Titre RNCP 39586

INGENIEUR EN SCIENCE DES DONNÉES SPECIALISÉ EN APPRENTISSAGE AUTOMATIQUE

Bloc 2 :
ANALYSE, ORGANISER ET VALORISER DES DONNÉES

Sujet :
Développement d'un outil d'analyse automatisé des commentaires YouTube

Auteur :
FURTADO LEAL Carla

Juillet 2025

Sommaire

I. Analyse des données

A. Analyse du besoin

Le travail d'un influenceur consiste à créer du contenu sur un ou plusieurs plateformes de réseaux sociaux. Son succès repose sur sa capacité à développer et fidéliser son audience. En plus de cet aspect création et production, il doit également gérer l'aspect relationnel avec ses auditeurs, l'aspect commercial et les négociations avec les marques partenaires pour de placements de produits. Ainsi, ils génèrent des revenus grâce à la monétisation de leur contenu sur les plateformes, les partenariats et la vente des leurs propres produits. Tout cela repose sur leur capacité à susciter l'intérêt et de l'engament des auditeurs et à comprendre leurs attentes et préférences. Or, plus il y a d'auditeur plus il peut être complexe de suivre et d'analyser manuellement les retours.

Le Youtubeur souhaite enrichir sa connaissance de sa base d'abonnés en utilisant les commentaires laissés sous ses vidéos. Il aimerait savoir si les réactions sont positives ou non et savoir globalement ce qui est dit, les sujets abordés. Jusqu'à maintenant il ne se basait que sur des indicateurs quantitatifs pour évaluer l'engagement de ses abonnés (nombre de likes, de vues, de commentaires…), le nombre de likes étant ce qui se rapproche le plus d'un indicateur qualitatif. En claire, le besoin est de pouvoir suivre ses performances, l'engagement de ses abonnées, pour mesurer l'impact de son contenu. Ce qui pose la problématique suivante :

Comment valoriser les retours client via l'analyse automatisé des commentaires You Tube ?

Il existe déjà plusieurs outils comme You Tube studio ou brand 24 qui apporte une réponse à cette question.
Ces outils leur permettent de d'apprécier l'évolution du nombre de vus, likes, commentaires… de mieux connaitre leur base d'abonnés (le profil, le genre de contenu qu'ils aiment…). Elles intègrent également des analyses plus poussées comme l'analyse de sentiment et de sujets. Si nous voulons également répondre à cette problématique, nous devons créer une infrastructure sécuriser qui permet de collecter, transformer et analyse les données de manière automatique. Ces outils, ces analyses devrons produire des insights claires et pertinents pour le client qui lui permettront de mieux apprécier son impact et de mieux comprendre son audience. Les enjeux sont pour lui stratégique et économiques.

B. Présentation d'un plan d'analyse

En claire, quelle valeur/informations peut-on tirer des commentaires YouTube, utile pour l'orientation stratégique du client. Soit, en d'autres termes : comment optimiser et appliquer l'analyse de sentiment et le topic modeling de manière automatique en garantissant un certain niveau de performance. Nous classons l'exploitation de ces données en 2 niveaux d'analyse :

- **Niveau 1 : analyse de surface**

Pour ce premier niveau de l'analyse nous exploitons le nombre de commentaires, la date de publication, le nombre de likes associés et la longueur des commentaires. Ces métriques nous permettent d'analyser l'engagement des abonnées dans le temps avec l'évolution du nombre de commentaires et de connaitre le contenu du commentaire avec le plus de like. La longueur du commentaire permet d'afficher le commentaire le plus long. Cela permet de donner un aperçu de l'engagement et les prémisses des résultats du deuxième niveau. Cette analyse de premier niveau se présente sous la forme d'un dashboard. Cette analyse utilise principalement les méta données associées et calculées à partir des commentaires.

- **Niveau 2 : analyse de fond**

L'analyse de fond exploite directement les commentaires pour en qualifier le contenu et son impact de 2 façons :

  - La polarité du sentiment exprimé

L'analyse de sentiment permet de classer les commentaires en 2 ou 3 catégories, positif, négatif et neutre. Ainsi, cet axe d'analyse permet qualifier l'impact du contenu sur les spectateurs. Les résultats seront également présentés sous forme de tableau de bord. Associées au méta données ils permettrons répondre aux questions :

- Il y a -t-il de commentaires positifs, négatifs ou neutre
- A quel catégorie les commentaires les plus liké appartiennent-ils ?

  - Les principaux sujets abordés

Le topic modeling est un type d'apprentissage non supervisé qui se base un ensemble de documents pour en déterminer les principaux sujets. Ainsi les résultats dépendent de l'ensemble. Cette analyse permettra de savoir ce qui est globalement dit dans les commentaires, les principaux sujets abordés. Cet axe d'analyse permet d'identifier les leviers de cet impact. Associées au méta données il sera possible de dire :

- A quel(s) sujet(s) se rapportent les commentaires avec le plus de likes
- Les sujets abordés dans les commentaires classés négatives, positifs, neutres et dans l'ensemble
- La part de commentaires positifs, négatifs ou neutre dans chaque sujet

Étant donnée la nature du modèle, il n'y a pas d'entrainement préalable. De plus, les résultats ne sont pas stockés dans la base de données.

C. Présentation des requêtes et des résultats sous forme de dashboard

Notre outil d'analyse est alimenté par des données stockées dans des bases de données MongoDB compartimentées par chaine YouTube. Ainsi, les analyses menées sont indépendantes les unes des autres mais les méthodes sont identiques.

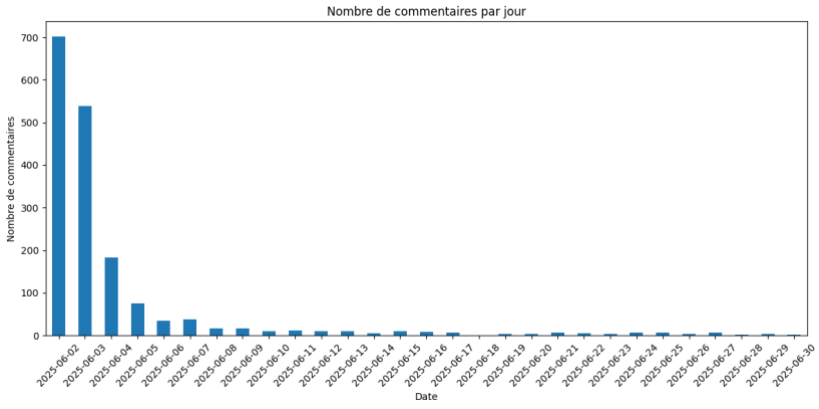
Phase d'exploration

Une fois l'étape d'extraction opérationnelle, nous avons mené une analyse exploratoire de nos données. Seule une partie des requêtes effectuées lors de cette analyse sera présentée au client en tant qu'indicateurs. Cette partie effectuée dans un Jupyter notbook, avant les étapes de transformation et de chargement dans la base. Dans un premier temps, les données sont stockées dans un objet data frame. C'est un format facilement manipulable avec pandas, qui nous permet de faire une analyse de premier niveau, « de faire connaissance » avec nos données et de répondre à quelques questions. L'instruction « shape » nous permet ici de compter le nombre de commentaires (équivalent au nombre de lignes en indice 0 de l'objet). Nous avons également cherché à savoir le nombre de personne ayant publié un commentaire et le nombre moyen de commentaire par personne.

```
# le nombre de commentaires
print(f'Nombre de commentaires : {df.shape[0]}')
df['author'].nunique()
# le nombre d'utilisateurs uniques
print(f'Nombre d\'utilisateurs uniques : {df["author"].nunique()}')
# le nombre de commentaires par utilisateur
print(f'Nombre moyen de commentaires par utilisateur : {df.groupby("author").size().mean():.2f}')
```

Une requête qui sera utilisé dans le Dashboard finale, car elle apport de l'information sur le comportement des abonnées, est celle qui produit le graphique de distribution du nombre de commentaire en fonction de leur date de publication.

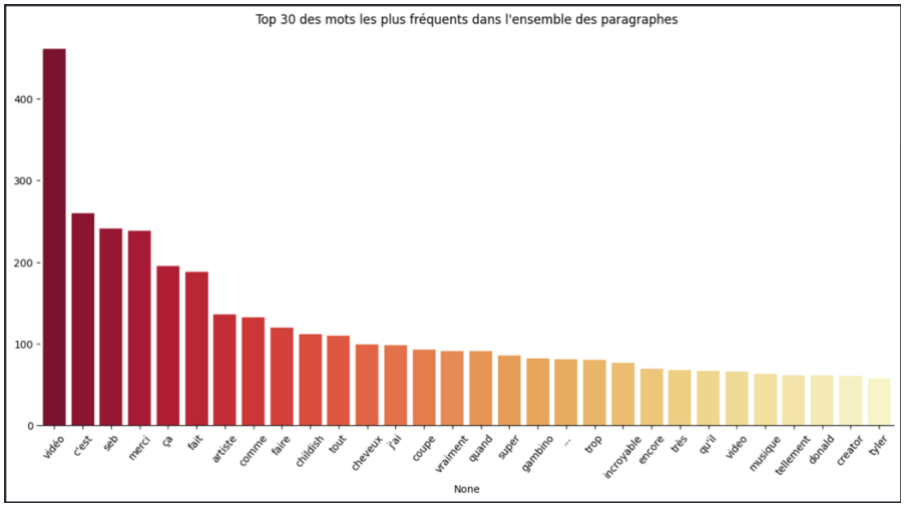
```
# le nombre de commentaires par jour
df['date'] = df['publishedAt'].dt.date
df['date'].value_counts().sort_index().plot(kind='bar', figsize=(12, 6))
plt.title('Nombre de commentaires par jour')
plt.xlabel('Date')
plt.ylabel('Nombre de commentaires')
plt.xticks(rotation=45)
plt.tight_layout()
```



Cette forme est commune à toutes les extractions et montre que la majorité des commentaires sont postés jusqu'à 2 jours après la publication du contenu.

Ensuite, nous nous sommes un peu plus intéressés au contenu des commentaires : nombre de noms, de mots uniques, les association fréquentes, les mots les plus fréquents… Ici nous avons utilisé NLTK pour certaines représentations.
La première commande permet de calculer la fréquence de chaque mot et de la stocker dans un dictionnaire grâce à « FreqDist ». On affiche les 30 premiers et donc les plus fréquents dans un graphique de type barplot.

```
fdist2 = FreqDist(text).most_common(30)
fdist2 = pd.Series(dict(fdist2))
fig, ax = plt.subplots(figsize=(15,7))
sns.barplot(x=fdist2.index, y=fdist2.values, ax=ax, palette = "YlOrRd_r")
#CMRmap
plt.title("Top 30 des mots les plus fréquents dans l'ensemble des paragraphes")
sns.despine(left=True, bottom=False)
plt.xticks(rotation=50)
```



L'information apporté dépend de la qualité du nettoyage, si les stopword n'ont pas été retiré du texte, ils apparaitront alors comme les mots les plus fréquents. Par exemple ici on peut des mots qui n'ont pas encore été traités comme « c'est », « ça », « j'ai », « qu'il » et la ponctuation « … ». Néanmoins, on apprend qu'on parle ici de musique, de certains artistes et apriori en positif. Nous avons également représenté la fréquence des bigrams dans un autre graphique similaire grâce à cette commande :

```
# afficher les bigrams les plus fréquents
from nltk import bigrams
bigrams_list = list(bigrams(text))
bigrams_freq = FreqDist(bigrams_list)
plt.figure(figsize=(15, 8))
bigrams_freq.plot(100, cumulative=False, title='Most Common Bigrams in Comments')
plt.show()
```

Ce graphique est moins lisible mais nous permet d'apprendre qu'il est également question de coupe de cheveux.
Une autre requête qui permet l'exploration du texte est l'affichage du contexte des mots sur un certain fenêtre centré sur un mot particulier, ici « video » :

```
from nltk import Text
text_nltk = Text(text)
text_nltk.concordance('video', width=100)
```

```
Displaying 3 of 3 matches:
art genre video_sur personne polyvalent penser video magnifique reve chacune artiste childish_gambi
iderman mile moral vraiment artiste admire yes video childish_gambino idole trop hater regarder bos
e pouce chial video_sur artiste preferer super video ! espece bruit bourdonnement agreable nouveau
```

Toutes ces requêtes son très utiles pour l'amélioration du nettoyage du texte.

Phase de développement du dashboard streamlit

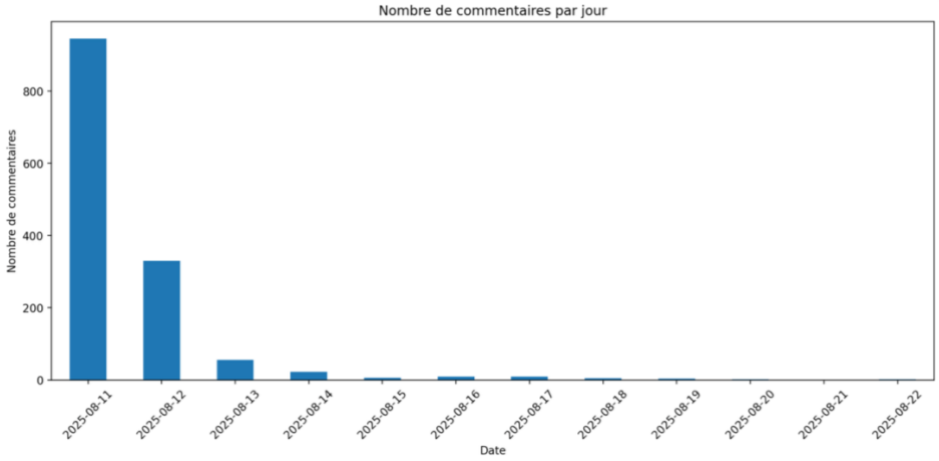
Ensuite, pour le dashboard nous avons principalement utilisé des requêtes pymongo pour interroger directement la base de données pour éviter de stocker toute une base dans un objet. Ce sont les requêtes Pymongo permettent d'accéder aux données stockées dans les collections.

En suivant le plan d'analyse, les indicateurs calculés pour l'analyse de surface sont :

- Le nombre de commentaires et commentaire avec le plus de like obtenu par ces requêtes :

```
db[videoid].count_documents({})
most_liked_comment = db[videoid].find_one(sort=[("likeCount", -1)])["comment"]
```

- Graphique de l'évolution du nombre de commentaires au cours du temps



- Nuage des mots

```
wordcloud = WordCloud(width=800, height=400, background_color='white', collocations=False).generate(text)
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.show()
    st.pyplot(plt)
```

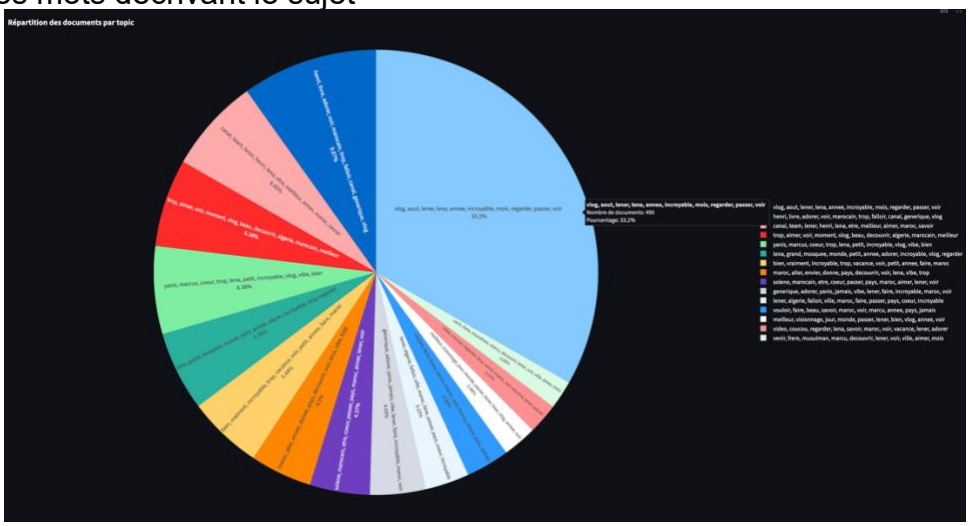
Pour l'analyse de fond :

- Graphique de répartition des commentaires en fonction du sentiment

```python
collection = client[db][videoid]
    # graphique camembert de répartition du nombre de commentaire positifs et négatifs et neutres
    sentiments = collection.aggregate([
        {"$group": {
            "_id": "$sentiment",
            "count": {"$sum": 1}
        }}
    ])
    sentiment_counts = {sentiment['_id']: sentiment['count'] for sentiment in sentiments}
    sentiment_labels = list(sentiment_counts.keys())
    sentiment_values = list(sentiment_counts.values())
    # Création du DataFrame pour Plotly
    df = pd.DataFrame({
        "Sentiment": sentiment_labels,
        "Valeur": sentiment_values
    })
    fig = px.pie(df, values="Valeur", names="Sentiment", title="Répartition des sentiments")
    st.plotly_chart(fig, use_container_width=True)
```

Ensuite le dashboard suivant permet de présenter :

- La répartition des commentaires dans les différents sujets
- Les mots décrivant le sujet



En fonction du nombre de sujet il peut être difficile à lire, comme ici, mais le type de graphique permet de se rendre compte visuellement de la répartition.

D. Méthodologie des tests statistiques

Pour valoriser les commentaires You Tube nous avons décidé de mener 2 analyses complémentaires : l'analyse de sentiment et de sujet (topic modeling).

Nos données ne sont pas étiquetées pour l'analyse de sentiments. Dans ce cas il existe 2 options :

- Utilisé un modèle pré entrainé
- Utiliser un modèle classique, l'entrainer sur des données similaires puis l'appliquer à nos données.

Les modèles pré-entrainés sont généralement très performants à condition d'avoir été entrainés sur des données similaires, notamment en termes de langue. Or très souvent les jeux de données disponibles sont en anglais, ce qui qui n'est pas idéal dans notre cas. De plus, l'utilisation de ce type de modèle nous fait perdre en explicabilité, à l'inverse des modèles de classifications classiques. Leur pouvoir explicatif est généralement très grand mais ils demandent une quantité de données important pour être performant.
Nous sommes donc face à 2 problèmes :

- Trouver un jeu de donnée assez proche de nos données
- Trouver le meilleur modèle

Pour les résoudre nous avons adopté la méthodologie suivante :

- Faire des tests de similitudes en des jeux de données tests et de données extraites de You Tube.
- Entrainer différents modèles sur le jeu de données sélectionné et sélectionner le plus performant.

Cette méthodologie nous permettra d'obtenir les meilleures performances. Cependant cela nécessite de trouver des données suffisamment approchantes de nos commentaires You tube, au niveau de :

- La langue
- Longueur des commentaires
- Le vocabulaire

Cela commence par la recherche d'un jeu de données de textes courts, idéalement issus des réseaux sociaux, rédigés en français et étiqueté pour l'analyse de sentiment. Il existe peu
Pour vérifier que les commentaires ont significativement les mêmes longueurs entre 2 jeux de données, nous utilisons le test de Man-Witneyu, pour savoir si la distribution des longueurs des commentaires sont significativement différentes.
Ensuite, nous testons le degré de similitude au niveau du vocabulaire. Idéalement, le jeu de données doit couvrir une large partie du vocabulaire You tube pour que le modèle puisse être adapté à divers sujets.

Nous avons trouvé sur Kaggle un ensemble de 3 jeux de données potentiel pour l'entrainement :

- Des tweets traduit de l'anglais vers le français
- Des review de film du site allocine rédigés en français
- Des commentaires YouTube extrait de l'api rédigés en anglais

Il s'est avéré qu'il y a peu de jeu de données disponibles qui correspondent à nos critères. Le premier, même s'il s'agit de textes cours issue d'un réseau social, a été traduit en français mais la traduction semble ne pas être exacte. Le second risque de ne pas passer les tests de similitudes notamment au niveau de la longueur des textes mais est bien rédigé en français. Le troisième est le plus prometteur provenant de la même source que nos données mais doit être à une traduction rigoureuse.

Nous avons donc testé si les distributions de la longueur des commentaires étaient significativement différentes grâce au test de Mann–Whitney U (Wilcoxon rank-sum test). Il est utilisé pour comparer la distribution de 2 échantillons indépendant et test l'hypothèse H0 que les 2 échantillons sont issus de la même distribution contre l'hypothèse alternative H1 que la distribution sont différentes. Nous obtenons que le data set de tweet suit la même distribution que nos données réelles. Ce qui prévisible.

Ensuite nous avons utilisé le coefficient de Jaccard pour évaluer les similitudes au niveau du vocabulaire. C'est encore le data set de tweet qui obtient la plus grande similitude.

Pour répondre à la deuxième problématique nous avons entrainés différents modèles de machine learning et comparer leurs performances entre eux et des modèles pré-entrainé. Nous nous attendions à avoir de bon résultat mais le taux de bonne prédiction le plus élevé était de 57%. De plus lorsque que nous avons appliqué le meilleur modèle aux données réelles, nous avons constaté que les sentiments prédits étaient en désaccord avec ce qu'exprimaient les commentaires. Cela s'explique par la qualité de nos données. En effets les jeux de données sont, d'une part pas adapté et d'autre part de mauvaise qualité à cause d'une mauvaise traduction. Nous avons décidé pour la suite d'utiliser une modèle pré-entrainé avec ses résultats plus cohérents.

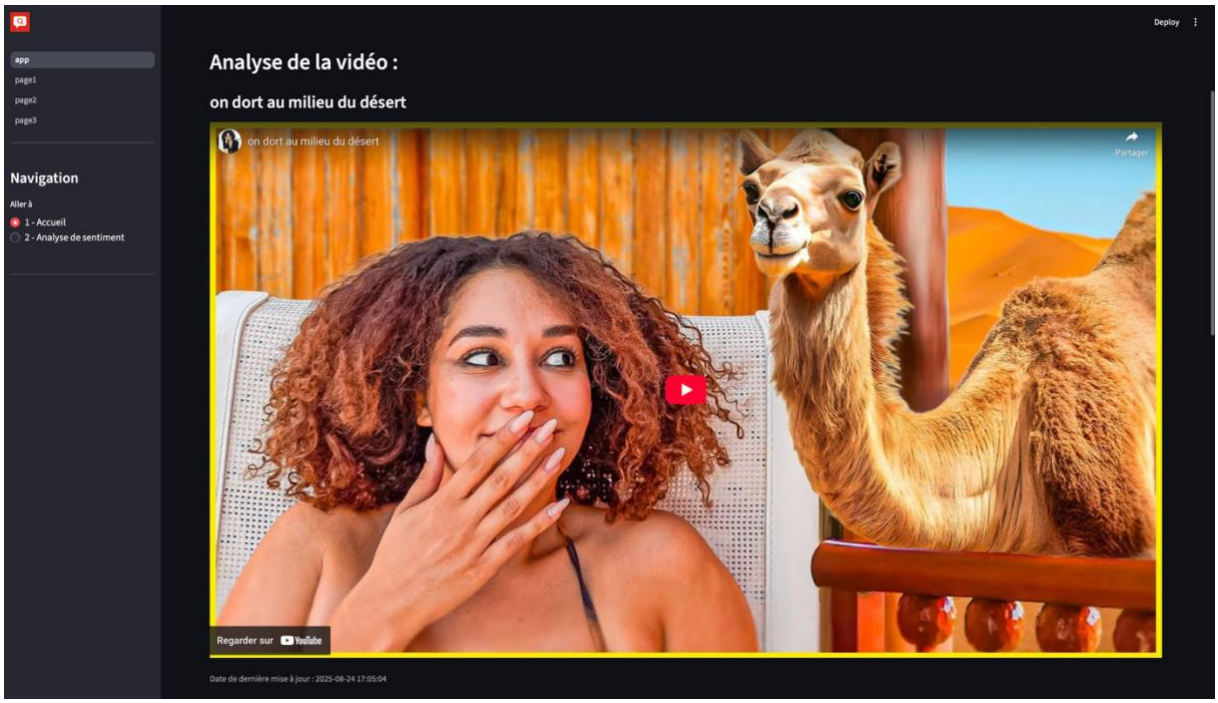
II. Visualisation des données, interprétation et communication des résultats

A. Visualisation des résultats de l'analyse

Étant donné que notre un outil d'analyse automatisé restitue les résultats sous forme de dashboard, nous retrouvons principalement les graphiques présentés précédemment.
Leur organisation se base sur le plan d'analyse :

- La première page présente les insights permettant de faire l'analyse de de surface
- La deuxième page regroupe résultats de l'analyse de fond

Les résultats sont présentés sur une page streamlit. La première information présentée est le du vidéo analysé accompagner du lien vers la vidéo. Même si l'utilisateur connait son contenu, cela constitue une aide visuelle, en plus du titre permettant de bien identifier le contenu traité. De plus si certains moments clés sont cités dans les commentaires, la vidéo est disponible pour visionner le moment en question. Sous l'image, on affiche également la date et l'heure de la dernière mise à jour.



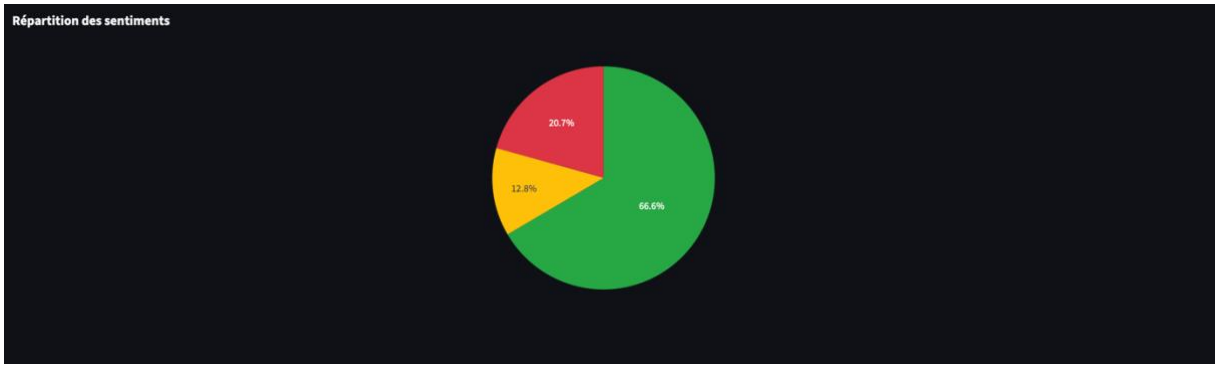
Ensuite, on représente les insights de l'analyse de premier niveau. On commence avec un cadre qui rappelle le nombre de commentaires traités, suivi d'un autre qui met en valeur le commentaire le plus liker. Ainsi disposez, la première information met en perspective la deuxième. On peut lire cette partie de cette façon : parmi les 1491 commentaire, l'idée qui a recolleté le plus de like est illustré par ce commentaire.

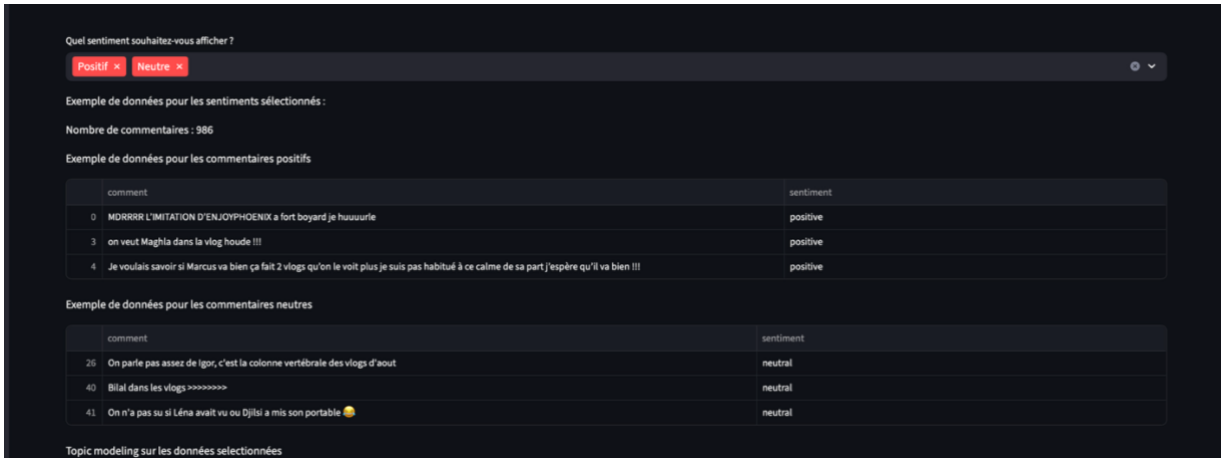
Ensuite, le graphique suivant est un barplot qui présente l'évolution du nombre de commentaire par jour. Ici la couleur utiliser n'a pas de réelle importance, à la différence du type de graphique choisi. En effet, nous avons volontairement utiliser un barplot au plutôt d'une courbe pour que la différence par jour soit plus visible. L'utilisation d'une courbe aurait atténuer cet écart que l'on peut constater et la nature discrète de la donnée implique l'utilisation d'une représentation qui permet une lecture précise des valeurs. De plus en survolant le graphique il est possible de voir les valeurs exactes dans des infos bulles. Elles redonnent la date exacte de publication et le nombre de commentaires publié, ce qui en fait de bons aide à la lecture.

La dernière information présentée dans la première page permet de faire la transition avec la deuxième page consacrée à l'analyse de fond. Le nuage de mots est une représentation ludique de la fréquence des mots dans l'ensemble des commentaires. Il est dimensionné de façon ne pas être trop petit ni trop grand, pour que l'information soit lisible dans être prédominante.

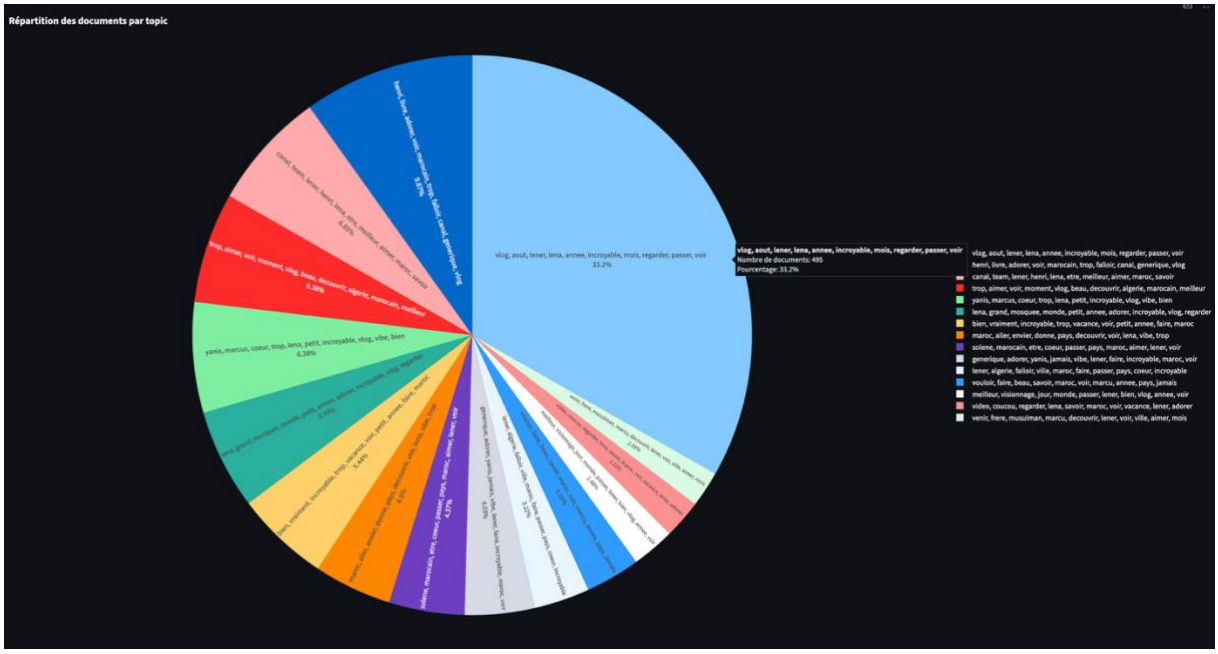
La deuxième page présente les résultats de l'analyse textuelle. La première information présenter est la répartition des commentaires en fonction du sentiment détecté. Le graphique de type « camembert » permet de bien représenter la part, l'important que chaque groupe représente. De plus chaque couleur est associé à un sentiment précis en fonction de sa connotation. Le vert renvoie au positif, le rouge au négatif et le jaune à ce qui est intermédiaire ou moyen. On utilise les couleurs comme langage commun en supposant qu'elles ont le même sens pour tout le monde. C'est la même logique que pour les feux tricolores, la couleur est aussi l'information. Ce graphique fonctionne aussi avec des info-bulles comme le précédent qui rappellent le sentiment concerné par la zone survolé et le nombre exacte de commentaires.

Le deuxième espace est dédié à l'illustration des sentiments. Pour chaque sentiment sélectionné un tableau affiche 3 exemples de commentaires. Les tableaux sont séparés et ils prennent tout le la largeur de la page pour plus de lisibilité.





La partie filtre permet également de filtrer les données pour l'analyse de sujet. S'ils sont tous les 3 sélectionné le topic modeling s'effectue sur toutes les données.
Là encore nous avons décidé d'afficher un graphique de type camembert, pour les mêmes raisons, la seule différence est le code couleur qui n'est plus appliqué. Nous ajoutons également quelques exemples de commentaire pour illustrer chaque sujet identifié.



B. Présentation de recommandations

Prenons les résultats précédents, obtenus après analyse des commentaires d'une vidéo des « vlog d'aout » de l'influenceuse, Léna Situation.

Nos recommandations s'articulent autour de 3 thèmes. Dans nous aborderons le sujet des interactions direct avec les abonnées, puis nous mettrons en avant un potentiel moteur d'engagement.

Tout d'abord, il apparait que le meilleur moment pour interagir avec les abonnées est les jour même où l'on poste la vidéo, voir le lendemain. C'est à ce moment qu'ils postent le plus de messages. On peut ajouter que cela montre que les abonnées « attendent » ou du moins sont présent sur la plateforme au moment de la publication. Sachant qu'il s'agit ici d'une vidéo issue de la série de vlog d'aout, les abonnées savent que tous les soirs pendant 1 mois il y aura une publication. Nous avons donc identifié un moment favorable et stratégique pour le contact. Dans le même, nous en déduisons qu'il est préférable de publier du contenu en fin d'après-midi ou début de soirée, pour être assez en forme pour réagir en direct aux commentaires.

Ensuite, faire participer ses abonnés à l'élaboration du générique lui permettra d'augmenter l'engagement de ses abonnés. C'est ce qui est abordé dans le commentaire le plus liké, avec 6 387 likes actuellement. Cela permettra de faire participer ses abonnés au contenu et de leur donner une raison supplémentaire de le visionner. Dans la même idée, faire intervenir des personnes extérieures dans son contenu semble plaire car cela apparait dans les commentaires positifs.

III. Support et accompagnement des utilisateurs

A. Support de formation

Ce document, présente les règles et mode d'utilisation de l'application « YOU REVIEW ». L'objectif est de vous permettre d'utiliser l'outil en total autonomie, vous sensibiliser à l'utilisation des données analysées en intégrant certaines bonnes pratiques et garantir votre compréhension lors analyse des indicateurs clés. Il s'adresse à toutes les personnes autorisées à utiliser l'application. C'est-à-dire toute personnes ayant un chaîne YouTube et possédant un identifiant et mot de passe permettant d'utiliser l'application. Si vous utilisez cette application vous vous engagez à respecter les règles et les conditions d'utilisation de l'outil et des données de You Tube.

Les données

Les données analysées et valorisées par l'outil sont des données de type textuelle, collectées via l'api de You Tube. Ce type de données est soumis à la réglementation RGPD car il s'agit de données sensibles. En effet, caractère sensible des données s'explique par le fait qu'il est possible qu'elles communiquent des informations sur l'orientation politique, sexuelle, l'origine ethnique et l'état mentale des personnes. C'est pourquoi les données présentées sont anonymes. L'utilisateur est donc tenu de n'utiliser ces données qu'à des fins professionnelles et en aucun cas essayer de d'utiliser ces données pour discriminer des individus.

Les indicateurs clés

Cette partie est dédié à la présentation des indicateurs présentés et la compréhension de leur construction. La majorité des indicateurs sont de simples comptage ou affichage de données brutes.

Le nuage de mot est une représentation des mots les plus fréquents dans l'ensemble des commentaires. Plus un mot est fréquent plus il sera gros.

Les résultats de la page 2 ont été obtenus en utilisant de modèle déjà entrainés sur d'autres données. Nous n'avons pas supervisé leur entrainement et ne pouvons donc pas expliquer les résultats. Il faut donc comprendre que ces résultats ne sont pas absolus. Plus particulièrement pour le Topic Modeling, les sujets identifiés ne sont pas figés et peuvent être sujet à interprétation. En claire si vous obtenez 10 sujets, il n'y en a peut-être en réalité que 5. Il faut garder son sens critique et ne pas prendre les résultats tel qu'ils sont.

Les fonctionnalités

L’outil s’articule autour de 2 pages résumées dans la partie gauche de l’application dans la partie « Navigation ».

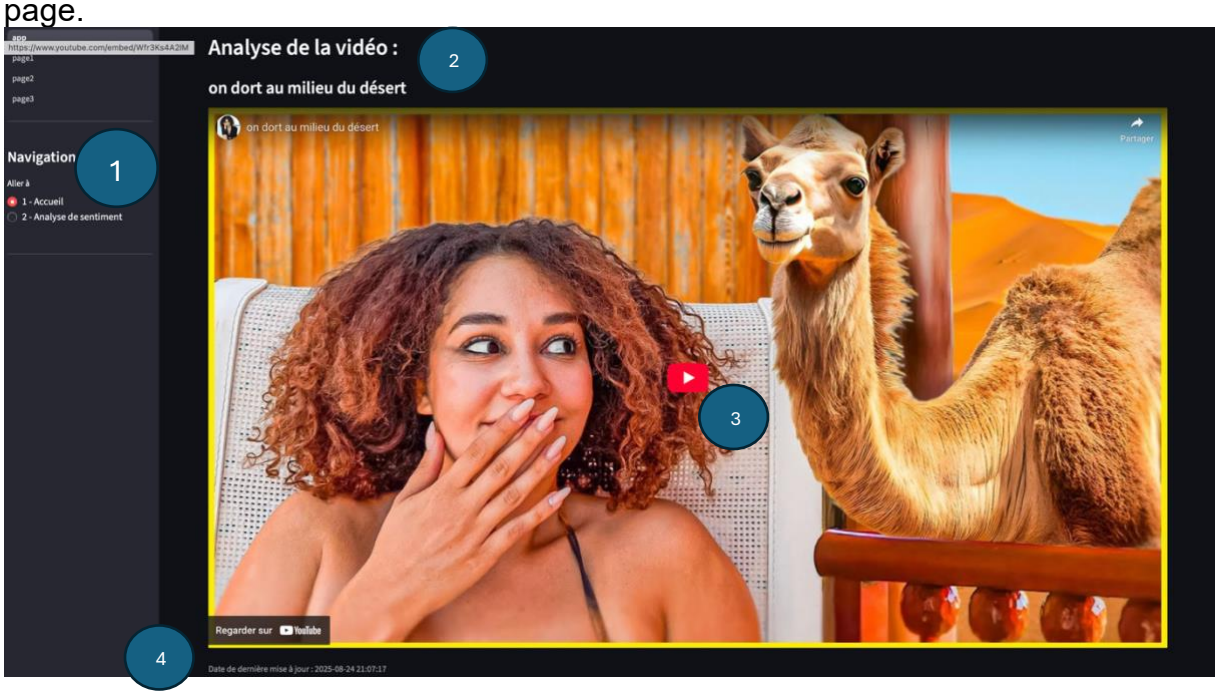
Lors du démarrage de l’application, l’utilisateur se trouve par défaut sur la page 1, la page d’accueil. La fonctionnalité principale de cette page est de lancer l’analyse. Pour ce faire vous devez renseigner une url YouTube valide dans la partie formulaire puis cliquer sur le bouton « Lancer l’analyse ».

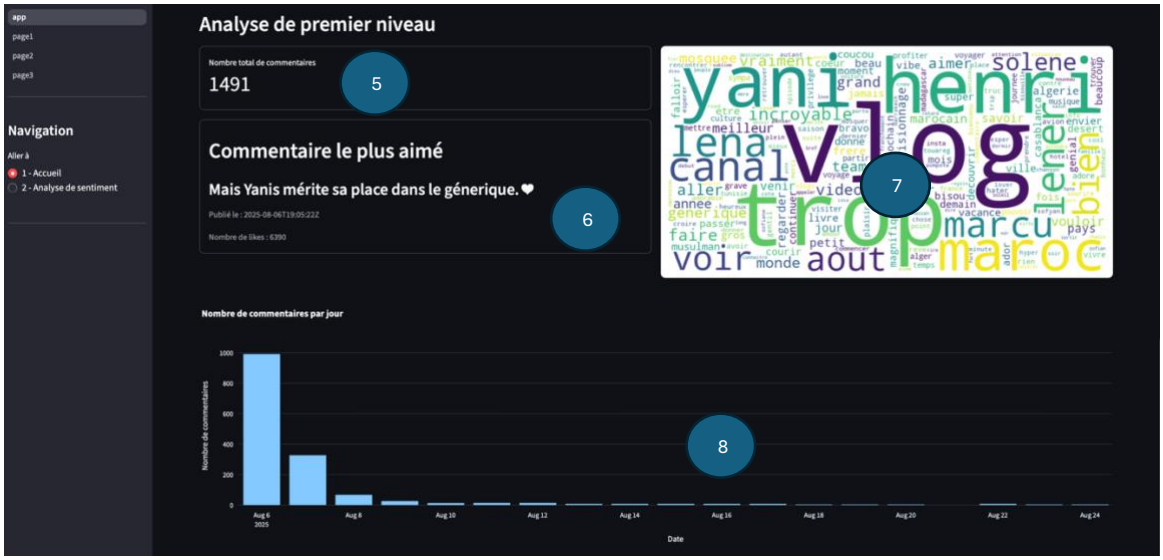
Point de vigilance :

- Si l’url n’est pas valide l’application ne fonctionnera pas et un message d’erreur s’affichera pour vous demander de rentrer une url valide.
- Lorsque vous entrez une url, il est obligatoire d’appuyer sur le bouton pour lancer l’analyse. Taper sur la touche « entrer » du clavier ne fonctionnera pas.

Une fois l’url validé, l’analyse peut démarrer. Cela peut prendre un peu de temps si c’est la première fois que l’utilisateur entre cette url dans l’outil. Dans ce cas un message s’affichera pour vous annoncer que les données vont être collectées.

Les images suivantes présentent les différents éléments présents dans la première page.

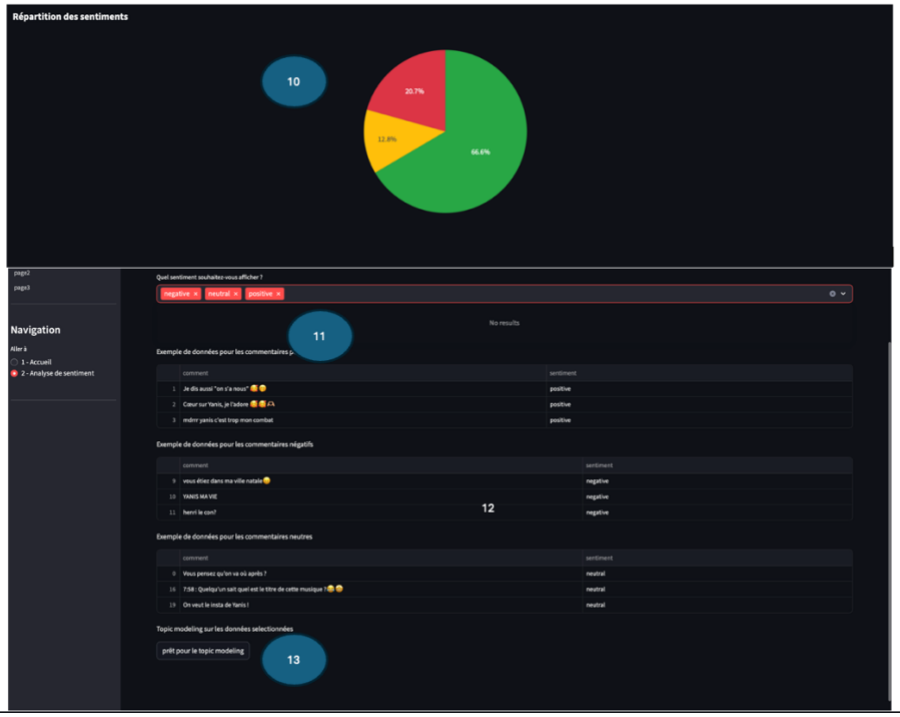


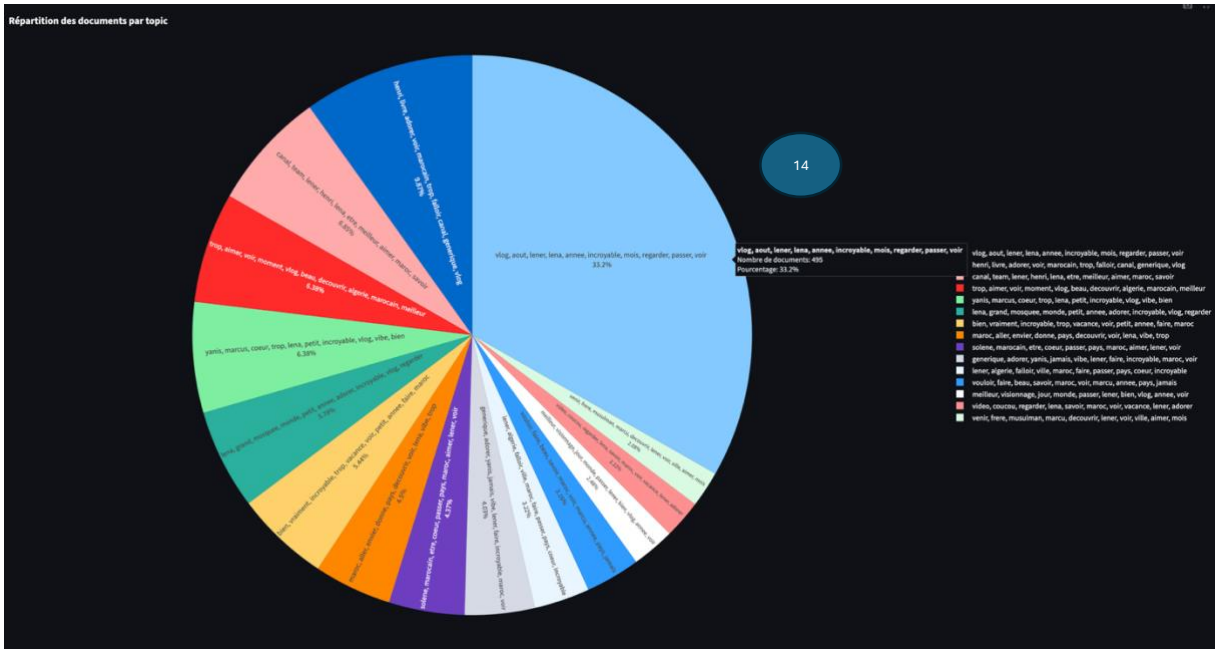


Nous retrouvons les éléments suivants :

- La partie navigation (zone1)
- La partie formulaire avec le bouton d'analyse
- Le titre de la vidéo (zone 2)
- La vidéo (zone 3)
- La date de la dernière mise à jour (zone 4 sous la vidéo)
- Le nombre de commentaires (zone 5)
- Le commentaire avec le plus de like et sa date de publication (zone 6)
- Le nuage de mots (zone 7)
- Le graphique de répartition du nombre de commentaires en fonction de la date de publication (zone 8)

L'image suivant présente les différents éléments de la deuxième page.





Les éléments sont :

- Graphique de réparation des commentaires selon le sentiment (zone 10)
- Filtre de sentiment (zone 11)
- Résultats du filtre (zone 12)
- Bouton de lancement pour l'analyse des sujets (zone 13)

Le filtre est une liste déroulante à choix multiples qui permet d'afficher des exemples de commentaires dans la zone 12.
Le bouton en zone 13 se base sur le filtre de la zone 11 pour lancer le topic modeling et affiche en zone 14 le graphique de répartition.

B. Documentation technique

Ce document s'adresse à des personnes ayant les profils suivants :

- Data scientist
- Data engineer

Le but est de permettre la transmission et la compréhension de l'outil d'un point de vue technique pour le profils spécifiés.

La source de données

Les données traitées par l'outil sont collectées via l'api You Tube de Google. Les champs collectés sont les suivants :

- `Id` : l'identifiant unique du commentaire
- `channelId` : l'identifiant de la chaîne YouTube
- `publishedAt` : date de publication
- `textOriginal` : le texte du commentaire tel qu'il a été publié ou mis à jour.
- `likeCount` : le nombre de like du commentaire

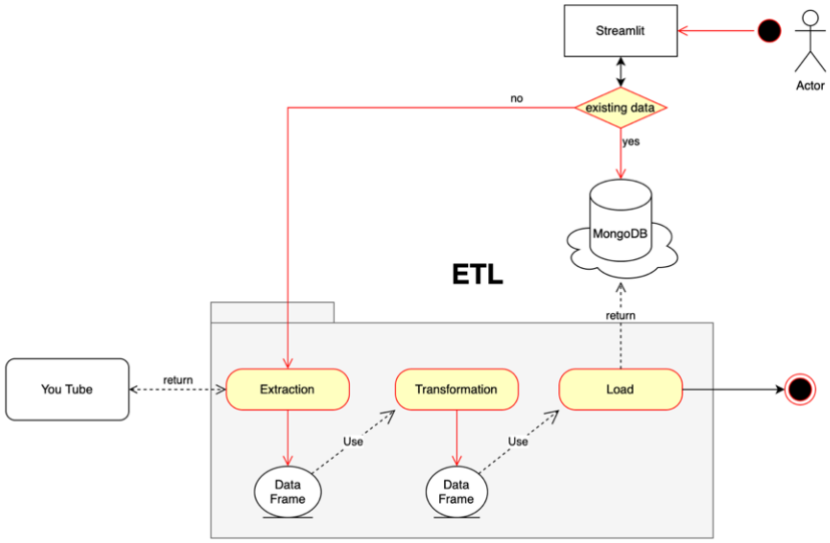
- `viedoId` : identifiant de la vidéo
- `authorChannelId` : l'identifiant de la chaîne de la personne qui écrit le commentaire
- `title` : titre de la vidéo
- `description` : description de la vidéo
- `commentCount` : le nombre de commentaires de la vidéo

Le périmètre se limite aux vidéos françaises avec des commentaires rédigés en français. De plus aucune donnée de type pseudo n'est collectée.

L'utilisation de l'api est sécurisée par une clé d'accès et les appels sont limité par un quota quotidien imposé par You Tube.

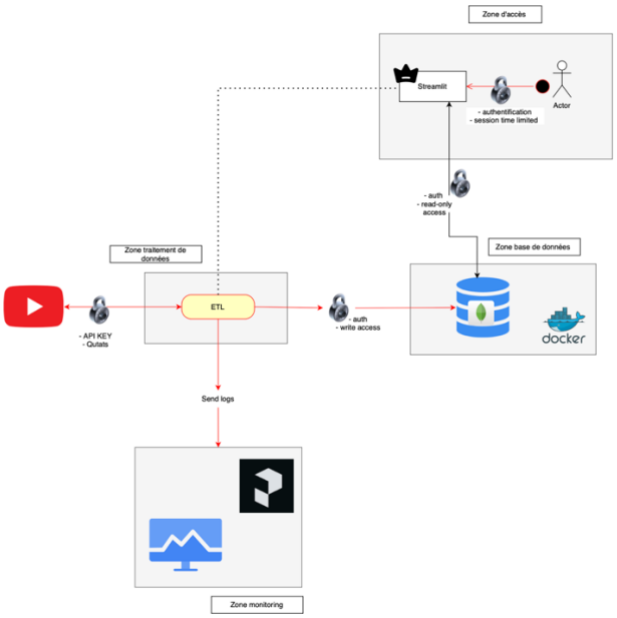
L'architecture du projet

L'architecture du projet suit le schéma suivant :



Le point d'entrée est l'application streamlit qui soit interrogé la base de données, soit déclenche le processus ETL.

Le schéma suivant présente d'architecture sécurité du projet :



L'orchestration des tâches et technologies utilisées

L'ensemble du projet a été développé en python (3.10.6) et fait intervenir les technologies suivantes :

- Docker
- Prefcect
- Mongodb

Il repose sur en ensemble de « requirements » à la base du projet, qui permet de construire un environnement de développement adapté.
Le projet repose sur différents services. Un service docker qui permet d'héberger notre base de manière sécurisé. Un ensemble des modules pythons qui permettent d'extraire, transformer (et analyser) puis de charger les données dans différentes bases de données sous mongodb. Le troisième service est l'orchestrateur Prefect qui permet d'exécuter et monitorer nos différents flows.
Tous ces services sont connectés via différents ports.