Assignment 1:  
Youtube Data and Google News Data

Joshwa Siju Thomas  
Artificial Intelligence & Machine Learning  
*Lambton College*Toronto, Canada  
C0903528@mylambton.ca

Ashick Rijo  
Artificial Intelligence & Machine Learning  
*Lambton College*Toronto, Canada  
C0910364@mylambton.ca

Rohan Aryan  
Artificial Intelligence & Machine Learning  
*Lambton College*Toronto, Canada  
C0912902@mylambton.ca

Nipun Narag  
Artificial Intelligence & Machine Learning  
*Lambton College*Toronto, Canada  
C0910386@mylambton.ca

Contents

[**Introduction** 3](#_Toc180916016)

[**PART – 1: YouTube Data Analysis**: 3](#_Toc180916017)

[**Data Collection Procedure:** 3](#_Toc180916018)

[**Data Cleaning Steps:** 3](#_Toc180916019)

[**Result of Analysis:** 3](#_Toc180916020)

[**PART -2: Google News Data Analysis:** 5](#_Toc180916021)

[**Data Collection Procedure:** 5](#_Toc180916022)

[**Data Cleaning Steps:** 5](#_Toc180916023)

[Result of Analysis: 5](#_Toc180916024)

[**References** 8](#_Toc180916025)

# **Introduction**

This projects deals with using of YouTube API (V3) to scrape data for about 26K records and details out procedures to clean and format the data, further shows visual details about some videos based on various factors. The Later part of this report talks about procedures to scrape data from Google news website using Pygooglenews tool. Details out procedures to clean and shows plots for descriptive features of the dataset. Further the data is cleaned and used for topic modelling (using LDA).

# **PART – I: YouTube Data Analysis**:

## **Data Collection Procedure:**

The data collection process followed the API documentation provided by the Google Developer Console to ensure accuracy and adherence to best practices. As part of the project, the team was tasked with retrieving metadata and comment data, including statistics like like count, comment count, and view count, for a given set of video IDs.

To achieve this, the team utilized two primary API calls, commentThreads and videos, to gather the necessary data. A global class (user-defined) was created to manage the core functions for data retrieval and processing, ensuring modularity and reusability within the notebook. After securely establishing the API connection using the API key, each video ID was iterated through, with exception handling in place to maintain data consistency, even for unavailable videos.

The data retrieval was divided into six subsets, with each team member responsible for one subset. This distribution helped maximize efficiency, ensuring that the process remained within the daily quota limits while minimizing delays.

## **Data Cleaning Steps:**

To maximize data retention and minimize void records, we systematically removed any entries where all feature values were empty, which typically indicated that the videos were either unavailable for retrieval or had been deleted. The original dataset comprised approximately 25,623 video IDs, and following this initial cleaning, the team retained 15,141 records with nine distinct features.

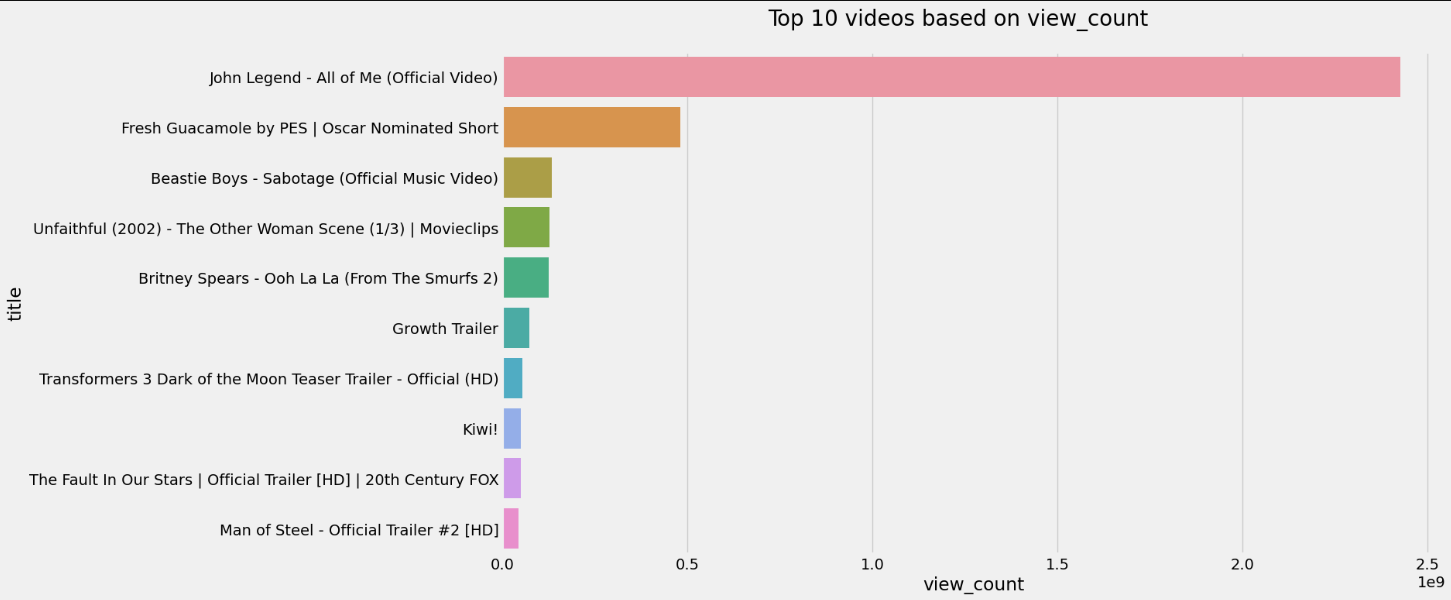
Each feature was carefully analyzed and understood, leading to a comprehensive report on the metadata associated with the dataset. To further establish consistency, we corrected the index column and ensured that all feature indices were appropriately aligned.

For the numerical features, we ensured consistency by converting the data types to numerical formats, enabling accurate analysis. Additionally, we addressed the video duration values, which were initially in a different format. New features were created to store each variable derived from the corrected duration format, allowing for a more precise analysis of video length.

## **Result of Analysis:**

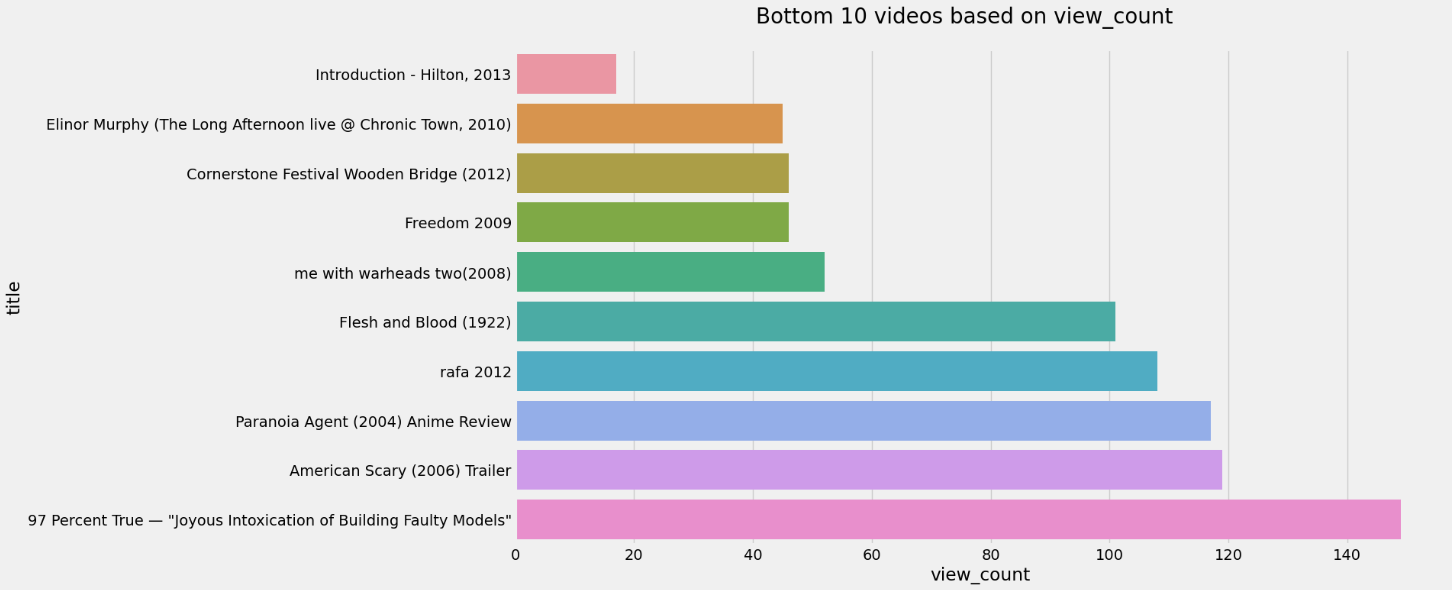
We then generated plots to gain insights on the following trends:

1. A list of the top-10 videos based on the total views:

  
*fig-1:- Top 10 Videos based on Views.*

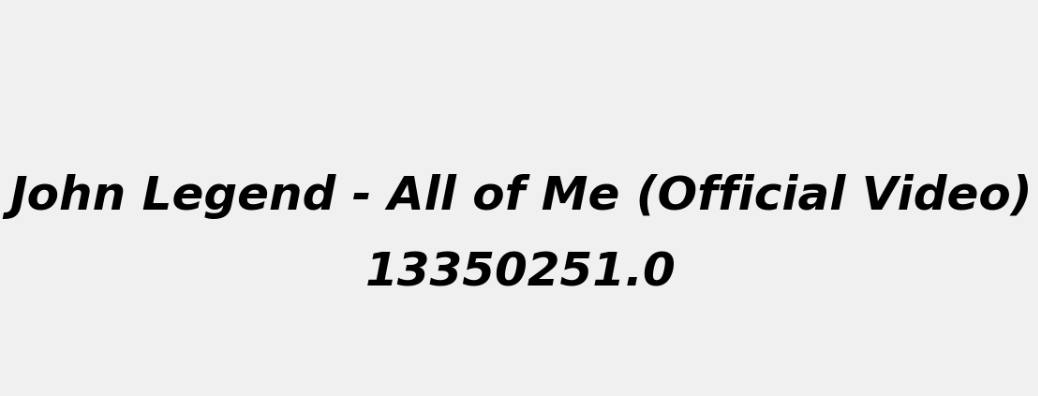
The video with the highest views in the dataset was:   
**John Legend- All of Me (official Video)** – with over **2.4B views**.

2. A list of the bottom-10 videos based on the total views:

  
*fig-2:- Bottom 10 Videos based on Views.*

The video with the Least views in the dataset was:   
**Introduction – Hilton, 2013**– with less than **20 views**.

3. The most liked video:

  
*fig-3:- Most Liked video.*

The video with the most likes in the dataset was:   
**John Legend- All of Me (official Video)** – with over **13.3M views**.

4.The least liked video:

  
*fig-4:- least Liked video.*

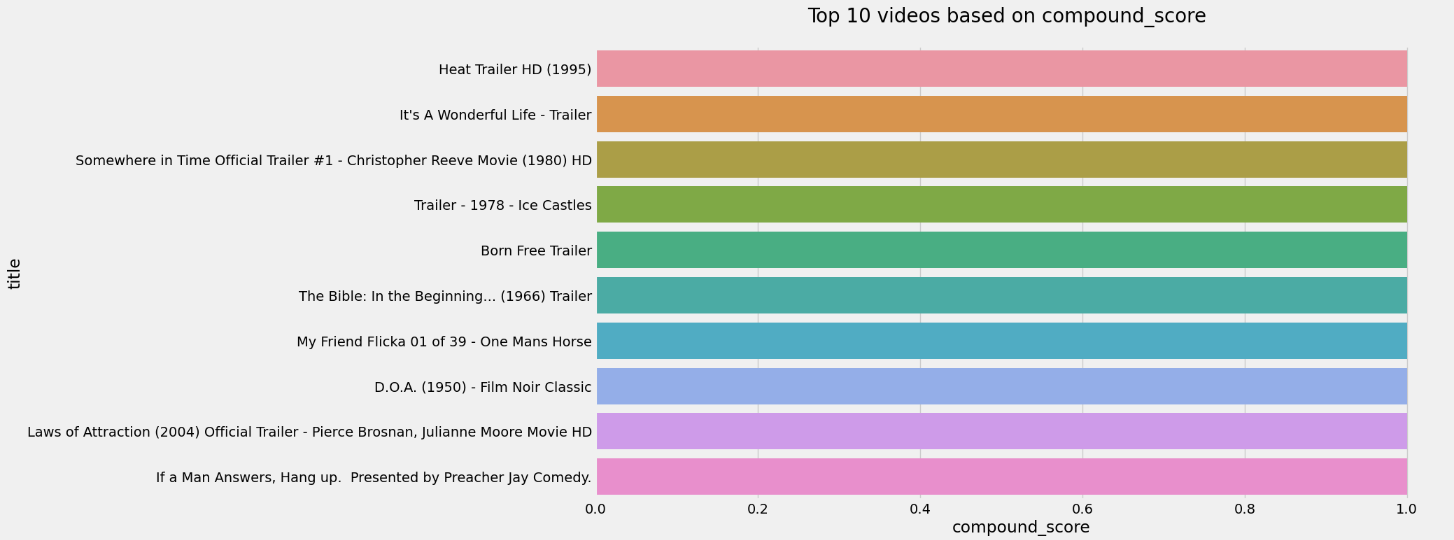
The video with the least likes in the dataset was:   
**Naked In New York Trailer 1994** – with **0 likes**.

5. The video with the highest duration:

  
*fig-5:- Longest Video.*

The video with the highest duration in the dataset was:   
**Escape Me Never (1935) 1/9** – **3 Days: 2 Hours: 40 Mins: 22 Secs long!**

6. The top-10 videos that have the highest positive sentiment scores:

  
*fig-6:- Top 10 most positive videos.*

# **PART -II: Google News Data Analysis:**

## **Data Collection Procedure:**

For this phase of the project, the team opted to utilize an open-source tool available on GitHub called PyGoogleNews. This tool facilitates the retrieval of news articles and relevant information from Google News by sending HTTP requests to the website, parsing the resulting HTML content, and extracting pertinent data in a structured manner.

The team focused on extracting information related to eight specific topics, including world news, science, health, and others, across thirteen countries, all in English. This targeted approach allowed for a comprehensive overview of current events and trends in various fields.

The data collection process was straightforward: we defined a function utilizing PyGoogleNews, where all parameters—such as topics, countries, and language—were inputted. Upon executing this function, the tool retrieved the corresponding news articles, which were then stored in a dictionary for easy access and manipulation. Subsequently, the collected data was saved as a CSV file, enabling further analysis and visualization.

By leveraging PyGoogleNews, the team efficiently gathered a wealth of information, setting the foundation for subsequent data cleaning and analysis tasks. This structured approach not only streamlined the data collection process but also ensured a rich dataset for exploration.

## **Data Cleaning Steps:**

The cleaning procedure commenced by ensuring that the published\_date feature was consistently formatted as a datetime object. This step was crucial for accurate temporal analysis, allowing us to leverage date-related functionalities in later stages of our analysis. Each variable from the dataset was then organized into separate columns for clarity and ease of use.

Next, we focused on the summary feature, which contained HTML content from the retrieved data. To extract meaningful text from this HTML, we developed a function that parsed the content, stripping away the HTML tags and retrieving the underlying text. This process ensured that our summaries were clear and ready for analysis.

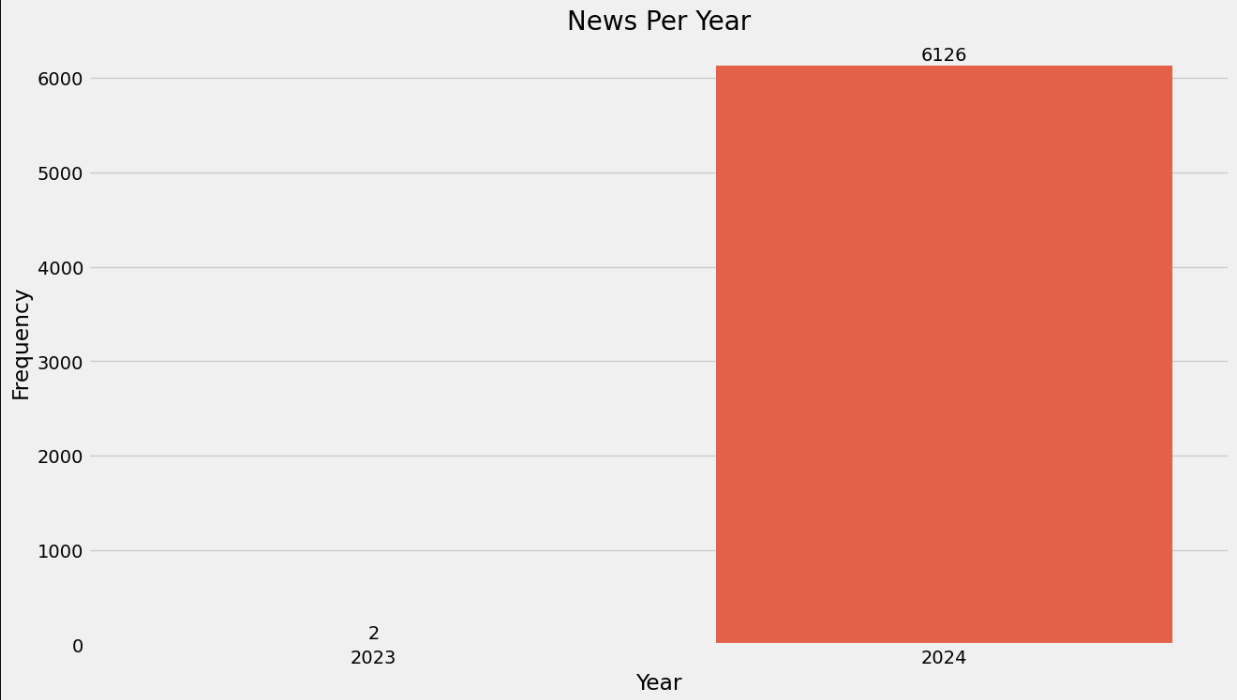
The team also addressed the news\_agency feature, which included unwanted extensions such as .gov, .au, .org, and others. By implementing a cleaning function, we were able to remove these extensions, standardizing the entries and improving the quality of the data.

To gain deeper insights into the writing trends of the news summaries, we created three new features: word\_count, sent\_count, and stopwords\_count. The word\_count feature allowed us to quantify the length of each summary, while the sent\_count feature provided the number of sentences. The stopwords\_count helped identify the prevalence of common, less informative words in the summaries. This analysis aimed to uncover any correlations between the length and structure of the summaries and the topics being covered.

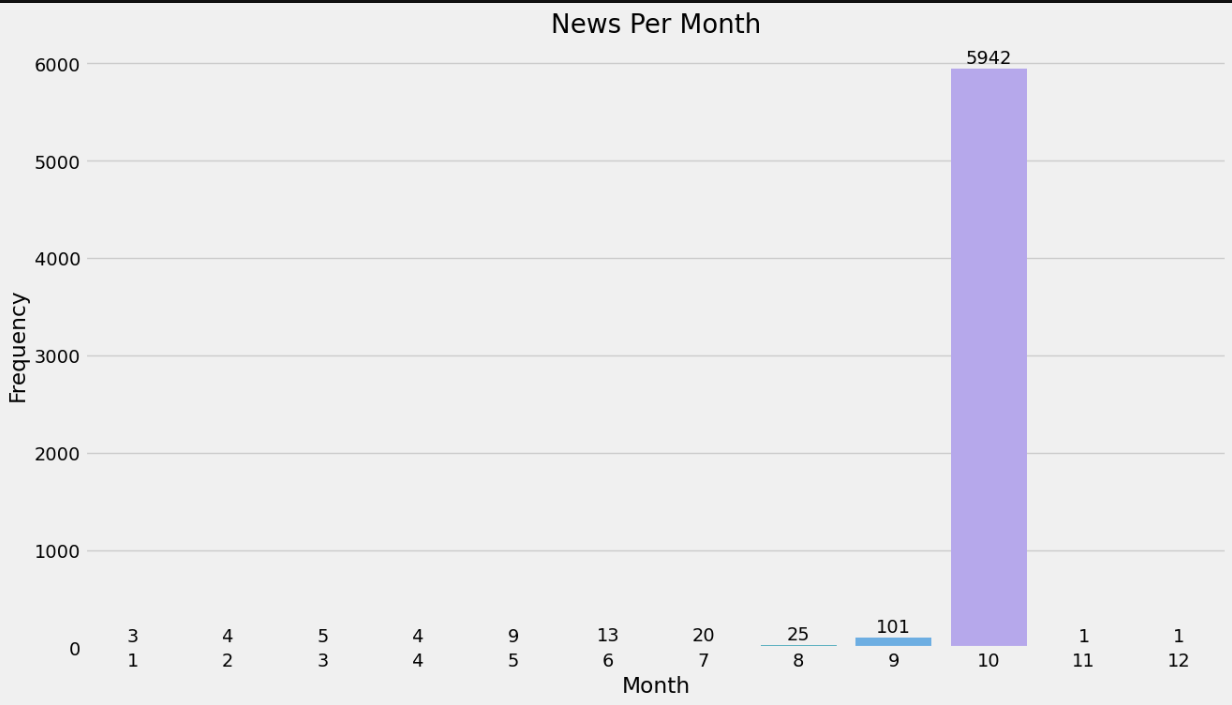
Finally, we conducted a review of the dataset and dropped redundant features that did not contribute meaningfully to our analysis. This decision was made to streamline the dataset, ensuring that our focus remained on the most relevant variables for subsequent analytical tasks.

## Result of Analysis:

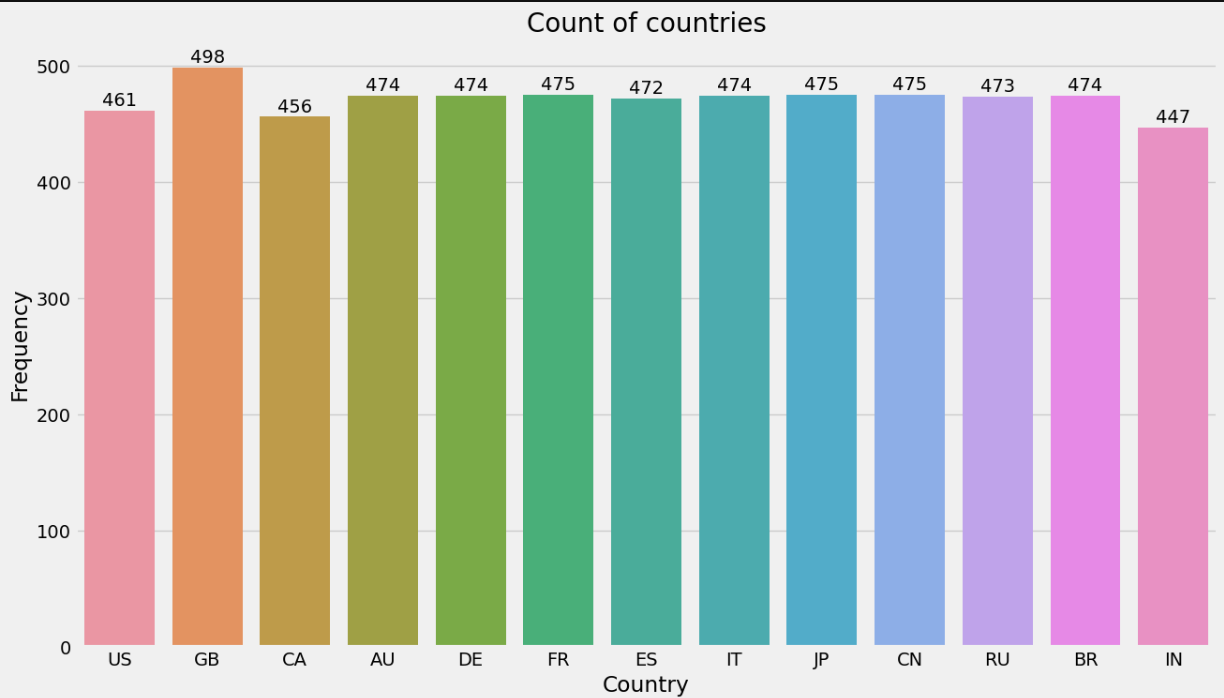
We then generated plots to gain insights on the following trends:

  
*fig-7:- Frequency of News Articles per Year.*

Most of the data we encountered was from the year **2024.**

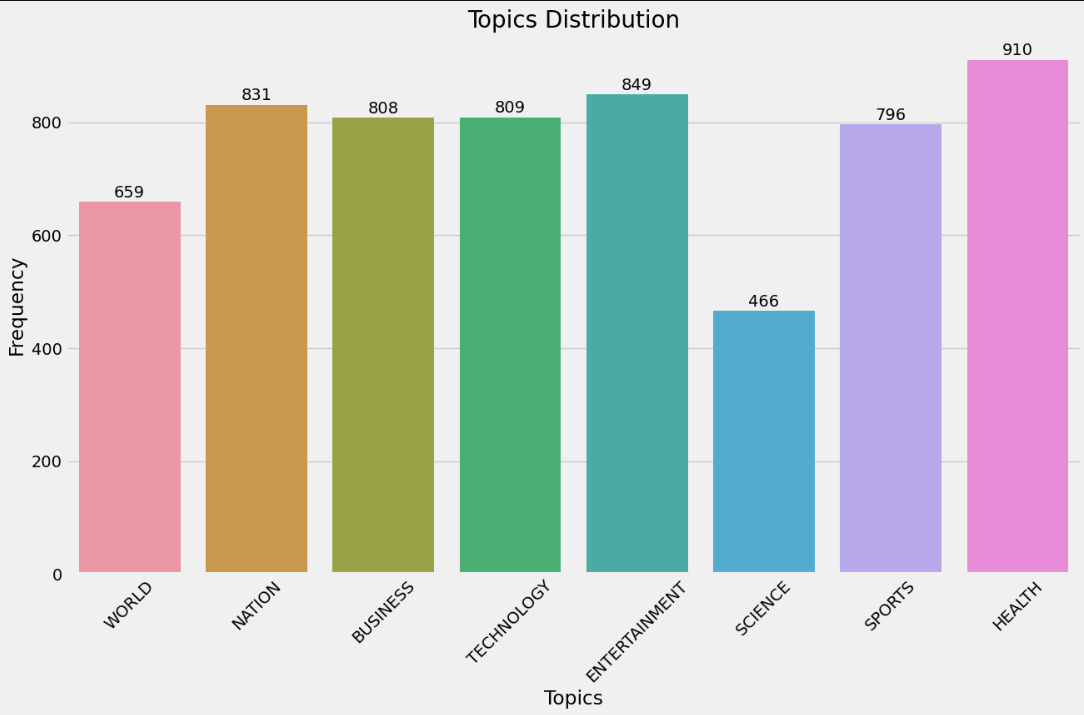
  
*fig-7:- Frequency of News Articles per Month.*

The News data we retrieved was from the 10th month of the year, indicating that the data collected was of recent times.

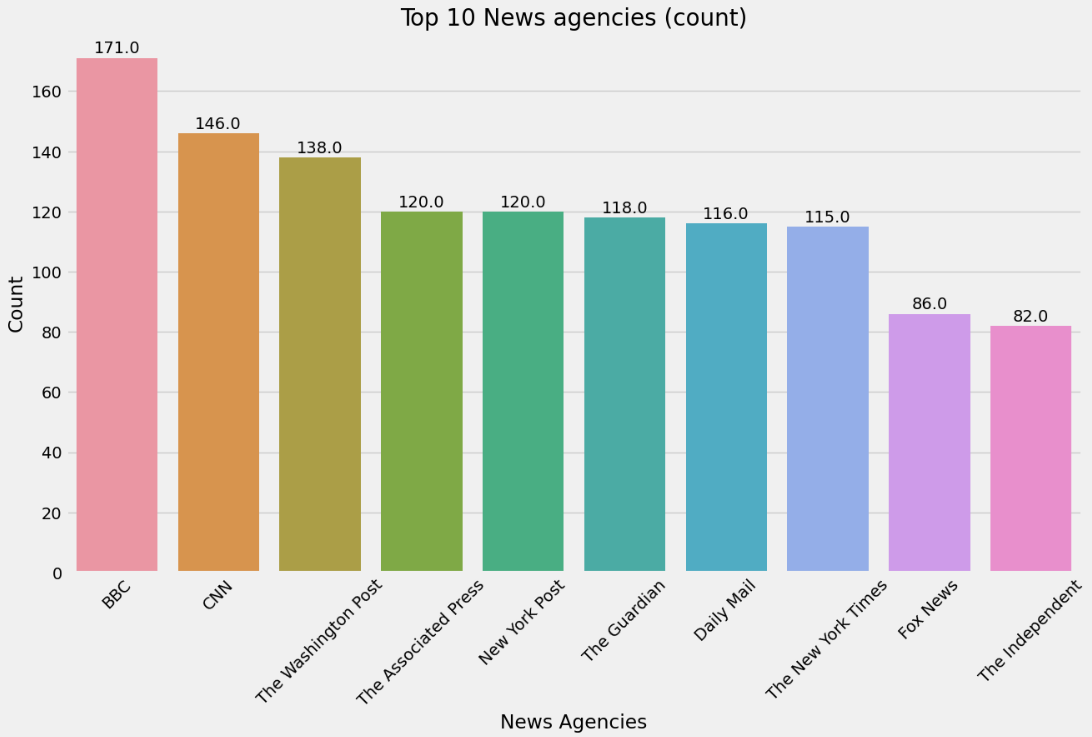


*fig-8:- Count of News Articles per Country.*

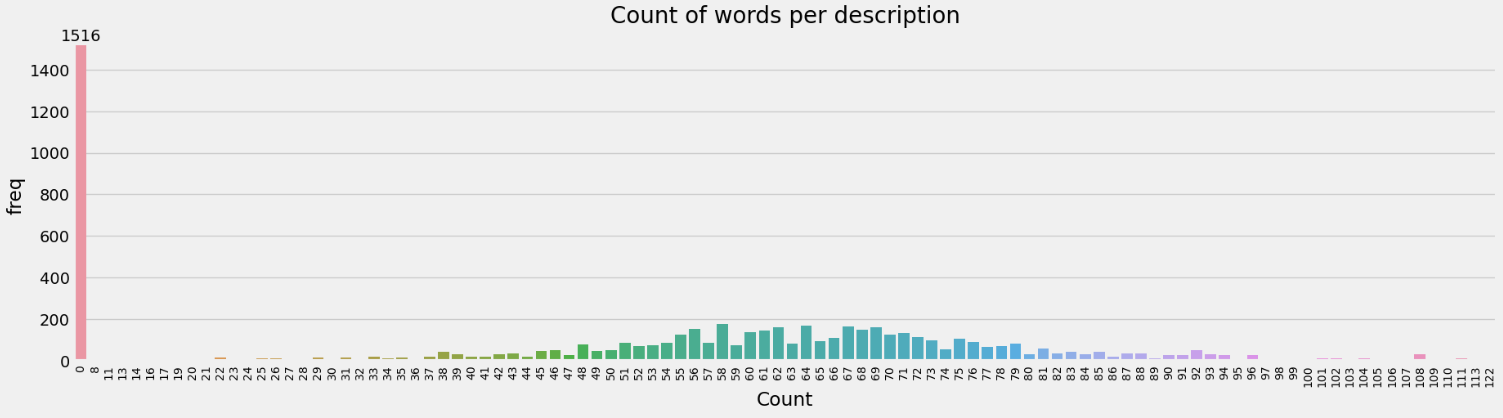
Dataset showed significant balance across all contries, with a little bias towards GB (Great Britan).

  
*fig-9:- Count of News Articles per TOPIC.*

Dataset showed significant balance across all contries except an imbalance in World and Science news.

  
*fig-9:- Count of News Articles per TOPIC.*

The Most Occuring News agency in the dataset was BBC, followed by CNN and The Washington Post.

  
*fig-10:- Distribution of words in summary across the dataset.*

Most of the word count across all topics was shown consistent behavior, but we encountered most records having no summaries in the dataset, which was handled in the preprocessing related to topic modelling (explained further).

**WORDCLOUDS:**

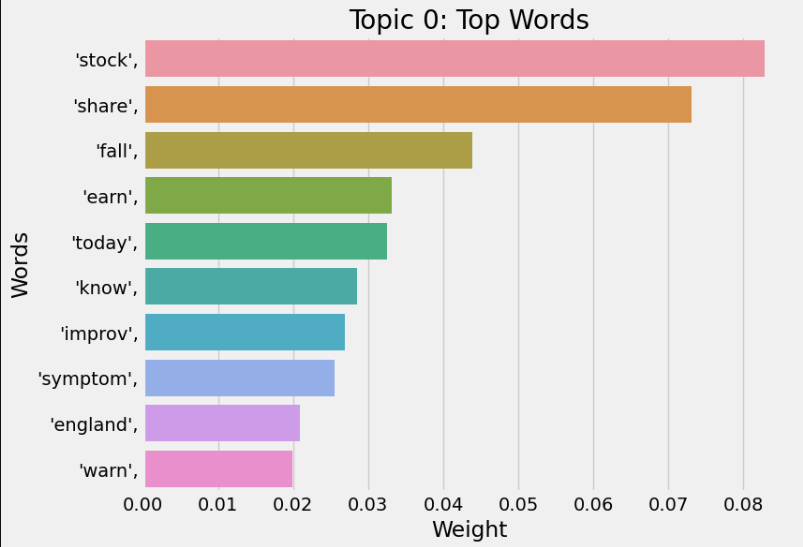
The team plotted wordclouds to understand distribution of words in 2 features. Before and after preprocessing

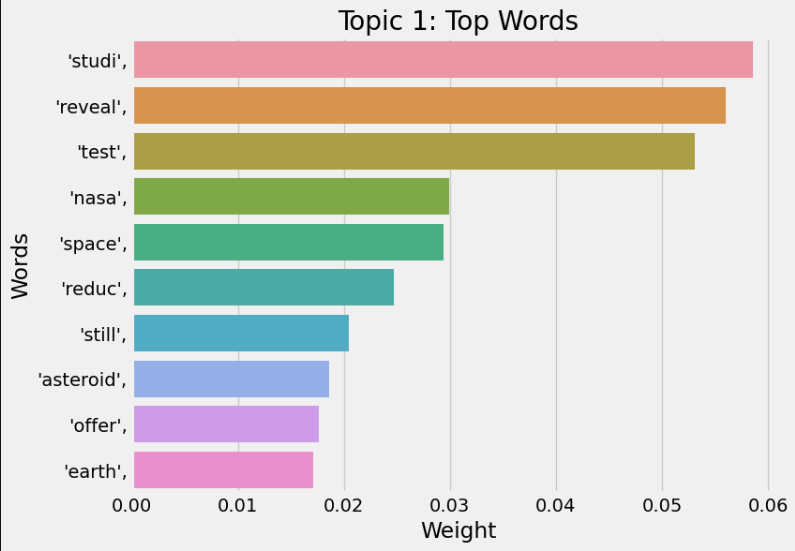
  
*fig-11:- Wordcloud (MERGED\_TS).*

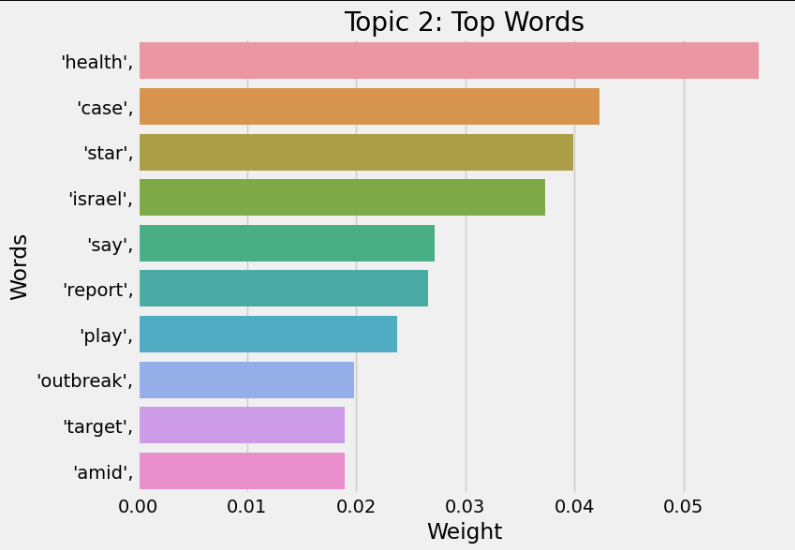
*  
fig-12:- Wordcloud (Preprocessed text).*

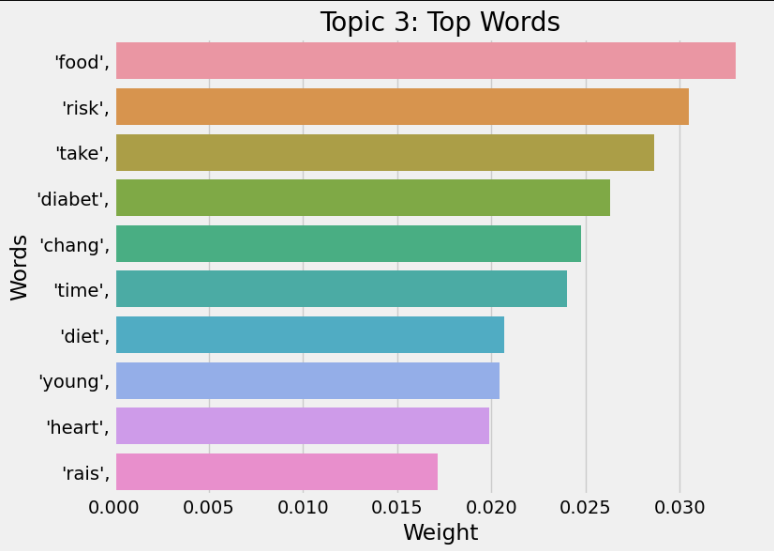
Top 5 topics and top 10 words in each topic (based on LDA model) :

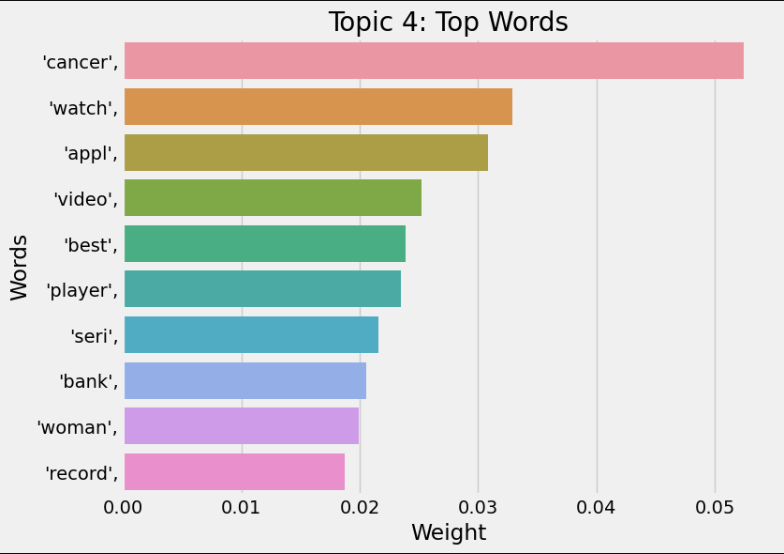
The team fed the preprocessed data into LDA model to understand the top 10 occuring topics as per the model. Here is a plot to show top 5 topics and its respective top 10 words.











# **References**

*API Reference | YouTube Data API | Google for Developers*. (n.d.). Google for Developers. <https://developers.google.com/youtube/v3/docs/?apix=true>

*Beautiful Soup Documentation — Beautiful Soup 4.12.0 documentation*. (n.d.). <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

*pandas documentation — pandas 2.2.3 documentation*. (n.d.). <https://pandas.pydata.org/docs/>

*NumPY Documentation*. (n.d.). <https://numpy.org/doc/>

*seaborn: statistical data visualization — seaborn 0.13.2 documentation*. (n.d.). <https://seaborn.pydata.org/>

*Matplotlib documentation — Matplotlib 3.9.2 documentation*. (n.d.). <https://matplotlib.org/stable/index.html>

Kotartemiy. (n.d.). *GitHub - kotartemiy/pygooglenews: If Google News had a Python library*. GitHub. <https://github.com/kotartemiy/pygooglenews>

*NLTK :: Natural Language Toolkit*. (n.d.). <https://www.nltk.org/>

*Regular Expression HOWTO*. (n.d.). Python Documentation. <https://docs.python.org/3/howto/regex.html>

*NLTK :: nltk.sentiment.SentimentIntensityAnalyzer*. (n.d.). <https://www.nltk.org/api/nltk.sentiment.SentimentIntensityAnalyzer.html?highlight=sentimentintensity>