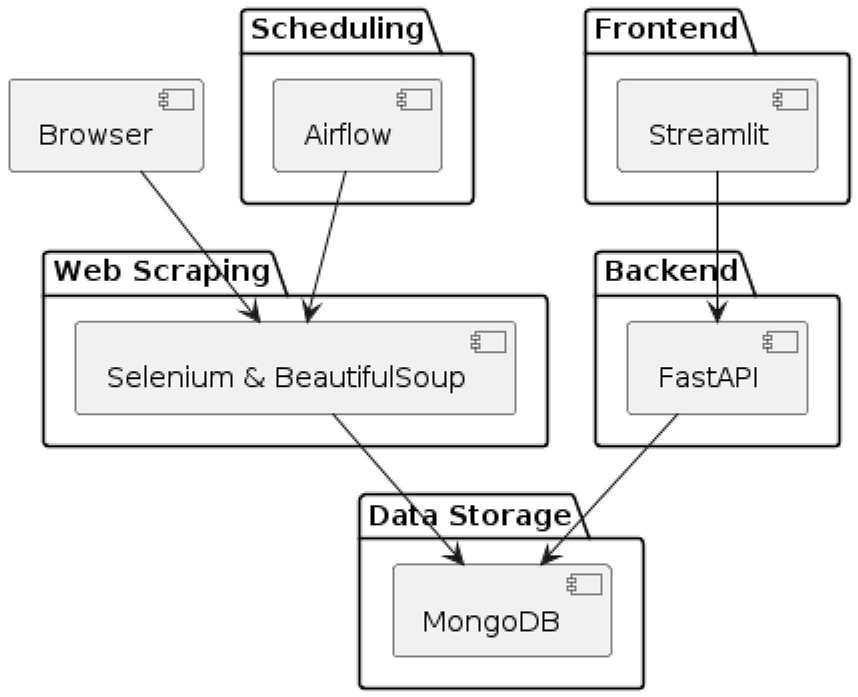
Introduction :

This report provides a comprehensive overview of the data engineering project aimed at scraping data from the BBC website, storing it in MongoDB, performing analysis, creating a backend Fast API, and presenting the data on a frontend using streamlit. The report includes details on each task, explanations of the chosen technologies, and the rationale behind the decisions made throughout the project.

Architecture Diagram :

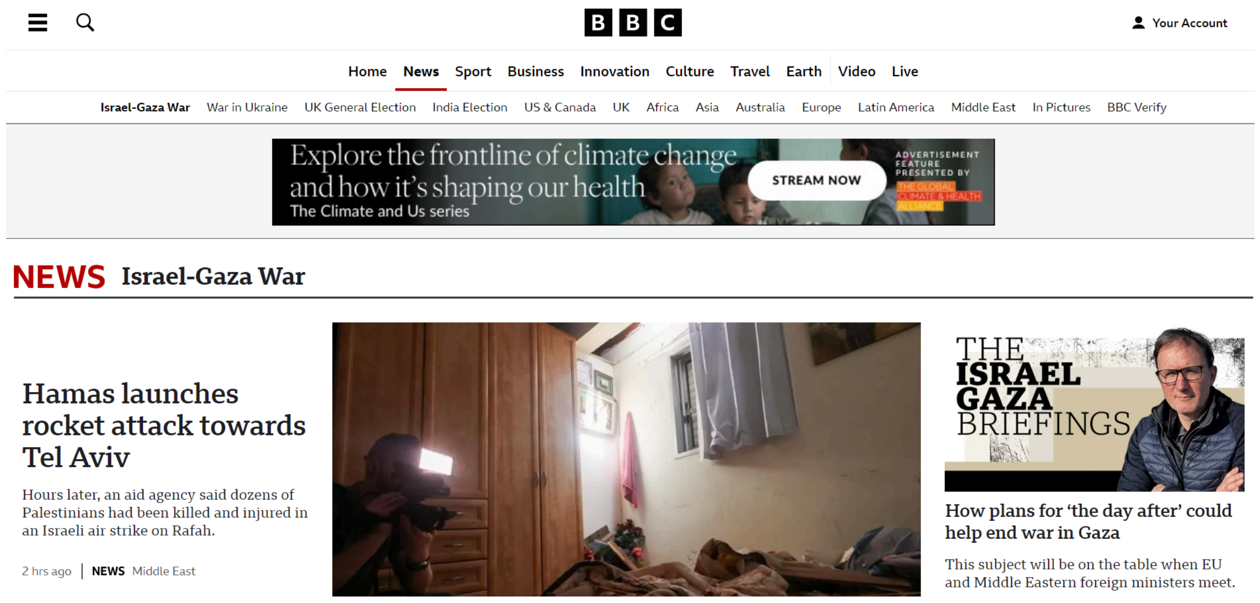
This project involves the continuous scraping of the BBC News Articles, storing the data in MongoDB, performing analysis, and creating a backend and frontend to display the analysis results. The implementation uses Airflow for scheduling, Selenium and BeautifulSoup for scraping, MongoDB for data storage, FastAPI for the backend, and Streamlit for the frontend. Here's the architecture diagram created using PlantUML:



I. Data Scraping :

Overview :

The data scraping process involves extracting information from the BBC website. Using **Selenium** and **BeautifulSoup**, we retrieved articles' data, including titles, subtitles, dates, authors, text, images, and video links.



BBC
Your Account
Home News Sport Business Innovation Culture Travel Earth Video Live
Israel-Gaza War War in Ukraine UK General Election India Election US & Canada UK Africa Asia Australia Europe Latin America Middle East In Pictures BBC Verify
Explore the frontline of climate change and how it's shaping our health
The Climate and Us series
STREAM NOW
ADVERTISEMENT FEATURE PRESENTED BY
NEWS Israel-Gaza War
THE ISRAEL GAZA BRIEFINGS

Hamas launches rocket attack towards Tel Aviv

Hours later, an aid agency said dozens of Palestinians had been killed and injured in an Israeli air strike on Rafah.

2 hrs ago | NEWS Middle East

How plans for 'the day after' could help end war in Gaza

This subject will be on the table when EU and Middle Eastern foreign ministers meet.

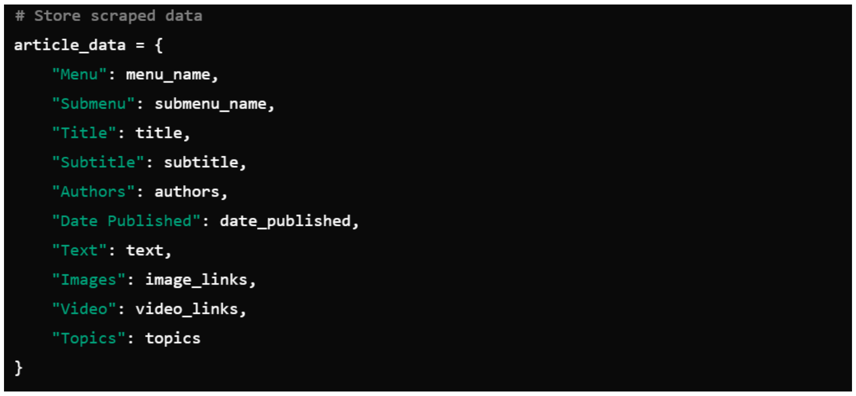
Tool Selection: We opted for Python along with libraries such as Selenium and BeautifulSoup for web scraping. Selenium facilitated dynamic web page interactions, while BeautifulSoup aided in parsing HTML content.

Historical Data Scraping: Initially, we aimed to gather historical data spanning the past week. However, due to resource constraints, we adjusted our approach to focus on scraping the most recent articles. This decision was made to accommodate storage limitations, as we opted for MongoDB's free tier for data storage.

Daily Data Collection: Following the initial scrape, we transitioned to a daily scraping. This was orchestrated using Apache Airflow, a workflow management platform. Airflow allowed us to schedule and automate the scraping process, ensuring a continuous influx of fresh data.

II. Data Storage :

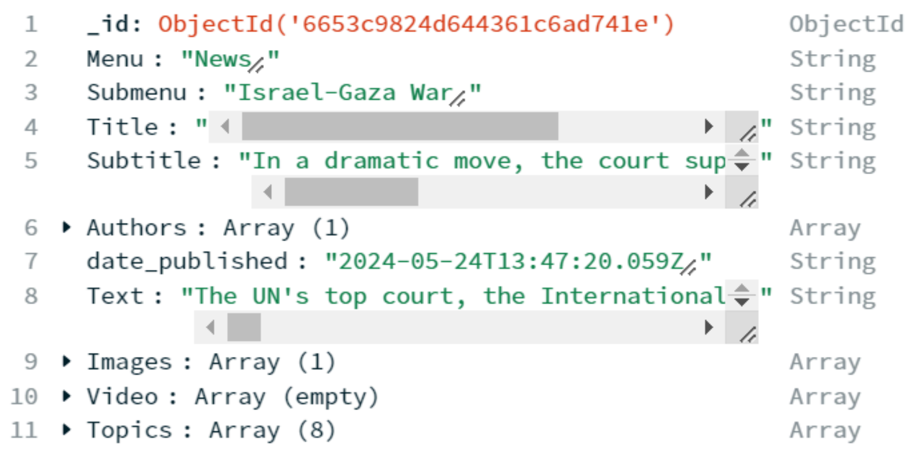
Once the data has been scraped, it needs to be stored efficiently for subsequent analysis and retrieval. In this project, we utilized MongoDB as our database management system due to its flexibility and scalability. Here's how we stored the scraped data:



```
# Store scraped data
article_data = {
    "Menu": menu_name,
    "Submenu": submenu_name,
    "Title": title,
    "Subtitle": subtitle,
    "Authors": authors,
    "Date Published": date_published,
    "Text": text,
    "Images": image_links,
    "Video": video_links,
    "Topics": topics
}
```

In the above code snippet, **article_data** is a dictionary containing the various fields extracted from each scraped article, such as **menu, submenu, title, subtitle, authors, date published, text, images, video links, and topics**.

By storing the data in MongoDB, we can leverage its document-oriented storage model to efficiently query and analyze the articles. This ensures that the data is stored in a structured and accessible manner, laying the groundwork for subsequent data analysis and application development phases. This is the first document from the mongodb database :



```
_id: ObjectId('6653c9824d644361c6ad741e')        ObjectId
Menu : "News"                                    String
Submenu : "Israel-Gaza War"                      String
Title : "                                      " String
Subtitle : "In a dramatic move, the court sup  " String
Authors : Array (1)                              Array
date_published : "2024-05-24T13:47:20.059Z"      String
Text : "The UN's top court, the International" String
Images : Array (1)                               Array
Video : Array (empty)                            Array
Topics : Array (8)                               Array
```

III. Backend Development :

FastAPI Implementation

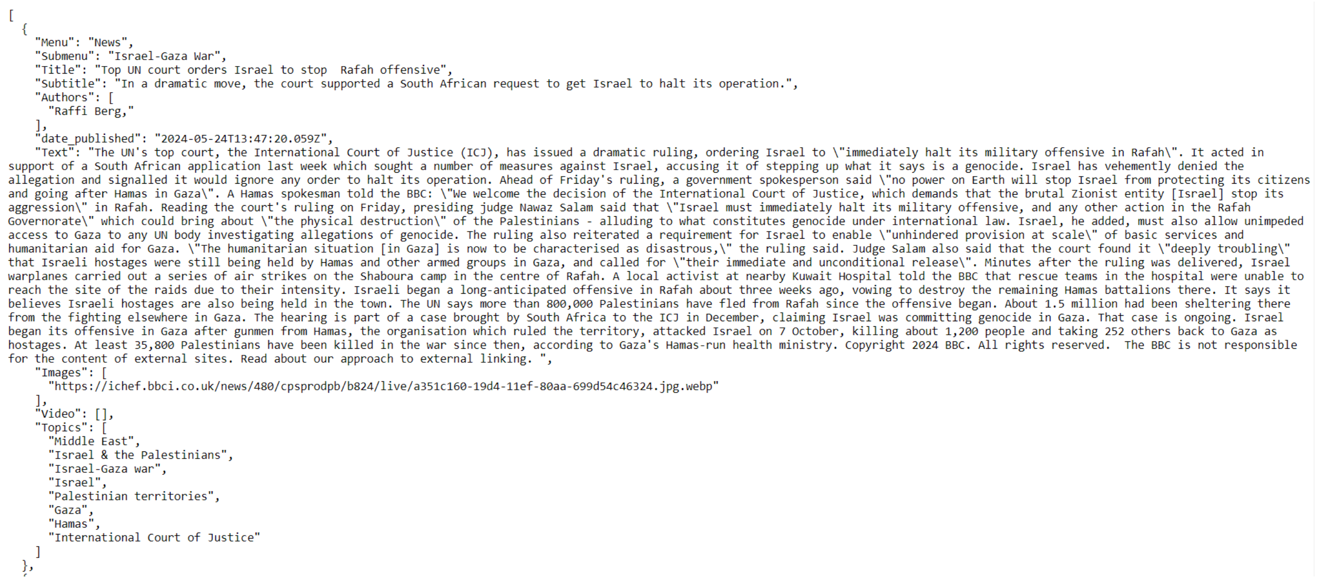
For the backend, FastAPI was chosen due to its high performance and ease of use for creating APIs. FastAPI's automatic data validation and type checking, along with its asynchronous capabilities, made it an ideal choice for this project. FastAPI automatically generates interactive API documentation using Swagger UI and Redoc. This makes it easy to test endpoints and understand the API structure.

API Endpoints

The FastAPI application was designed to serve multiple endpoints to interact with the MongoDB database and provide the required data to the frontend. The following endpoints were created:

- **/articles:** This endpoint returns a list of all articles. It is useful for displaying all the collected data on the frontend. The response includes metadata such as the title, subtitle, text, authors, publication date, and links to images and videos.

http://127.0.0.1:8000/articles :



```
[
  {
    "Menu": "News",
    "Submenu": "Israel-Gaza War",
    "Title": "Top UN court orders Israel to stop  Rafah offensive",
    "Subtitle": "In a dramatic move, the court supported a South African request to get Israel to halt its operation.",
    "Authors": [
      "Raffi Berg,"
    ],
    "date_published": "2024-05-24T13:47:20.059Z",
    "Text": "The UN's top court, the International Court of Justice (ICJ), has issued a dramatic ruling, ordering Israel to \"immediately halt its military offensive in Rafah\". It acted in
support of a South African application last week which sought a number of measures against Israel, accusing it of stepping up what it says is a genocide. Israel has vehemently denied the
allegation and signalled it would ignore any order to halt its operation. Ahead of Friday's ruling, a government spokesperson said \"no power on Earth will stop Israel from protecting its citizens
and going after Hamas in Gaza\". A Hamas spokesman told the BBC: \"We welcome the decision of the International Court of Justice, which demands that the brutal Zionist entity [Israel] stop its
aggression\" in Rafah. Reading the court's ruling on Friday, presiding judge Nawaz Salam said that \"Israel must immediately halt its military offensive, and any other action in the Rafah
Governorate\" which could bring about \"the physical destruction\" of the Palestinians - alluding to what constitutes genocide under international law. Israel, he added, must also allow unimpeded
access to Gaza to any UN body investigating allegations of genocide. The ruling also reiterated a requirement for Israel to enable \"unhindered provision at scale\" of basic services and
humanitarian aid for Gaza. \"The humanitarian situation [in Gaza] is now to be characterised as disastrous,\" the ruling said. Judge Salam also said that the court found it \"deeply troubling\"
that Israeli hostages were still being held by Hamas and other armed groups in Gaza, and called for \"their immediate and unconditional release\". Minutes after the ruling was delivered, Israel
warplanes carried out a series of air strikes on the Shaboura camp in the centre of Rafah. A local activist at nearby Kuwait Hospital told the BBC that rescue teams in the hospital were unable to
reach the site of the raids due to their intensity. Israeli began a long-anticipated offensive in Rafah about three weeks ago, vowing to destroy the remaining Hamas battalions there. It says it
believes Israeli hostages are also being held in the town. The UN says more than 800,000 Palestinians have fled from Rafah since the offensive began. About 1.5 million had been sheltering there
from the fighting elsewhere in Gaza. The hearing is part of a case brought by South Africa to the ICJ in December, claiming Israel was committing genocide in Gaza. That case is ongoing. Israel
began its offensive in Gaza after gunmen from Hamas, the organisation which ruled the territory, attacked Israel on 7 October, killing about 1,200 people and taking 252 others back to Gaza as
hostages. At least 35,800 Palestinians have been killed in the war since then, according to Gaza's Hamas-run health ministry. Copyright 2024 BBC. All rights reserved.  The BBC is not responsible
for the content of external sites. Read about our approach to external linking. ",
    "Images": [
      "https://ichef.bbci.co.uk/news/480/cpsprodpb/b824/live/a351c160-19d4-11ef-80aa-699d54c46324.jpg.webp"
    ],
    "Video": [],
    "Topics": [
      "Middle East",
      "Israel & the Palestinians",
      "Israel-Gaza war",
      "Israel",
      "Palestinian territories",
      "Gaza",
      "Hamas",
      "International Court of Justice"
    ]
  },
```

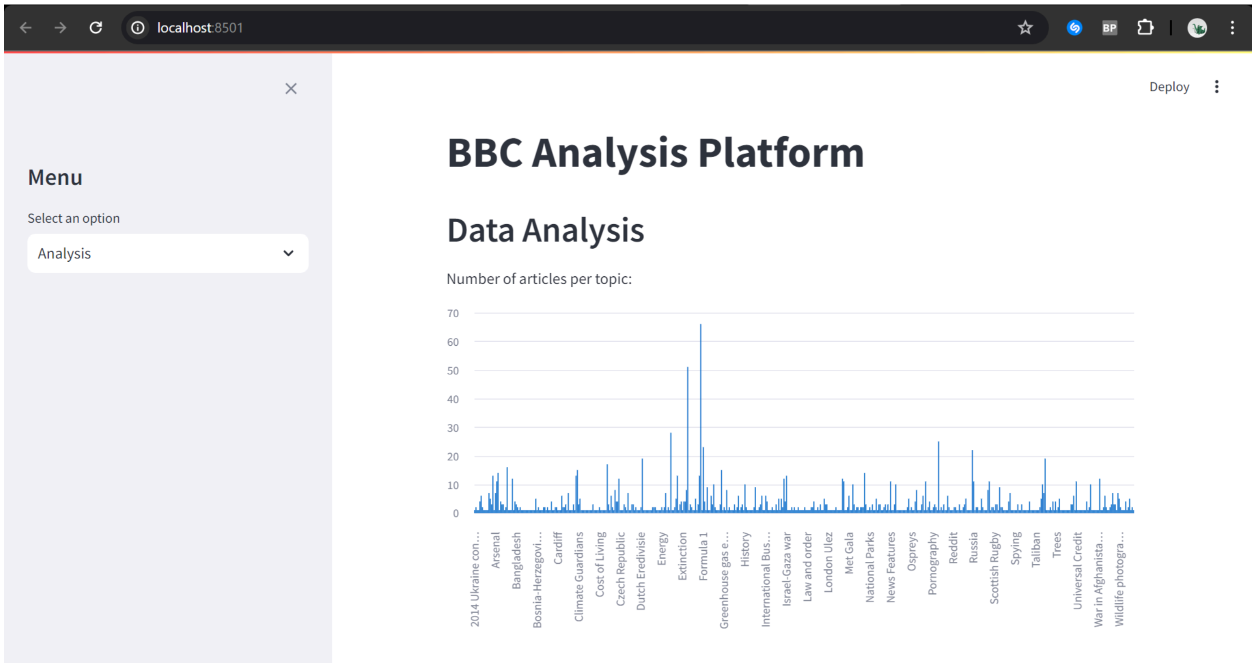
FastAPI's features like automatic documentation, type safety, and asynchronous capabilities made it an excellent choice for this project, providing a seamless and efficient way to serve the data collected from the BBC articles.

IV. Analysis:

The analysis section involves using the FastApi data from the BBC website to identify potential biases, categorize articles, and extract meaningful insights. This section will be divided into several parts: Bias Analysis, Categorization of Articles, classifying the articles into categories of my choice, and Additional Insights. The results will be presented using Streamlit, an interactive web application framework for Python.

Bias Analysis :

To identify any biases in the dataset by examining the distribution of articles across different topics and categories.



Menu

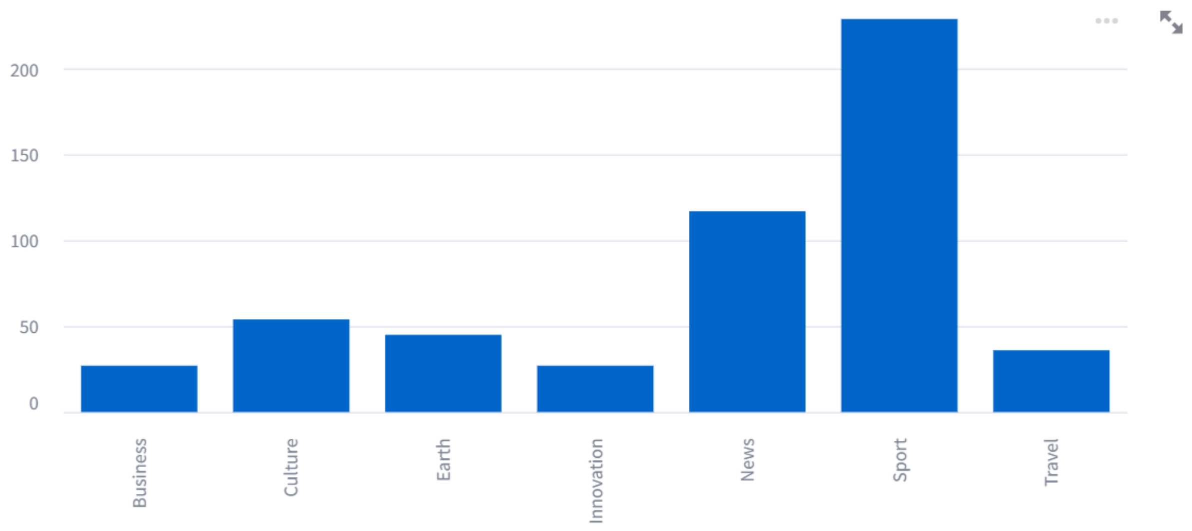
Select an option: Analysis

BBC Analysis Platform

Data Analysis

Number of articles per topic:

Number of articles per menu:



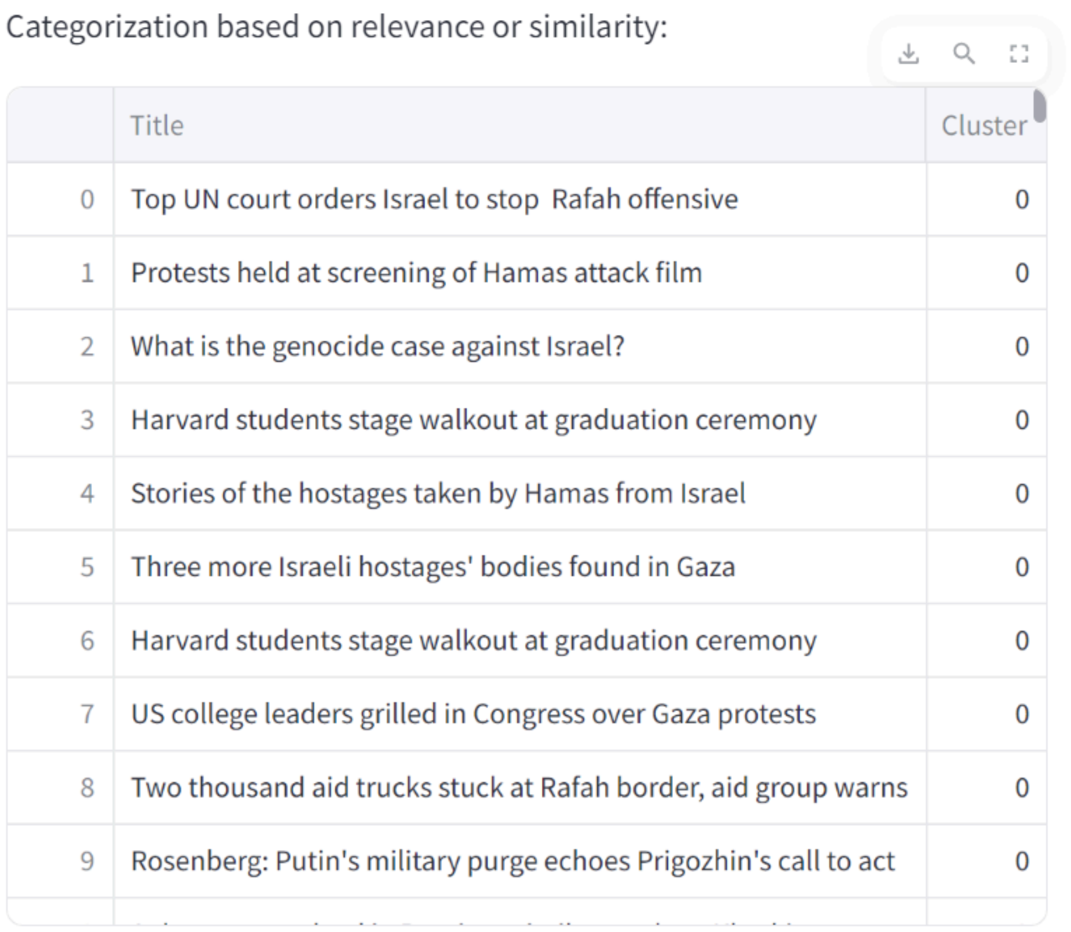
Bias detection:

Most common topic: Football

Least common topic: Religion

Categorization of Articles :

To categorize articles into meaningful groups based on their content.

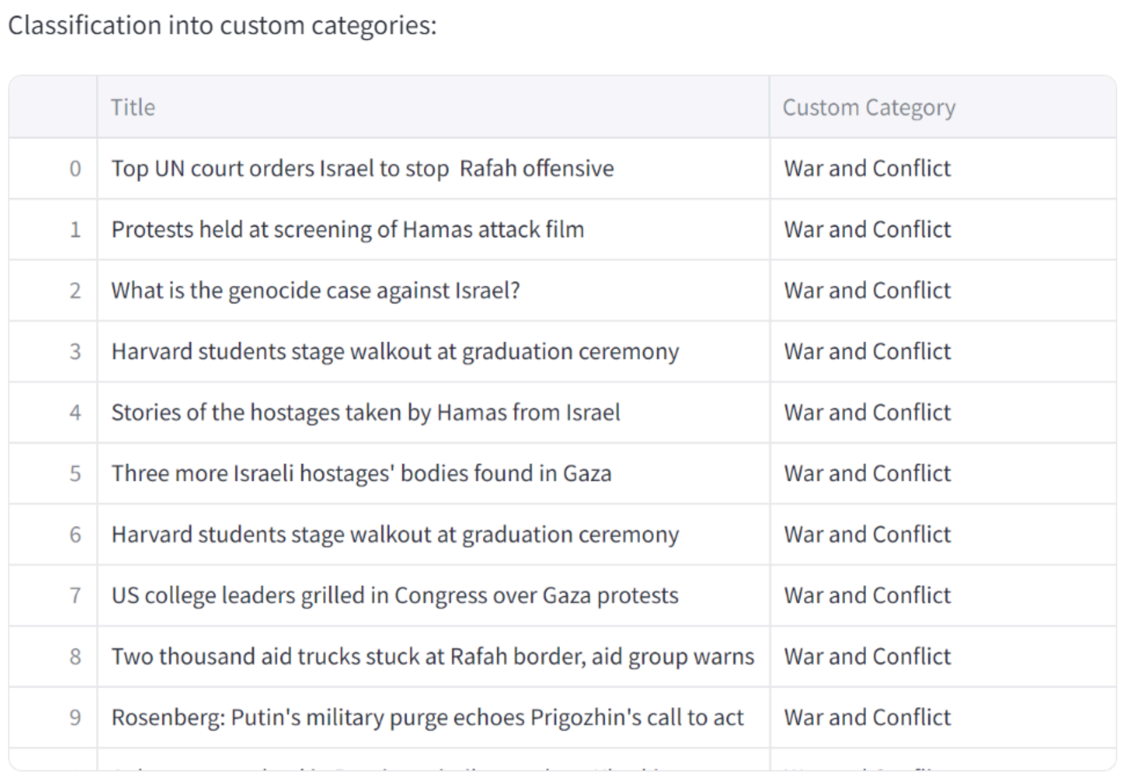


Categorization based on relevance or similarity:

	Title	Cluster
0	Top UN court orders Israel to stop Rafah offensive	0
1	Protests held at screening of Hamas attack film	0
2	What is the genocide case against Israel?	0
3	Harvard students stage walkout at graduation ceremony	0
4	Stories of the hostages taken by Hamas from Israel	0
5	Three more Israeli hostages' bodies found in Gaza	0
6	Harvard students stage walkout at graduation ceremony	0
7	US college leaders grilled in Congress over Gaza protests	0
8	Two thousand aid trucks stuck at Rafah border, aid group warns	0
9	Rosenberg: Putin's military purge echoes Prigozhin's call to act	0

Classifying the articles into categories of my choice :

To classify articles into custom categories of your choice based on their content.


Classification into custom categories:

	Title	Custom Category
0	Top UN court orders Israel to stop Rafah offensive	War and Conflict
1	Protests held at screening of Hamas attack film	War and Conflict
2	What is the genocide case against Israel?	War and Conflict
3	Harvard students stage walkout at graduation ceremony	War and Conflict
4	Stories of the hostages taken by Hamas from Israel	War and Conflict
5	Three more Israeli hostages' bodies found in Gaza	War and Conflict
6	Harvard students stage walkout at graduation ceremony	War and Conflict
7	US college leaders grilled in Congress over Gaza protests	War and Conflict
8	Two thousand aid trucks stuck at Rafah border, aid group warns	War and Conflict
9	Rosenberg: Putin's military purge echoes Prigozhin's call to act	War and Conflict

The analysis conducted provides insights into the distribution of articles, potential biases, and categorization based on content. The use of Streamlit for visualization allows for an interactive exploration of the data, making it easier to identify patterns and trends. The categorization techniques, including topic modeling, clustering, classification, and custom categorization, help in organizing the articles into meaningful groups, providing a deeper understanding of the dataset.

Bonus 1: Docker Images

In the context of this project, Docker is used to containerize an Airflow setup. Airflow is a platform to programmatically author, schedule, and monitor workflows. By using Docker, the entire Airflow environment, including its dependencies and configurations, is encapsulated in a container, ensuring consistency and ease of deployment across different environments.

- Setup Airflow DAG for Scraping:

    - Define a Directed Acyclic Graph (DAG) in Airflow that schedules and runs the scraping tasks daily. This DAG specifies the sequence of operations to scrape data from the BBC website and store it in MongoDB.
- Automated Scheduling:
    - Airflow's scheduler runs the scraping task at the defined intervals (e.g., daily), ensuring that the data is collected consistently without manual intervention.

This approach leverages Docker's containerization to create an isolated, reproducible environment for Airflow, simplifying deployment and scaling.

Bonus: Explain how you can create a Neo4j DB that is scalable using Kubeflow and Kubernetes :

The Neo4j database, a Kubernetes cluster, was set up using Helm charts to deploy and manage Neo4j. Kubeflow was employed to orchestrate machine learning workflows, leveraging Kubernetes for resource management and scalability. This setup ensures that the Neo4j database can handle large volumes of interconnected data efficiently, with the ability to scale dynamically based on demand.

To create a scalable Neo4j database using Kubeflow and Kubernetes, follow these steps:

1. **Kubernetes Cluster Setup:**
    - Start by setting up a Kubernetes cluster using a cloud provider such as Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS), or Azure Kubernetes Service (AKS). Kubernetes provides the infrastructure for automating deployment, scaling, and operations of application containers.
2. **Deploy Neo4j on Kubernetes:**
    - Use Helm, a Kubernetes package manager, to deploy Neo4j. Helm simplifies the deployment process by using predefined templates and configurations to set up the Neo4j cluster.

Helm Chart for Neo4j:

- A Helm chart specifies the configuration for deploying Neo4j, including the number of core servers and read replicas, resource requests and limits, and persistent storage configurations.
- Deploying Neo4j using Helm ensures that the database can be easily scaled and managed within the Kubernetes cluster.

3. **Kubeflow for Machine Learning Workflows:**
    - Install Kubeflow on the Kubernetes cluster to manage machine learning workflows. Kubeflow provides tools for developing, orchestrating, deploying, and running scalable and portable ML workloads.

Kubeflow Pipeline:

- Define a pipeline in Kubeflow to automate data processing and analysis tasks. A pipeline consists of a series of steps, each represented by a containerized component, which can be executed in a sequence or in parallel.
- For example, a pipeline can be created to extract data from Neo4j, process it, and run machine learning models on the processed data.

4. **Scalability Considerations:**
    - Use Kubernetes to manage and scale Neo4j instances based on the workload. Kubernetes handles the orchestration of containerized applications, ensuring that Neo4j can scale up or down as needed.
    - Implement load balancing to distribute traffic across multiple Neo4j instances, ensuring high availability and reliability.
    - Use persistent storage solutions provided by Kubernetes to ensure data durability and resilience.

Conclusion :

This report outlines the steps taken to scrape data from the BBC website using Airflow daily, store the data in MongoDB, and present the data using a FastAPI backend and Streamlit frontend. Additionally, it includes the analysis of the dataset to identify potential biases and categorize articles. The document also discusses the implementation of best practices and bonus tasks, such as setting up a scalable Neo4j database using Kubeflow and Kubernetes and creating and using Docker images.