniques de traitement du langage naturel (NLP) pour analyser les sentiments et résumer les commentaires.
5. **Synthèse Audio :** Convertir les résumés de commentaires en fichiers audio pour une accessibilité accrue.
6. **Visualisation des Résultats :** Développer un tableau de bord interactif pour visualiser les insights extraits des commentaires.

CHAPITRE 1

DOCKERISATION DU PIPELINE

1.1 Introduction

Docker est un outil de conteneurisation qui permet de créer, déployer et exécuter des applications de manière isolée et reproductible. Dans ce chapitre, nous allons explorer comment Docker est utilisé pour encapsuler les composants de notre pipeline d'analyse des commentaires YouTube. Nous aborderons les concepts de base de Docker, l'utilisation de Dockerfile et Docker Compose pour la gestion des conteneurs, et nous présenterons en détail le fichier docker-compose.yaml utilisé dans ce projet.

1.2 Présentation de Docker

Docker facilite la création et la gestion de conteneurs, qui sont des environnements légers et portables pour exécuter des applications. Les conteneurs incluent tout ce dont une application a besoin pour fonctionner : le code, les bibliothèques, les dépendances et les variables d'environnement, garantissant ainsi que l'application fonctionne de manière cohérente, quel que soit l'environnement où elle est déployée.

FIGURE 1.1 – Logo de Docker

1.2.1 Historique de Docker

Docker Inc a été fondée par *Solomon Hykes, Kamel Founadi et Sebastien Pahl* au cours du groupe d'incubation de startups Y Combinator Summer **2010**. L'entreprise fut lancée en **2011**.

Elle fut aussi l'une des 12 startups de la première cohorte de Founder's Den. Le projet fut initié par Solomon Hykes en France, sous la forme d'un projet interne de l'entreprise de plateforme en tant que service dotCloud.

En **2013**, Docker fut présentée au public à Santa Clara dans le cadre de la PyCon. Le logiciel a été lancée en open-source en mars **2013**. A l'époque, LXC était utilisé comme environnement d'exécution par défaut, avant d'être remplacé un an plus tard avec la version 0.9 de Docker par son propre composant libcontainer écrit en langage Go.

Au fil des années, Docker a noué de nombreux partenariats stratégiques avec les géants du Cloud et de l'IT : Red Hat en **2013**, Microsoft , iBM et Amazon Web Services en 2014, Oracle en **2015**, mais aussi Cisco, Google ou Huawei.

Depuis **2016**, Docker peut être utilisé nativement sur Windows 10. La même année, une analyse de LinkedIn révèle que le nombre de mentions du logiciel sur les profils des utilisateurs a augmenté de 160

1.2.2 Avantages de Docker

a **Isolation :** Chaque conteneur fonctionne de manière indépendante, ce qui évite les conflits entre les applications.

b **Portabilité :** Les conteneurs peuvent être déployés sur n'importe quel système supportant Docker, facilitant le transfert des applications entre différents environnements (développement, test, production).

c **Efficacité :** Les conteneurs partagent le noyau du système d'exploitation hôte, ce qui les rend plus légers et plus performants que les machines virtuelles.

1.3 Utilisation de Docker pour l'encapsulation des composants du pipeline

Dans notre projet, Docker est utilisé pour encapsuler les composants clés du pipeline d'analyse des commentaires YouTube, notamment Airflow, PostgreSQL, et Redis. En utilisant Docker, nous nous assurons que chaque composant est isolé et configuré de manière optimale pour fonctionner ensemble sans conflits.

Composants Docker du pipeline :

— **Airflow** : Utilisé pour orchestrer et automatiser le workflow du pipeline.

— **PostgreSQL** : Base de données pour stocker les métadonnées d'Airflow.

— **Redis** : Utilisé comme backend pour la file d'attente des tâches d'Airflow.

1.4 Dockerfile et Docker Compose pour la gestion des conteneurs

Pour chaque composant de notre pipeline, nous définissons un Dockerfile qui spécifie les dépendances et les configurations nécessaires. Nous utilisons également Docker Compose pour orchestrer et gérer l'ensemble des conteneurs. Docker Compose nous permet de spécifier les volumes partagés, les réseaux, et les dépendances entre les services, facilitant ainsi le déploiement et la gestion de notre pipeline.

1.4.1 Dockerfile

Le Dockerfile est un fichier de configuration qui définit les étapes nécessaires pour construire une image Docker. Pour notre projet, nous avons envisagé d'utiliser un Dockerfile qui spécifie les dépendances et les configurations nécessaires tout au long des différentes étapes présentes dans notre projet.

Dans notre cas, nous avons utilisé un Dockerfile contenant Airflow et les dépendances supplémentaires nécessaires à notre pipeline, telles que Torch, Transformers, Pandas, SciPy et Dash. Ces bibliothèques seront utilisées par la suite dans les modèles d'analyse des sentiments.

1.4.1.1 Torch

PyTorch est une bibliothèque optimisée pour l'apprentissage profond utilisant des GPU et des CPU. Dans notre projet de pipeline d'analyse des commentaires YouTube, PyTorch est utilisé comme dépendance essentielle pour les tâches de summarisation et d'analyse des commentaires. Le Dockerfile inclut l'installation de PyTorch pour permettre l'exécution des modèles d'apprentissage profond nécessaires à ces analyses.

FIGURE 1.2 – Logo de torch

1.4.1.2 Transformers

ransformers offre des milliers de modèles pré-entraînés pour accomplir des tâches sur différents types de données tels que le texte, les images et l'audio. Dans notre projet de pipeline d'analyse des commentaires YouTube, les modèles Transformers sont utilisés pour la summarisation et l'analyse des commentaires. Le Dockerfile inclut l'installation de Transformers pour permettre l'utilisation de ces modèles pré-entraînés. Cela facilite l'extraction d'informations pertinentes, la summarisation de commentaires et la génération de résumés audio.

FIGURE 1.3 – Logo de transformers

1.4.1.3 Pandas

Pandas est une bibliothèque open-source en Python utilisée pour la manipulation et l'analyse de données. Elle offre des structures de données flexibles et des outils performants pour travailler avec des données étiquetées ou relationnelles. Dans notre projet de pipeline d'analyse des commentaires YouTube, Pandas est utilisé pour la manipulation et la préparation des données collectées.

FIGURE 1.4 – Logo de pandas

1.4.1.4 Scipy

SciPy est une bibliothèque open-source en Python utilisée pour les calculs scientifiques et techniques. Elle s'appuie sur NumPy et fournit un large éventail de fonctions et de modules pour l'optimisation, l'intégration, l'interpolation, l'algèbre linéaire, les statistiques et d'autres tâches liées au calcul scientifique. En intégrant SciPy dans notre pipeline via Docker, nous nous assurons que tous les outils nécessaires pour le calcul scientifique sont disponibles, ce qui permet de réaliser des analyses de données avancées et de transformer les résultats de manière significative.

FIGURE 1.5 – Logo descipy

1.4.1.5 FPDF

FPDF est une bibliothèque open-source en Python conçue pour la génération de fichiers PDF. Grâce à sa flexibilité, FPDF supporte diverses fonctionnalités telles que l'ajout de textes, d'images, de tableaux et de graphiques, offrant ainsi un contrôle complet sur la mise en page et le format des documents. En intégrant FPDF dans notre pipeline, nous nous assurons que nous avons tous les outils nécessaires pour créer des tableaux de bord visuellement engageants.

Cela permet de rendre les résultats de notre analyse des commentaires YouTube facilement accessibles et compréhensibles pour les utilisateurs finaux.

FIGURE 1.6 – Logo de FPDF

1.4.2 Docker Compose

Docker Compose est un outil qui nous permet de spécifier et de gérer plusieurs conteneurs Docker en même temps. Le fichier Docker Compose configure un environnement Airflow pour le développement local, utilisant l'image personnalisée créée à partir du Dockerfile décrit précédemment. Il inclut des services pour PostgreSQL et Redis, essentiels pour le stockage des données et la gestion des files d'attente de tâches respectivement. Les services Airflow configurés comprennent le serveur web, le planificateur, le déclencheur et un service initial pour les configurations de base. Des volumes locaux sont montés pour persister les DAGs, les logs, les configurations et les plugins.

1.5 Conclusion

La dockerisation de notre pipeline offre de nombreux avantages, notamment la portabilité, la reproductibilité et la facilité de gestion. En encapsulant chaque composant dans un conteneur Docker, nous assurons une isolation et une cohérence de l'environnement de développement.

Après avoir créé et configuré l'environnement Docker nécessaire pour construire notre projet, nous allons maintenant définir le workflow et l'architecture adoptés pour construire le pipeline d'analyse des sentiments.

CHAPITRE 2

INTRODUCTION À APACHE AIRFLOW

2.1 Introduction

Ce chapitre offre une introduction approfondie à Apache Airflow, une plateforme puissante pour l'orchestration des workflows. Airflow permet de définir, de planifier et de surveiller des workflows complexes sous forme de graphes acycliques dirigés (DAG). Nous explorerons ses concepts fondamentaux ainsi que l'architecture adoptée pour la construction de la pipeline, ainsi que la gestion et l'orchestration des tâches nécessaires au sein du DAG de notre projet.

2.2 Présentation d'Apache Airflow

2.2.1 Definition de Apache Aiflow

Apache Airflow est une plate-forme open source permettant de créer, planifier et surveiller des flux de travail . Il a été créé chez Airbnb en 2015 (comme un meilleur moyen de créer, d'itérer et de surveiller rapidement les pipelines de données par lots gérant l'énorme quantité de données traitées par Airbnb)

FIGURE 2.1 – Logo de Apache Airflow

2.2.2 Historique d'Apache Airflow

Apache Airflow a une histoire fascinante, reflétant l'évolution des besoins en gestion de workflows dans l'industrie du logiciel. L'outil a été initialement conçu par *Maxime Beauchemin* chez Airbnb en **2014**. À cette époque, Airbnb rencontrait des défis croissants en matière de gestion de données et avait besoin d'un système plus robuste que les solutions existantes pour orchestrer ses workflows complexes.

En **2015**, Airflow a été open-sourcé, permettant à des développeurs du monde entier de contribuer et d'améliorer l'outil. Cette décision a été un tournant, ouvrant la voie à des innovations et des extensions continues.

En **2016**, Airflow a rejoint l'Apache Software Foundation (ASF), en tant que projet incubateur, marquant un autre jalon important dans son histoire. L'association avec l'ASF a renforcé la crédibilité d'Airflow et a élargi sa communauté d'utilisateurs et de contributeurs. En **2019**, il est devenu un projet de premier niveau sous l'égide d'Apache, confirmant sa maturité, sa stabilité et sa popularité au sein de la communauté open-source.

Le parcours d'Apache Airflow est un exemple remarquable de la manière dont un projet open-source peut évoluer et s'adapter aux besoins changeants de l'industrie.

2.2.3 Avantages du airflow

1. Il peut gérer les données en amont/en aval avec élégance
2. Accès facile aux données historiques (remplissage et réexécution des données historiques)
3. Gestion facile des erreurs (réessayez en cas d'échec)
4. Il a une communauté fantastique (vous pouvez rejoindre la communauté airflow sur slack)
5. Il a des capacités de journalisation fantastiques
6. Évolutivité et gestion des dépendances
7. Surveillance puissante que vous pouvez exécuter et visualiser l'exécution du flux de travail en temps réel
8. Créer, planifier et surveiller par programmation des flux de travail ou des pipelines de données

2.2.4 Cas d'utilisation de Airflow

Le flux d'air Apache peut être utilisé à diverses fins

— Tâches ETL (extraire, transformer, charger)
— Extraire des données de plusieurs sources
— Pipelines d'apprentissage automatique
— Entreposage de données
— Orchestrer les tests automatisés

2.3 Concepts de base : DAG, Opérateurs, Tâches

Les principaux concepts d'Airflow incluent :

— **DAG (Directed Acyclic Graph) :** Une structure représentant un workflow où les nœuds sont des tâches et les arêtes définissent les dépendances.

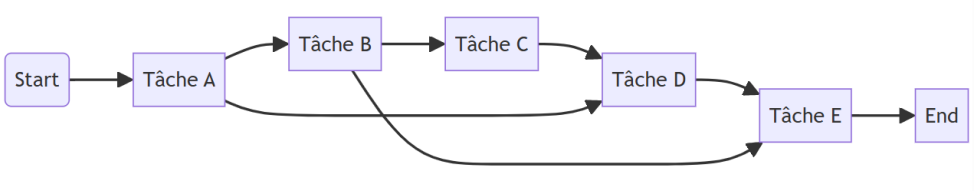


FIGURE 2.2 – Exemple de DAG

— **Opérateurs :** Les composants de base dans Airflow pour définir les tâches. Il existe différents types d'opérateurs, comme PythonOperator, BashOperator, et bien d'autres.

— **Tâches :** Les actions à exécuter dans un DAG.

2.4 Conception de l'architecture du pipeline

Pour notre projet de construction du pipeline d'analyse des sentiments des vidéos YouTube, nous avons adopté l'architecture suivante :

Nous avons conçu et utilisé un seul DAG pour gérer les tâches nécessaires aux différentes étapes du projet. Ce DAG comprend les tâches suivantes :

— **Scraping_comments** : Cette tâche utilise un opérateur Python (*PythonOperator*) pour extraire les commentaires des vidéos YouTube depuis des chaînes spécifiques.

— **Cleaning_comments** : Une fois les commentaires extraits, cette tâche nettoie les données en supprimant les éléments indésirables et en normalisant le texte.

— **Analysing_comments** : Cette tâche effectue l'analyse des sentiments sur les commentaires nettoyés, en utilisant des modèles de machine learning pour déterminer les sentiments exprimés.

— **Summarizing_comments** : Cette tâche résume les commentaires pour en extraire les principales informations.

— **to_audio** : Transforme les commentaires résumés en fichiers audio.

— **DashBoard** : La dernière tâche compile les résultats de l'analyse des sentiments et génère un tableau de bord interactif pour la visualisation des données.

2.4.1 Dépendances des Tâches

Les tâches dans ce DAG ont des dépendances spécifiques, comme illustré dans l'image précédente, et certaines tâches peuvent s'exécuter en parallèle pour optimiser l'efficacité du pipeline. Les dépendances sont définies de la manière suivante :

— **Scraping_comments** est la première tâche qui démarre le processus.
— **Cleaning_comments** dépend de **Scraping_comments** et nettoie les données extraites.
— Après **Cleaning_comments**, deux branches de traitement s'exécutent en parallèle :
  — **Analysing_comments** est exécuté pour analyser les sentiments des commentaires nettoyés.
  — **Summarizing_comments** est exécuté pour résumer les commentaires.
— **to_audio** dépend de **Summarizing_comments** et transforme les commentaires résumés en fichiers audio.
— **DashBoard** dépend à la fois de **Analysing_comments** et de **to_audio**, et compile les résultats pour créer un tableau de bord interactif.

La structure des dépendances des tâches est définie comme suit :

```
Scraping_comments >> Cleaning_comments >> [Analysing_comments, Summarizing_comments]
Summarizing_comments >> to_audio
[Analysing_comments, to_audio] >> DashBoard
```

Cette structure de dépendances permet d'optimiser le flux de travail en permettant à certaines tâches de s'exécuter en parallèle, réduisant ainsi le temps global nécessaire pour compléter le pipeline. En structurant le DAG de cette manière, nous assurons une progression fluide et efficace du traitement des données, depuis l'extraction initiale jusqu'à la visualisation finale.

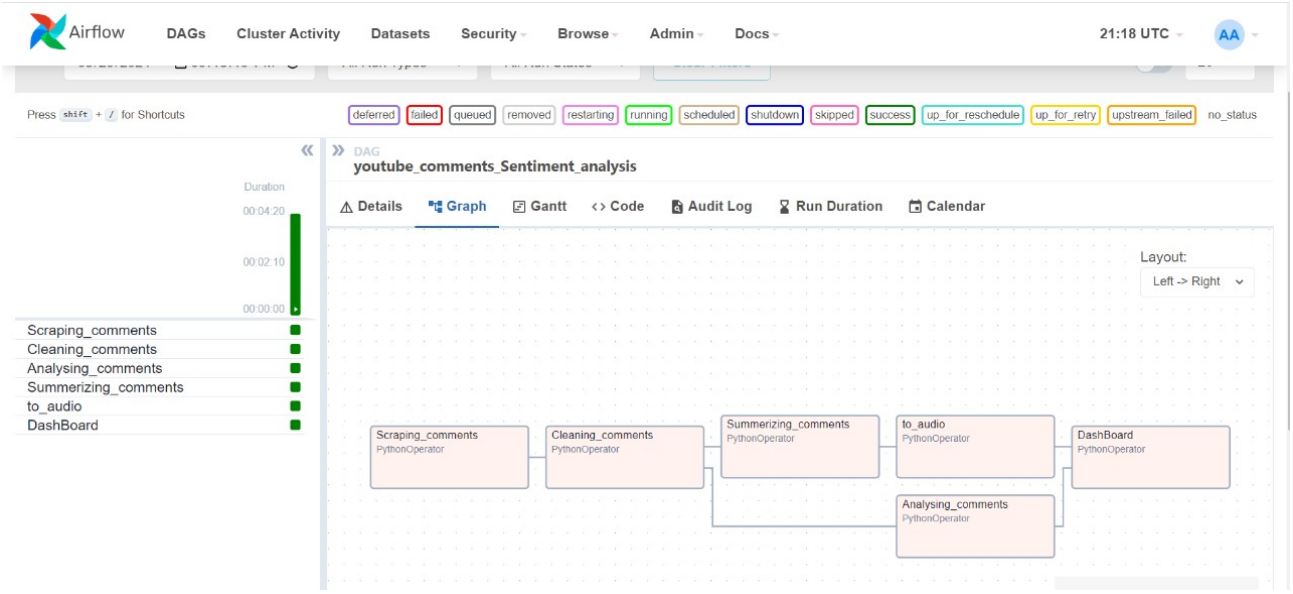


FIGURE 2.3 – Pipeline avec Airflow

2.5 Conclusion

En conclusion, ce chapitre est consacré à l'introduction de l'outil principal de ce projet, Apache Airflow, ainsi qu'à la conception du pipeline et des tâches contenues dans le DAG.

CHAPITRE 3

COLLECTE DES DONNÉES

3.1 Introduction

La collecte de données est une étape cruciale dans tout projet d'analyse. Dans ce chapitre, nous détaillons la tache de collecte des commentaires YouTube en utilisant l'API YouTube, de l'authentification via Google Cloud à l'extraction et au stockage des données en fichiers CSV.

3.2 Introduction à l'API YouTube

L'API YouTube Data permet d'accéder à divers aspects de la plateforme YouTube, y compris les vidéos, les playlists, les commentaires, et bien plus encore. Cette API offre des méthodes robustes pour interagir avec les données publiques de YouTube et est largement utilisée pour des projets d'analyse de contenu et de commentaires.

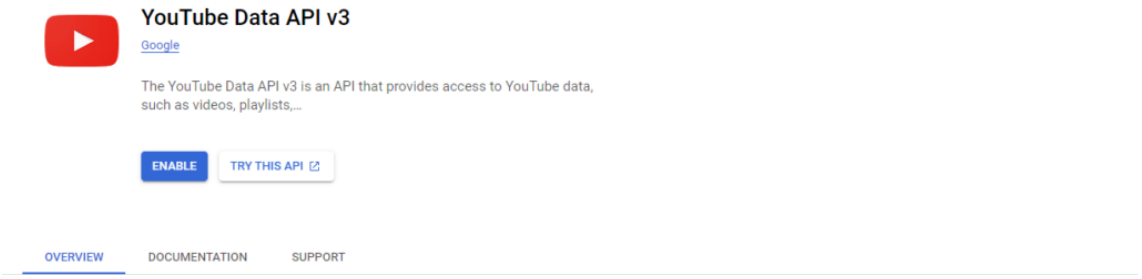


FIGURE 3.1 – l'API YouTube

3.3 Authentification et autorisation via Google Cloud

Pour accéder à l'API YouTube, il est nécessaire d'obtenir une clé API via Google Cloud. Voici les étapes pour configurer l'authentification et l'autorisation :

1. Créer un projet sur Google Cloud Console.
2. Activer l'API YouTube Data v3 pour le projet.
3. Générer une clé API et restreindre son usage pour des raisons de sécurité.

3.4 Méthodologie de Web Scraping

La méthodologie de web scraping dans ce DAG est implémentée dans la tâche `Scraping_comments`, utilisant l'API YouTube Data pour extraire les commentaires des vidéos. Voici les étapes clés du processus de scraping :

1. **Initialisation de l'API YouTube** : La connexion à l'API YouTube est établie en utilisant la clé API fournie.
2. **Récupération des Identifiants des Vidéos** : Pour chaque chaîne YouTube listée, les vidéos sont récupérées à partir de la playlist de téléchargements de la chaîne.
3. **Extraction des Commentaires** : Pour chaque vidéo, tous les commentaires sont extraits en utilisant des requêtes paginées pour gérer les commentaires nombreux.
4. **Filtrage et Validation des Commentaires** : Les commentaires sont filtrés pour enlever ceux qui sont trop longs, contiennent des liens ou des balises HTML, ou ont une proportion élevée de caractères non alphabétiques.
5. **Sauvegarde des Commentaires** : Les commentaires valides sont sauvegardés dans un fichier CSV, incluant le nom de la chaîne, le titre de la vidéo, le texte du commentaire et la date de publication.

3.5 Stockage des données en fichiers CSV

Les commentaires extraits sont ensuite stockés dans des fichiers CSV pour une utilisation ultérieure. Chaque fichier CSV contient les informations suivantes : nom de la chaîne, titre de la vidéo, commentaire, et date de publication

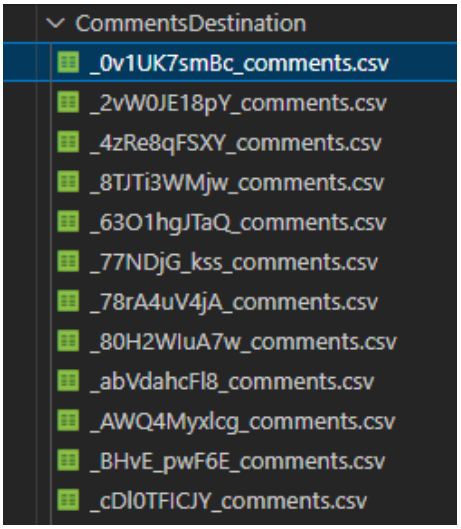


FIGURE 3.2 – Les données en fichiers CSV

3.6 Conclusion

Ce chapitre a détaillé les étapes nécessaires pour collecter des commentaires YouTube en utilisant l'API YouTube. Nous avons abordé l'authentification, la méthodologie de scraping, l'extraction des commentaires et leur stockage en fichiers CSV. Ces données serviront de base pour les analyses ultérieures dans les chapitres suivants.

CHAPITRE 4

CLEANING ET PRÉPARATION DES DONNÉES

4.1 Introduction

Le nettoyage et la préparation des données sont des étapes cruciales dans tout projet d'analyse de données. Ces étapes garantissent la qualité et la fiabilité des données avant leur analyse. Dans ce chapitre, nous aborderons les techniques de nettoyage des fichiers CSV contenant les commentaires YouTube, y compris la suppression des caractères indésirables, la gestion des valeurs manquantes, et la fusion des fichiers CSV en un seul fichier.

4.2 Introduction au nettoyage des données

Le nettoyage des données vise à améliorer la qualité des données brutes en supprimant ou en corrigeant les erreurs et les incohérences. Cette étape est essentielle pour garantir des analyses précises et des résultats fiables. Dans le cadre de notre projet, nous nous concentrons sur le nettoyage des commentaires YouTube extraits en supprimant les emojis, la ponctuation et les valeurs manquantes.

4.3 Techniques de nettoyage des fichiers CSV

4.3.1 Suppression des caractères indésirables

Les commentaires peuvent contenir des caractères indésirables tels que des emojis et de la ponctuation. Ces caractères sont supprimés pour normaliser les données.

Commentaire avant cleaning	Commentaire après cleaning
Really Incredible project, just amazing :)	Really Incredible project just amazing
Bounce ho raha hai sir :(	Bounce ho raha hai sir
i love O_O end to end ML series	i love end to end ML series

4.3.2 Fusion des fichiers CSV

4.3.2.1 Présentation de Apache NIFI

FIGURE 4.1 – Logo de Apache NIFI

Apache NiFi est une plateforme open source dédiée à la gestion des flux de données en temps réel. Elle offre une interface graphique intuitive pour concevoir, automatiser et surveiller les transferts de données entre divers systèmes. NiFi est reconnu pour sa fiabilité, sa scalabilité et sa capacité à traiter de gros volumes de données de manière efficace.

Pour prétraiter et fusionner les fichiers CSV, l'utilisation d'Apache NiFi est nécessaire pour répondre aux exigences du projet. Cependant, lors de nos tentatives pour intégrer NiFi avec Airflow, nous avons rencontré des problèmes de performances. Cette intégration a entraîné une lenteur du fonctionnement de l'ordinateur voire même des plantages. Après plusieurs essais pour résoudre ces problèmes sans succès, nous avons décidé d'opter pour une approche basée sur Python. Cette décision a été motivée par sa simplicité d'utilisation et sa capacité à être personnalisée.

Toutefois, afin de répondre aux exigences du projet et dans un souci d'exploration et d'acquisition de compétences, nous avons décidé de travailler avec Apache NiFi de manière indépendante par rapport à Airflow. L'objectif était de maîtriser les fonctionnalités offertes par NiFi et d'enrichir notre ensemble de compétences. En déployant cette approche, nous avons pu nous concentrer pleinement sur l'apprentissage et l'exploration des capacités de NiFi sans les contraintes liées à son intégration avec Airflow.

4.3.2.2 fusion des fichier csv avec python

Les valeurs manquantes peuvent poser des problèmes pour l'utilisation des modèles Hugging Face. Dans notre cas, les lignes où les commentaires sont manquants ou illisibles sont éliminées.

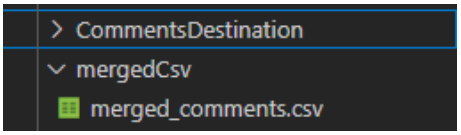


FIGURE 4.2 – Les données fusionnées CSV

Ainsi, nous avons introduit une tâche de prétraitement et de fusion des données extraites lors du scraping dans notre DAG d'Airflow. Ce script Python permet de nettoyer les commentaires et de les fusionner à partir de plusieurs fichiers CSV. En dépit des contraintes rencontrées avec NiFi, cette solution basée sur Python nous offre la flexibilité nécessaire pour atteindre nos objectifs de manière efficace.

4.4 Conclusion

Le nettoyage et la préparation des données sont des étapes fondamentales pour garantir des analyses précises et significatives. Grâce à des techniques de suppression des caractères indésirables, de gestion des valeurs manquantes, et de fusion des fichiers CSV, nous avons transformé des commentaires bruts en données prêtes à l'emploi. Ces données nettoyées serviront de base pour les analyses détaillées et les visualisations dans les chapitres suivants.

CHAPITRE 5

INTÉGRATION DES MODÉLES DE HUGGING FACE

5.1 Introduction

Après avoir extrait, traité et fusionné les commentaires dans un fichier CSV, cette partie se concentrera sur l'utilisation des modèles de Hugging Face. Ces modèles seront exploités pour deux tâches : l'analyse des sentiments exprimés dans les commentaires et la génération de résumés audio de ces commentaires.

5.2 Présentation de Hugging Face

Hugging Face est une plateforme d'IA axée sur la mise à disposition de modèles de pointe en traitement du langage naturel (NLP) et dans d'autres domaines de l'intelligence artificielle. Fondée en 2016, Hugging Face est devenue une référence incontournable pour les chercheurs, les développeurs et les entreprises qui souhaitent accéder à des modèles NLP pré-entraînés, ainsi qu'à une variété d'outils et de services associés.

5.2.1 Points forts de Hugging Face

1. **Modèles de pointe :** Hugging Face offre une vaste bibliothèque de modèles NLP pré-entraînés pour une multitude de tâches, allant de la classification de texte à la génération de langage naturel en passant par la traduction et la résumé automatique.
2. **Communauté active :** La plateforme Hugging Face a une communauté dynamique de chercheurs, de développeurs et d'enthousiastes de l'IA qui contribuent régulièrement en partageant des modèles, des astuces et des idées.
3. **Hub de modèles :** Hugging Face propose un Hub de modèles où les utilisateurs peuvent trouver, partager et télécharger des modèles NLP pré-entraînés, ainsi que des artefacts associés tels que des tokenizer et des embeddings.

4. **Facilité d'utilisation :** Les API Hugging Face sont conçues pour être simples et conviviales, permettant aux utilisateurs d'accéder facilement aux fonctionnalités avancées des modèles NLP sans avoir à se soucier des détails de leur mise en œuvre.
5. **Open source :** La plupart des projets Hugging Face sont open source, ce qui favorise la transparence, la collaboration et l'innovation dans le domaine de l'IA.

FIGURE 5.1 – Logo de hugging face

5.3 Modèle de résumé des commentaires

Application d'un modèle de résumé Hugging Face aux commentaires fusionnés :

Dans cette partie, nous appliquerons un modèle de résumé de texte provenant de Hugging Face aux commentaires fusionnés que nous avons précédemment traités. Le modèle utilisé est chargé à partir du chemin spécifié et est ensuite utilisé pour générer des résumés pour chaque groupe de commentaires, regroupés par nom de chaîne.

Ce code charge les commentaires fusionnés à partir d'un fichier CSV, les regroupe par nom de chaîne, puis utilise un modèle de résumé Hugging Face pour générer des résumés pour chaque groupe de commentaires. Les résumés sont ensuite enregistrés dans un nouveau fichier CSV pour une utilisation ultérieure :

1. **Chargement des données** : Les commentaires fusionnés sont chargés à partir d'un fichier CSV spécifié.
2. **Initialisation du modèle de résumé** : Un modèle de résumé de texte provenant de Hugging Face est initialisé à l'aide du chemin spécifié.
3. **Préparation de la structure de sortie** : Un DataFrame vide est créé pour stocker les résumés générés, avec des colonnes pour le nom de la chaîne et le résumé associé.
4. **Résumé des commentaires** : Les commentaires sont regroupés par nom de chaîne, et pour chaque groupe de commentaires, tous les commentaires sont combinés en une seule chaîne de texte. En utilisant le modèle de résumé, un résumé est généré pour chaque chaîne de texte combinée.
5. **Enregistrement des résumés** : Les résumés générés sont stockés dans le DataFrame, puis sauvegardés dans un nouveau fichier CSV.

5.4 Modèle de Synthèse Vocale

Il s'agit d'un modèle de synthèse vocale pré-entraîné pour convertir du texte en fichiers audio. Le modèle, issu de la bibliothèque Transformers de Hugging Face, est chargé et utilisé pour générer des données audio à partir du texte donné en entrée. Cette capacité de conversion texte-parole ouvre des perspectives dans divers domaines, notamment l'accessibilité, la création de contenu multimédia et la génération de podcasts.

Voici les principales étapes que le modéle traverse pour convertir les résumés textuels en fichiers audio

1. **Chargement du modèle et du tokenizer** : Il commence par charger un modèle de synthèse vocale (TTS) pré-entraîné pour l'anglais ainsi que le tokenizer associé à partir des répertoires spécifiés.
2. **Extraction des résumés depuis un fichier CSV** : Ensuite, il extrait les résumés à convertir en audio à partir d'un fichier CSV donné.
3. **Préparation du répertoire de sortie** : Le script crée un répertoire de sortie pour stocker les fichiers audio générés, s'il n'existe pas déjà.
4. **Conversion des résumés en audio** : Pour chaque résumé extrait, il le tokenise à l'aide du tokenizer. Il utilise ensuite le modèle de synthèse vocale pour générer un fichier audio à partir du texte tokenisé.
5. **Sauvegarde des fichiers audio** : Les fichiers audio générés sont sauvegardés dans le répertoire de sortie, chaque fichier étant nommé d'après le nom de la chaîne suivi de "summary.wav".
6. **Affichage des chemins de sauvegarde** : Finalement, le code affiche les chemins complets de sauvegarde de chaque fichier audio généré.

5.5 Modéle d'analyse de sentiments

Le modèle utilisé est un modèle de classification de texte pré-entraîné, spécifiquement conçu pour l'analyse de sentiment des commentaires. Il tire parti de techniques avancées en traitement du langage naturel pour attribuer des étiquettes de sentiment à chaque commentaire, permettant ainsi une analyse approfondie des opinions exprimées.

1. **Importation des bibliothèques** : importation des bibliothèques pandas pour la manipulation des données et la bibliothèque transformers pour utiliser les pipelines de traitement du langage naturel.
2. **Fonction d'analyse de sentiment** : La fonction `analyze_sentiment` prend en entrée une liste de commentaires et un pipeline de classification de texte pour l'analyse de sentiment. Elle vérifie d'abord si les commentaires sont fournis en tant que chaîne unique ou sous forme de liste. Ensuite, elle applique le pipeline d'analyse de sentiment aux commentaires et retourne une liste d'étiquettes de sentiment correspondant à chaque commentaire.
3. **Fonction principale d'analyse** : La fonction `Analyse` :

- Charge le pipeline d'analyse de sentiment à partir d'un modèle pré-entraîné spécifié.
- Charge un fichier CSV contenant les commentaires fusionnés.
- Applique l'analyse de sentiment à la colonne 'Comment' du DataFrame en utilisant la fonction `analyze_sentiment`.
- Ajoute les étiquettes de sentiment obtenues comme une nouvelle colonne 'Sentiment' dans le DataFrame.
- Enregistre le DataFrame résultant dans un nouveau fichier CSV.

4. **Appel de la fonction principale** : appelle la fonction `Analyse` pour exécuter l'analyse de sentiment sur les commentaires et enregistrer les résultats.

5.6 Conclusion

Ce chapitre nous permet d'appliquer des modèles dérivés de Hugging Face, afin d'analyser les sentiments des commentaires, de générer des résumés et de les transformer en audio.

CHAPITRE 6

DASHBOARDING

6.1 Introduction

Après avoir effectué toutes les tâches de notre DAG, la dernière étape consiste en la visualisation des résultats.

6.2 Objectifs du dashboard

Le dashboarding est une méthode de visualisation des données qui permet de représenter des informations clés de manière concise et visuelle. Les tableaux de bord sont utilisés pour surveiller et analyser des données complexes en temps réel, facilitant ainsi la prise de décision rapide et informée.

L'objectif principal de notre tableau de bord est de fournir une vue d'ensemble des sentiments exprimés dans les commentaires de différentes chaînes, ainsi que de résumer les informations pertinentes pour chaque chaîne. Ce tableau de bord est destiné aux gestionnaires de contenu et aux analystes de données qui souhaitent comprendre et agir sur les commentaires reçus.

6.3 Présentation du Tableau de Bord Final

Le tableau de bord final est un fichier PDF structuré en deux parties principales :

6.3.1 Graphique des Sentiments

Cette section présente la visualisation des commentaires positifs, négatifs et neutres par chaîne. Le graphique des sentiments permet de voir rapidement la répartition des différents types de commentaires pour chaque chaîne. Cette visualisation aide à identifier les chaînes qui reçoivent principalement des commentaires positifs, négatifs ou neutres.

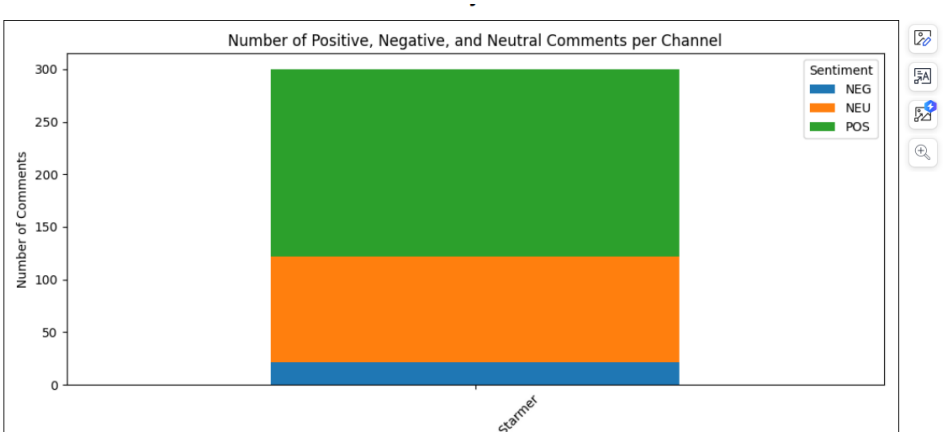


FIGURE 6.1 – Analyse des commentaires de la chaine Statquedt with josh starmer

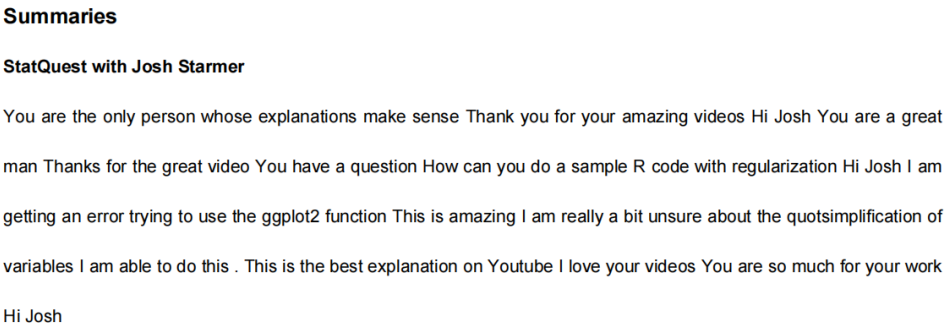
Description du Graphique : Le graphique représente les résultats de l'analyse des sentiments pour les commentaires associés à la chaîne "StatQuest with Josh Starmer". Il se compose de trois types de commentaires empilés les uns sur les autres, différenciés par des couleurs distinctes :

- **Bleu** : Représente le nombre de commentaires négatifs (NEG).
- **Orange** : Représente le nombre de commentaires neutres (NEU).
- **Vert** : Représente le nombre de commentaires positifs (POS).

Interprétation Le graphique montre que la majorité des commentaires pour la chaîne "StatQuest with Josh Starmer" sont positifs (vert), suivis par un nombre significatif de commentaires neutres (orange) et un très petit nombre de commentaires négatifs (bleu). Cela indique une tendance globalement favorable dans les commentaires pour cette chaîne.

6.3.2 Résumés des Chaînes

Dans cette section, chaque chaîne est accompagnée d'une description détaillée et de points saillants. Les résumés des chaînes fournissent des informations complémentaires qui permettent de comprendre le contexte et les points clés de chaque chaîne. Cela inclut les thèmes récurrents, les retours des utilisateurs et les aspects importants à considérer pour chaque chaîne.



Summaries

StatQuest with Josh Starmer

You are the only person whose explanations make sense Thank you for your amazing videos Hi Josh You are a great man Thanks for the great video You have a question How can you do a sample R code with regularization Hi Josh I am getting an error trying to use the ggplot2 function This is amazing I am really a bit unsure about the quotsimplification of variables I am able to do this . This is the best explanation on Youtube I love your videos You are so much for your work Hi Josh

FIGURE 6.2 – Summaries de la chaine StatQuest with Josh Starmer

6.4 Conclusion

Après avoir conçu un pipeline pour réussir notre projet, qui consiste principalement en l'analyse des commentaires des chaînes YouTube, et après avoir accompli les tâches précédentes, nous avons finalement visualisé les données sous forme de tableau de bord. Ce tableau de bord montre les commentaires positifs, négatifs et neutres pour chaque chaîne.

CONCLUSION GÉNÉRALE

En conclusion, la mise en place d'un pipeline complet pour l'analyse des commentaires YouTube représente une étape cruciale dans l'extraction de connaissances et d'informations pertinentes à partir de vastes ensembles de données provenant de sources variées. Ce projet a permis de démontrer l'efficacité et l'importance d'une approche systématique et méthodique, allant de la collecte initiale des données à leur nettoyage, leur analyse et leur présentation visuelle sous forme d'un tableau de bord interactif.

Grâce à ce pipeline, les utilisateurs peuvent non seulement accéder aux commentaires YouTube, mais aussi les comprendre plus en profondeur en identifiant les tendances, les sentiments et les sujets discutés. L'utilisation de technologies telles que l'API YouTube, le web scraping, Dash et d'autres outils Python a permis de créer une solution robuste et flexible, adaptable à divers besoins et cas d'utilisation.

En outre, ce projet met en évidence l'importance croissante de l'analyse de données dans le domaine du contenu en ligne, où les commentaires des utilisateurs jouent un rôle significatif dans la compréhension des préférences, des opinions et des comportements des audiences. En fournissant des informations exploitables à partir de ces commentaires, ce pipeline contribue à améliorer la prise de décision et la compréhension des tendances émergentes dans l'écosystème YouTube et au-delà.

BIBLIOGRAPHIE

1. Cours du professeur Nourddine KERZAZI
2. https://airflow.apache.org/docs/ (Consulté le 23 mai 2024)
3. https://medium.com/analytics-vidhya/apache-airflow-what-it-is-and-why-you-should-start-using-it-c6334090265d (Consulté le 23 mai 2024)
4. https://airflow.apache.org/use-cases/ (Consulté le 24 mai 2024)
5. https://blog.stephane-robert.info/docs/services/scheduling/apache-airflow/ (Consulté le 24 mai 2024)
6. https://developers.google.com/youtube/v3/getting-started (Consulté le 24 mai 2024)
7. https://docs.docker.com/desktop/ (Consulté le 24 mai 2024)
8. https://datascientest.com/docker-guide-complet (Consulté le 24 mai 2024)
9. https://pytorch.org/docs/stable/index.html (Consulté le 24 mai 2024)
10. https://pypi.org/project/transformers/ (Consulté le 24 mai 2024)
11. https://huggingface.co/facebook/mms-tts-eng
12. https://huggingface.co/finiteautomata/bertweet-base-sentiment-analysis
13. https://huggingface.co/Falconsai/text_summarization