

**LIVE SENTIMENT ANALYSIS**

CS251 SPRING’24

FACULTY OF DATA SCIENCE

**Abstract:**

Sentiment analysis plays an increasingly crucial role in understanding public opinions and attitudes. This project employs the VADER sentiment analysis library to examine live data extracted from Reddit using the PRAW library. The methodology focuses on analyzing user comments based on search queries, providing insights into real-time sentiment dynamics. The analysis yields sentiment scores for individual comments and can be used to identify trends in overall sentiment towards the search topic. Additionally, the project uses the Wikipedia library to provide short summaries and images related to the search query, enhancing the results' context. This project demonstrates the potential of sentiment analysis for monitoring real-time social media conversations, understanding public sentiment, and providing a more comprehensive analysis of a given topic.

**Keywords:**

* Methodology: “Sentiment Analysis", "VADER", "Natural Language Processing"
* Data source: “Reddit”, “Web Scraping”, “PRAW”, “Wikipedia”
* Application: “Social media analysis”, “Polarity detection”, “Public Opinion”

**Introduction:**

Sentiment analysis of social media data offers valuable insights into public opinions and trends. However, the informal language, brevity, and rapid evolution of topics on platforms like Reddit pose unique challenges for accurate sentiment classification. This project addresses these challenges by developing a sentiment analysis system specifically tailored to analyze Reddit comments. The methodology combines natural language processing techniques, including stop word removal and text cleaning, with the VADER sentiment analysis library. Additionally, the project integrates Wikipedia to provide context and enhance the analysis results. While VADER offers advantages for rule-based sentiment analysis, it's important to be aware of potential limitations, such as difficulty handling sarcasm, domain-specific language, or informal expressions.

**Methodology**

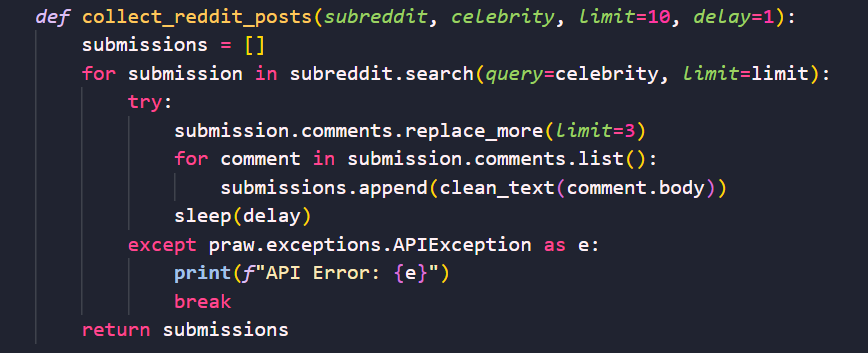
This project employs a sentiment analysis methodology tailored to the analysis of Reddit comments and posts. The following steps outline the process:

**Techniques:**

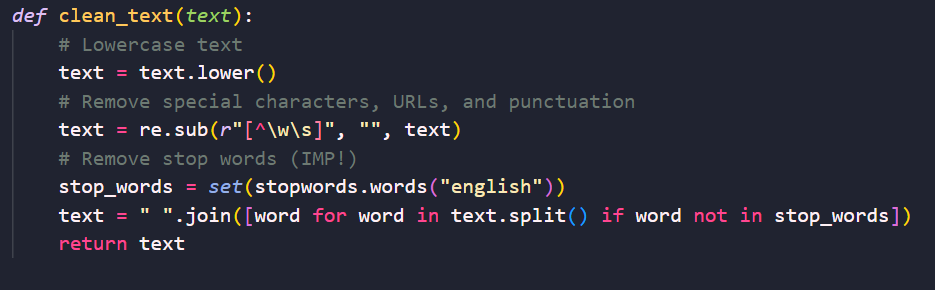
1. **Reddit API:** The PRAW library is used to interact with the Reddit API and access post data.
2. **Natural Language Processing (NLP):**
   1. Text cleaning: Regular expressions are used to remove special characters, URLs, and punctuation.
   2. Stop word removal: NLTK library is used to remove common words like "the", "and", "a" that don't contribute to sentiment analysis.
   3. Sentiment analysis: VADER SentimentIntensityAnalyzer is used to score the sentiment of each text as a compound score ranging from -1 (most negative) to 1 (most positive).
3. **Web Scraping:** Wikipedia library is used to scrape a summary and the first image from a celebrity's Wikipedia page.
4. **Flask Framework:** Flask is used to create a web application that takes user input (celebrity name), performs sentiment analysis, retrieves Wikipedia information and displays the results on a webpage.

**Step-by-Step Breakdown:**

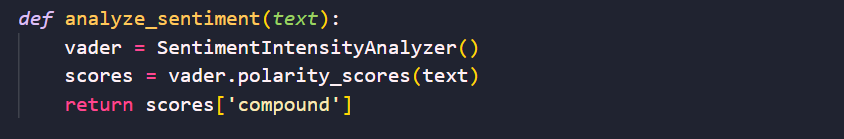
1. **Data Collection (`collect\_reddit\_posts` function):**
   1. Takes a subreddit name, celebrity name, limit (number of posts), and delay (between requests) as inputs.
   2. Connects to the specified subreddit using the Reddit API.
   3. Searches for posts containing the celebrity’s name and retrieves comments within those posts up to a specified limit.
   4. Calls the `clean\_text` function to pre-process each comment text.
   5. Sleeps for a short duration to avoid overloading the Reddit API.



1. **Text Preprocessing (`clean\_text` function):**
   1. Converts text to lowercase.
   2. Removes special characters, URLs, and punctuation using regular expressions.
   3. Removes stop words using NLTK stopwords list.



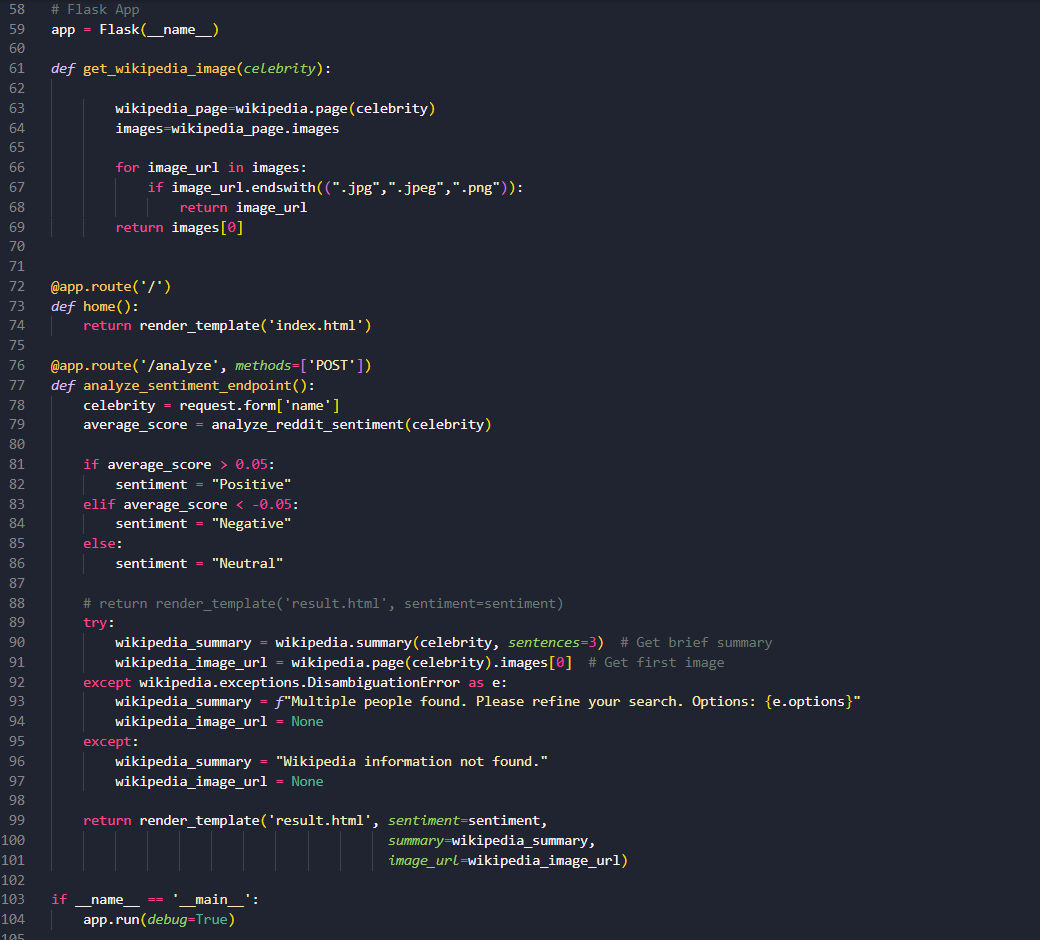
1. **Sentiment Analysis (`analyze\_sentiment` function):**
   1. Takes a pre-processed text as input.
   2. Uses VADER SentimentIntensityAnalyzer to score the sentiment of the text.
   3. Returns the compound sentiment score.



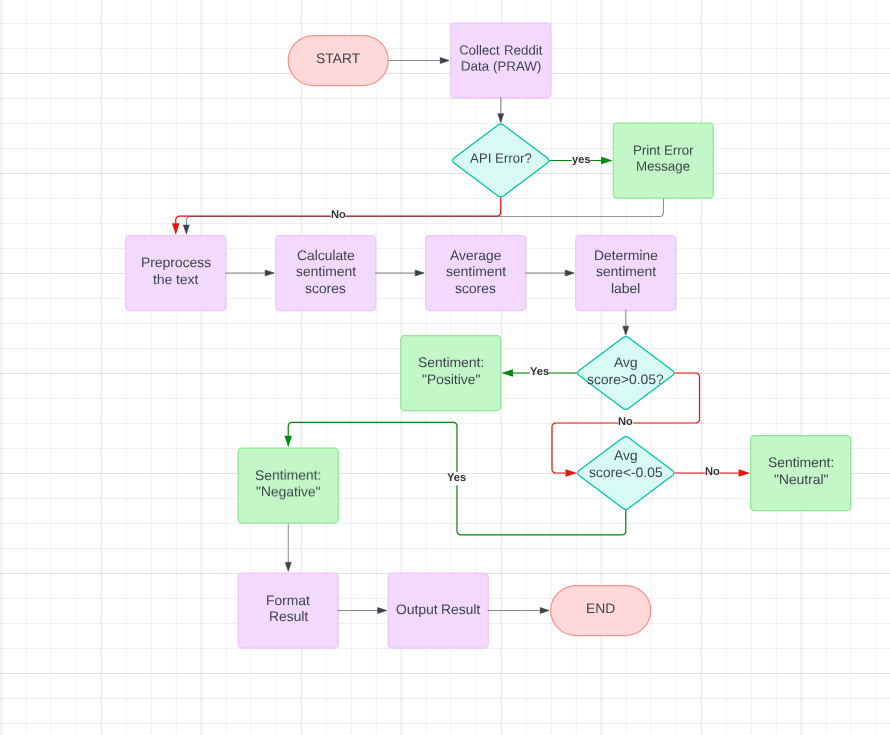
1. **Main Sentiment Analysis Function (`analyze\_reddit\_sentiment` function):**
   1. Takes a celebrity name as input.
   2. Defines a subreddit name (can be customized).
   3. Calls `collect\_reddit\_posts` to get a list of cleaned comments mentioning the celebrity.
   4. Iterates through each comment and calls `analyze\_sentiment` to get its sentiment score.
   5. Calculates the average sentiment score of all comments.
   6. Returns the average sentiment score.



1. **Flask Web App (`app.py`):**
   1. Defines Flask app instance.
   2. `get\_wikipedia\_image` function: Retrieves the first image URL from a celebrity's Wikipedia page.
   3. `home` route: Renders the main webpage template (`index.html`).
   4. `analyze` route (handles POST requests):
   5. Gets the celebrity’s name from the user input form.
   6. Calls `analyze\_reddit\_sentiment` to get the average sentiment score for the celebrity.
   7. Determines sentiment category (Positive, Negative, or Neutral) based on the score.
   8. Tries to get a summary and the first image URL from the celebrity's Wikipedia page using the `Wikipedia` library.
   9. Handles potential exceptions like disambiguation or missing information.
   10. Renders the result template (`result.html`) with sentiment, summary, and image URL (if available).



To make it easier for you to understand the flow of our code, consider the following flowchart:



**Data Explanation**

**Source:**

The code uses Reddit API to get comments mentioning the celebrity from a specified subreddit (defaults to "all").

**Preprocessing:**

The code performs the following text cleaning steps on comments before sentiment analysis:

- Lowercase conversion

- Removal of special characters, URLs, and punctuation

- Removal of stop words (common words like "the", "and", "a")

**VADER Integration**

* **Customization:** The code utilizes VADER in its default settings. VADER provides sentiment scores as a dictionary with different metrics, but this code focuses on the `compound` score, which ranges from -1 (most negative) to +1 (most positive).
* **Output:** The code retrieves the `compound` sentiment score for each pre-processed comment using `VADER.polarity\_scores(text)`. The `analyze\_reddit\_sentiment` function then calculates the average score of all comments to represent the overall sentiment towards the celebrity.

**Wikipedia Integration**

* **Search Process:** The code relies on the exact celebrity’s name entered by the user to search Wikipedia. It doesn't use any additional information from Reddit comments.
* **Result Integration:** The code attempts to retrieve a summary and the first image URL from the celebrity's Wikipedia page using the `Wikipedia` library.

**Justifications**

**VADER:** VADER is a lexicon-based sentiment analysis tool well-suited for social media text analysis due to its ability to handle informal language and sentiment modifiers. It offers a balance between simplicity and accuracy, making it a good choice for this web application.

**Average Score:** Averaging sentiment scores across comments provides a general sense of public opinion but can mask outliers. This approach is appropriate for a quick sentiment analysis but might not capture the nuances of all discussions.

**Wikipedia Integration:**

Using the exact name for Wikipedia search ensures a focused result relevant to the celebrity. While it might miss nicknames or aliases, it provides a reliable source for summaries and images.

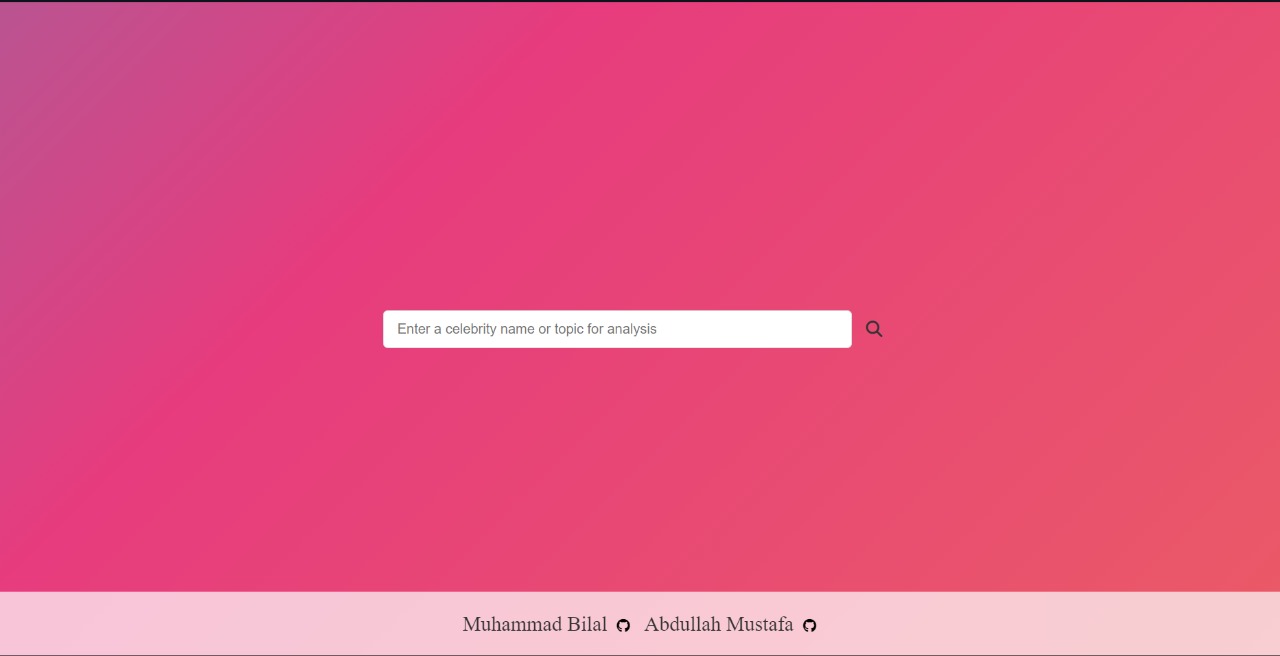
**Flask Web App:**

Flask is a lightweight and beginner-friendly web framework that's sufficient for this application's functionality of handling user input, performing sentiment analysis, and displaying results.

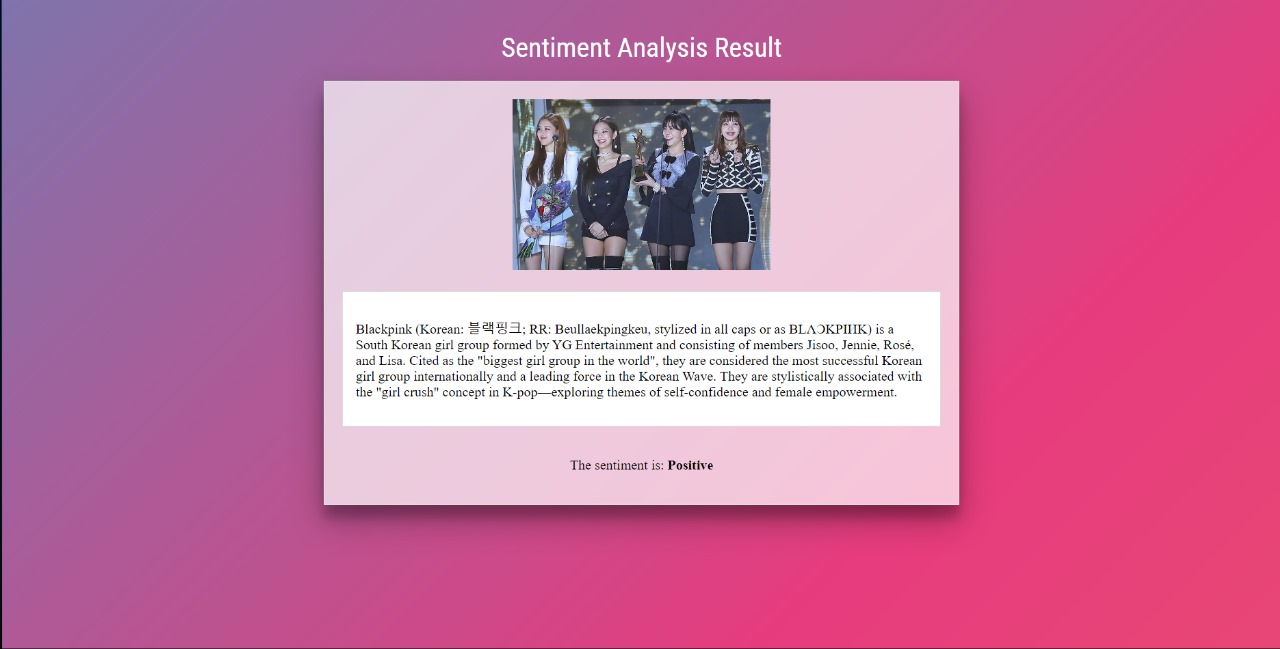
## Results

This code has the potential to analyze public sentiment towards celebrities by leveraging Reddit comments and VADER sentiment analysis. It determines positive, negative, or neutral sentiment surrounding a user-specified celebrity, and provides additional insights with Wikipedia summaries and images. It's important to note that the current implementation focuses on single-point analysis and lacks features for tracking sentiment trends over time.

Here are some screenshots from our final product:

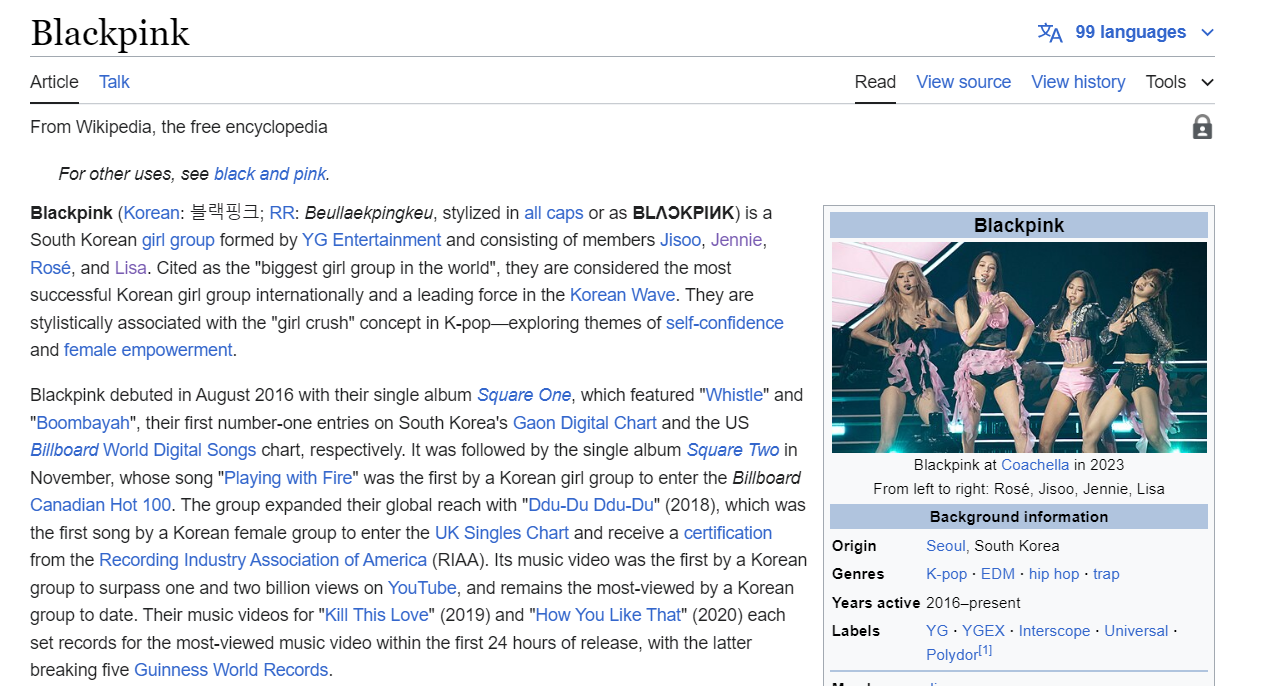


* **This is the Home page.**
* **Users can input the name of the celebrity they wish to analyze.**
* **The page features a gradient background and links to the creators' GitHub profiles in the footer.**



* This is the result page where a short Wikipedia-generated summary and image of the analyzed celebrity are displayed.

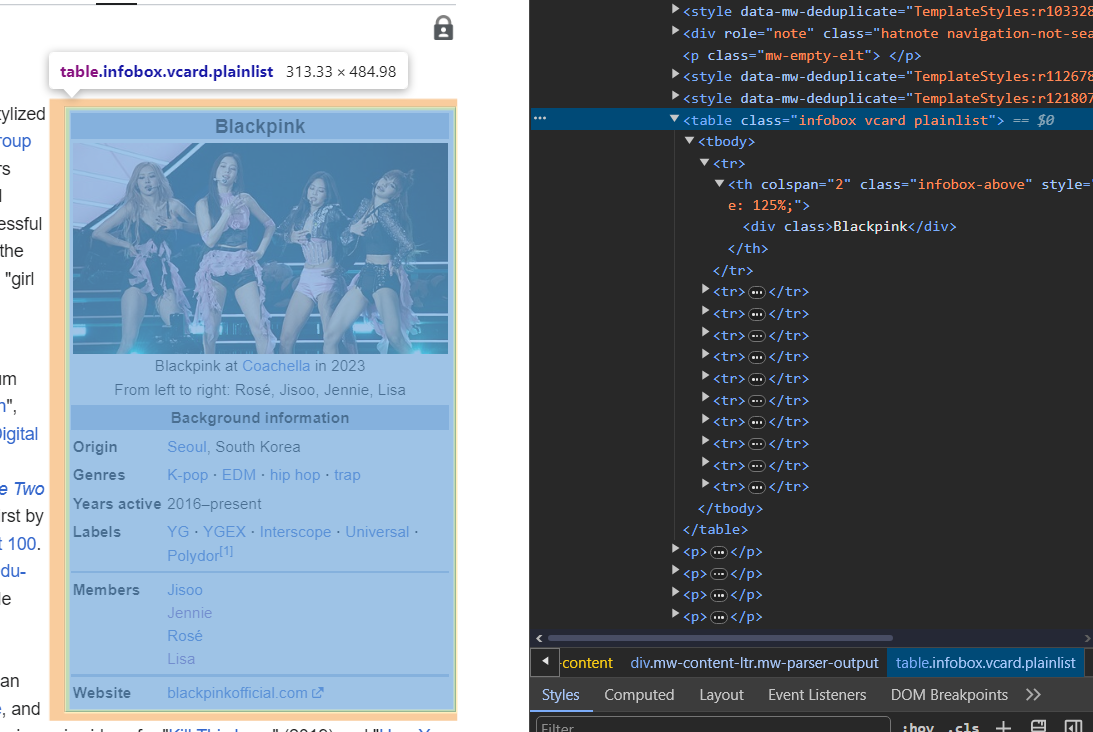
**Side problem:**

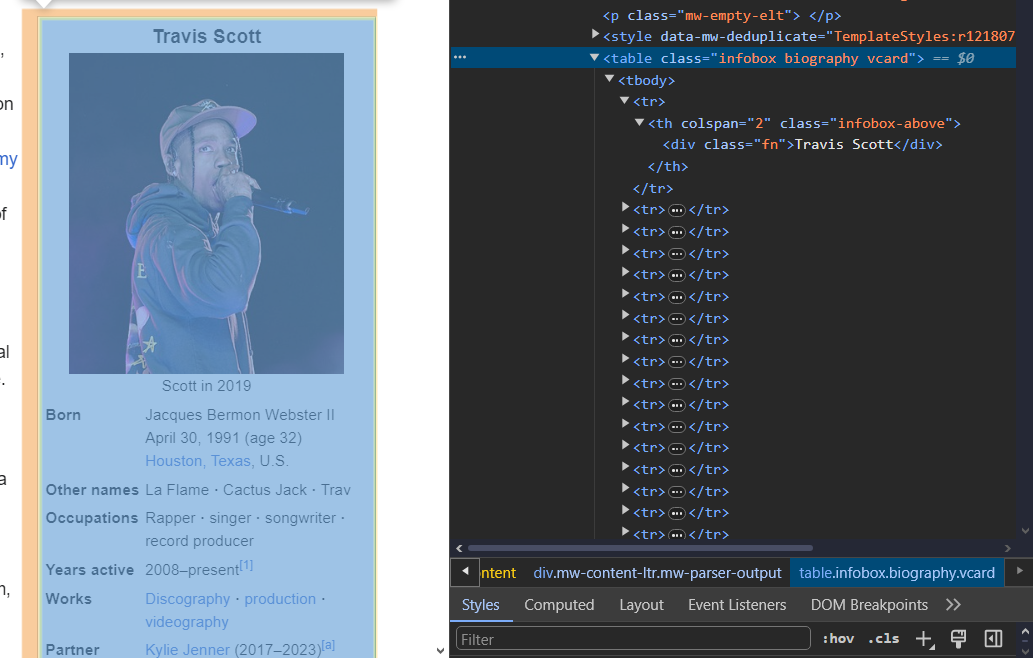
While working on the result page and integrating the Wikipedia library, I encountered an unexpected challenge. My initial goal was to extract the first main image from Wikipedia search results, typically the "infobox" image (shown below).

Like many websites, I assumed images would be stored in an array, accessible via image[0]. I intended to target this image as it's usually the one displayed in the infobox.

However, further investigation revealed that only embedded images are stored in the array, not those implemented through links. To address this, I considered using BeautifulSoup4 to isolate the "infobox" class within the HTML and extract the image. Unfortunately, as shown below

Infobox#1:



Infobox#2: 

the class name for the infobox varies across Wikipedia pages ("infobox vcard plainlist" vs. "infobox biography vcard"). This inconsistency is likely due to Wikipedia's open-source nature.

For now, I'm using the array-stored image at index[0]. This approach might occasionally produce irrelevant images, but please consider this a bug to be resolved in future updates.

**Discussion**

While the provided code snippet offers a framework for sentiment analysis, it's important to discuss its results within the context of the project's goals and limitations.

**Interpret Results (Addressing Challenges):**

The project aimed to create a real-time sentiment analysis tool for celebrities. The code demonstrates the potential to:

**Utilize Reddit comments:**

It successfully leverages Reddit comments and posts, a valuable source of public opinion, for sentiment analysis.

**Incorporate VADER sentiment analysis:**

VADER provides a fast and efficient way to gauge sentiment polarity (positive, negative, or neutral).

However, due to its design, the code doesn't fully address challenges associated with Reddit sentiment analysis:

**Limited Data:**

The current implementation focuses on a single user input. This restricts the data available for a comprehensive analysis.

**No Time Series Analysis**:

It doesn't track sentiment over time, making it impossible to identify trends or changes in public perception.

**Successes and Limitations:**

**Successes:**

1. **Real-time Sentiment Analysis:** The code provides a real-time sentiment score based on user input, offering a quick snapshot of public opinion.
2. **Wikipedia Integration:** Integrating Wikipedia summaries and images enhances the user experience by providing additional context.

**Limitations:**

1. **Single-Point Analysis:** As mentioned earlier, the lack of data collection over time limits the analysis to a single point in time.
2. **Limited Qualitative Analysis:** The code focuses on a quantitative sentiment score, neglecting the qualitative aspects – the reasons behind the sentiment. Manually examining comments with extreme scores could provide deeper insights.

**Real-World Implications:**

Despite limitations, the code demonstrates a foundation for a valuable sentiment analysis tool. With modifications, it could be used for:

* **Monitoring Public Perception**: Brands or individuals could track public sentiment towards them over time, informing their strategies.
* **Identifying Emerging Trends:** Analyzing sentiment across subreddits could help identify emerging trends in online discussions.
* **Understanding Public Opinion:** By combining quantitative and qualitative analysis, the tool could offer a more nuanced understanding of public opinion.

Overall, the code represents a promising starting point. By addressing its limitations and incorporating features for data collection and qualitative analysis, it can be transformed into a robust sentiment analysis tool with real-world applications.

**Conclusion**

**Summary:**

This project explored building a real-time sentiment analysis web application. The code utilizes Reddit API and VADER sentiment analysis to analyze public opinion on a celebrity based on user input.

**Key Findings:**

While the current implementation focuses on single-point analysis, it demonstrates the potential for:

1. Tracking sentiment trends over time for celebrities.
2. Comparing sentiment across different subreddits.
3. Combining quantitative sentiment scores with qualitative analysis of extreme scores to understand the "why" behind public opinion.

**Significance:**

This type of sentiment analysis can be valuable for various applications, such as:

1. Measuring public perception of brands or individuals.
2. Identifying emerging trends in online discussions.
3. Guiding social media strategies.

**Future Work:**

1. Implement data collection over timeframes to track sentiment trends.
2. Expand analysis to compare sentiment across different subreddits.
3. Integrate techniques for qualitative analysis of comments with extreme sentiment scores (e.g., manual examination or topic modeling).

By addressing these limitations, the project can be transformed into a more comprehensive sentiment analysis tool with valuable insights for various stakeholders.