Raquel Benjumea(251235)
Clara Jaurés(254177)
Naia Gorostiaga(253149)

Part 4:User Interface and Web Analytics

GitHub: https://github.com/Clarajaures/IRWA_2024_.git
(we have changed the repo due to a conflict of merges)

The objective of this lab is different compared to the previous labs. This lab has two parts,the first one is focused on the user interface where the goal is to create a web server. The other part is doing web analytics where the goal is understanding how the users use the web server and the search engine.

1. User Interface

To start with this part of the project we have used the template of Skeleton Project Flask provided in the Aula Global.

This template featured a search engine that allowed users to input queries. Upon submission, the app displayed a results page with all relevant documents, in our case tweets, ranked by relevance. Users could click on a document to view additional details. Additionally, selecting a document not only revealed its specific information but also provided access to app-wide statistics. These included data such as the most frequently searched queries and the number of clicks on individual documents. The app also featured a dashboard page that presented these statistics visually using graphs for improved user understanding

After downloading the template, we customized the provided code to align with our project goals. We started by loading the dataset needed, *farmers-protest-tweets.json*.

Then we have implemented the functions we used in previous labs to create the user interface. We have started by cleaning the json and creating a dictionary as we did in the previous labs.
We have used the functions of *preprocess_text*, *extract_tweets* from the previous labs and we applied some modifications of the functions given by the template.

When we completed the modifications mentioned, we modified how to search queries. After the user gave it a query we implemented a tf idf search. We have used this way to search queries due to it being the way of searching queries we have used the most and we felt more comfortable using this searching.
For searching the query we used *create_index_tfidf*, *search_tf_idf* and *rank_documents*, all these functions we have already used previously. When we had the documents with bigger relevance we converted them into ResultItem due to it being easier to display the documents using this object in HTML.

To monitor and analyze user interactions, we created a separate JSON file named *data_collection.json*. This file stores user behavior data in four dictionaries:

- fact_clicks: Tracks document IDs and the number of times each document was clicked.
- fact_two: Records user IP addresses and the frequency of their visits.
- fact_three: Logs detailed user information, including platform and browser version, whether the user is a bot or a real user among other information.

- fact_query_terms: Tracks the terms entered by users in their queries and their frequency of use.

To present each page and its associated information effectively, we customized the HTML files provided in the template.

2. Web Analysis

In this section, we will focus on the web analytics part of the project, where we explore how we tracked and analyzed user interactions with the web server. We used various methods to capture meaningful data, which we can later visualize on the dashboard.

Once the data was collected and stored (with the *data_collection.json* we mentioned in the last part), we turned our attention to the dashboard, which visually presents this data to help us understand user behavior.

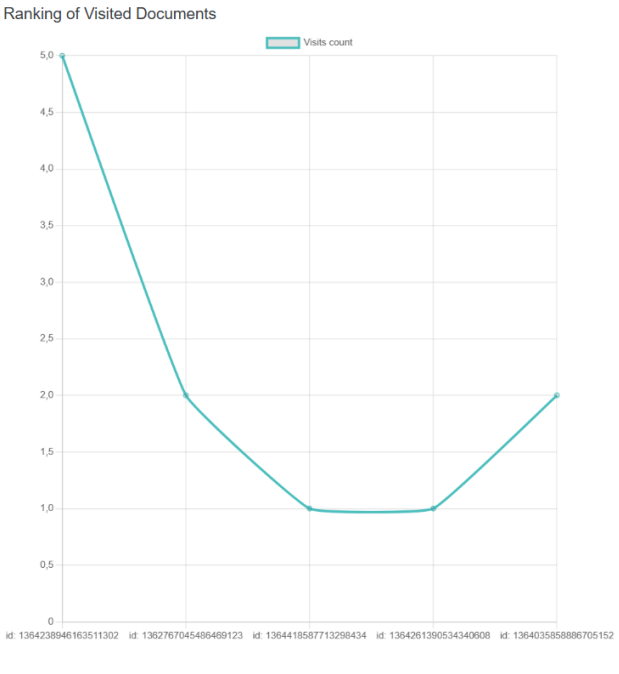
Dashboard Visualization

The data collected through these dictionaries was then used to create an informative dashboard that visually presents insights about user interactions. The dashboard plays a key role in helping us understand patterns and behaviors, guiding future decisions for improving the user experience.

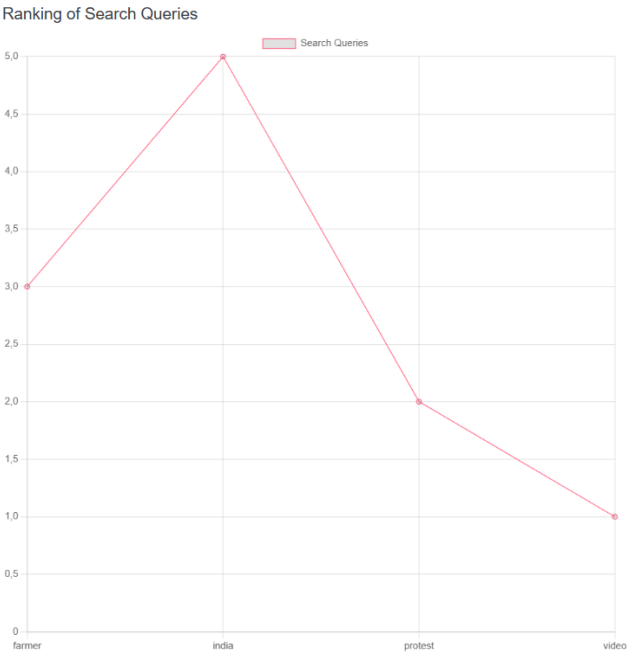
For the implementation of dashboards in the file (myapp/templates/dashboards.html), we also followed the pattern of the example dashboard, changing the values that we wanted it to return since not every dashboard provided the same information or contained the same variables as we wanted to inform about different aspects. We also modified the color of each graph for a more visual distinction (aside from the different shape in general) so that at first glance, each dashboard looked different, and the user wouldn't confuse one dashboard with another. With these modifications to these files, we achieved the graphs shown above

The dashboard includes several key sections:

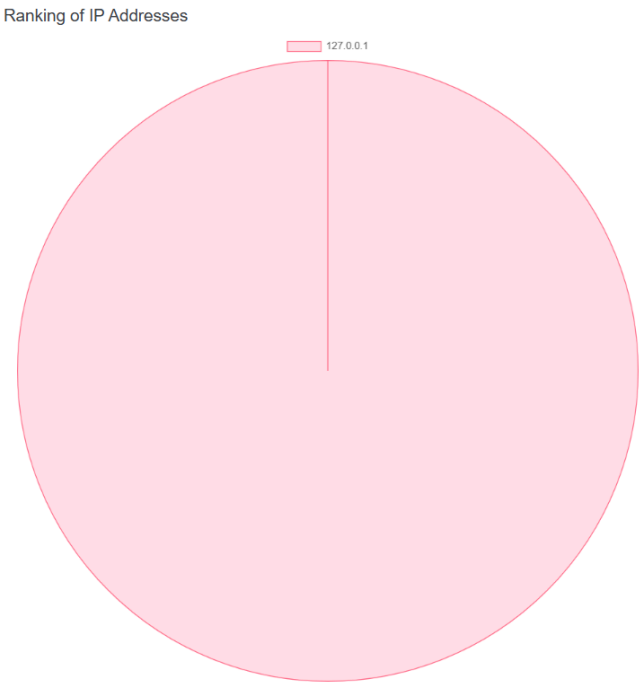
Document Clicks: We created a graph to show how many times each tweet has been opened. This graph helps us identify which documents are most frequently accessed by users. Alongside the graph, we also display additional information, such as the number of visits, tweet ID, and tweet text.



Search Queries: A separate graph is used to display the frequency of different search queries. This visualization helps us identify which topics are most frequently searched, offering valuable insights into what users are looking for on the platform. The data is displayed both graphically and in textual form for a comprehensive overview.



IP Address and Geographical Distribution: While the application is being tested locally, all requests share the same IP address. However, in a production environment, IP data can provide useful insights into the geographical locations of users. We track the number of visits per unique IP address to analyze the global reach of the platform.



User-Agent Information: The dashboard also provides a breakdown of the types of devices and browsers used to access the platform. This includes information about whether users are on Windows or macOS, and which browsers and versions they are using. This data allows us to optimize the platform's functionality and appearance for the most commonly used devices.