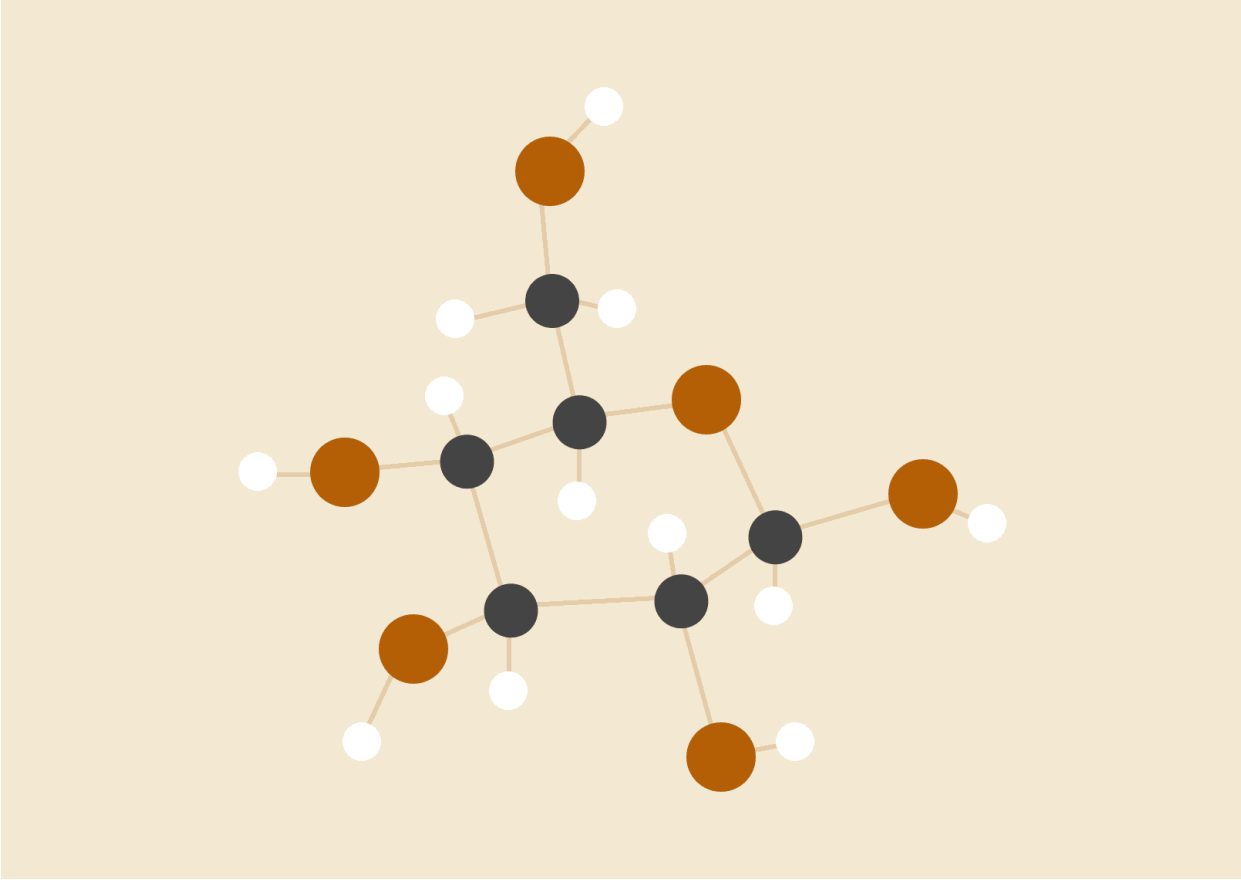
Sentiment Analysis

Sam Malik, Sulayman Ahmed, Hasan Ahmed



Dec 12th, 2023

SI 206

Goals for The Project

The primary goal of our project was to gain insights into the general public opinion on various topics by leveraging the vast user-generated content available on major social media platforms. To achieve this, we planned to work with three prominent platforms: Reddit, Twitter, and YouTube, utilizing their respective APIs to pull relevant data. The focus was on extracting a rich dataset that included popularity metrics, such as upvotes and likes, as well as timestamps of posts and tweets. This data harvesting was crucial to understanding how certain topics trended and resonated with users over time. Additionally, our project aimed to meticulously visualize the sentiment analysis results and the gathered popularity metrics using Matplotlib, a powerful plotting library in Python. The visualization aspect was intended to provide clear, insightful, and accessible representations of our findings, thereby making the analysis of public sentiment on various topics both comprehensive and intuitive.

Goals That Were Achieved

In the project, we successfully achieved key goals that involved working with specific APIs from Reddit, Twitter, and YouTube. Utilizing Reddit's PRAW (Python Reddit API Wrapper), Twitter's Tweepy and Twitter API v2, and the Google API Python Client for YouTube, we were able to systematically gather a diverse array of data from these platforms. This data encompassed not only the textual content of posts and tweets but also included essential popularity metrics like upvotes, likes, and timestamps. Such information enabled us to analyze trends and sentiments associated with various topics as they unfolded on these social media platforms. Moreover, through the integration of sentiment analysis techniques, we could interpret and quantify the general public sentiment expressed in these posts and comments. The project effectively gathered and processed data from these platforms, providing a comprehensive view of public opinion and trends across different social media landscapes.

The Problems We Faced

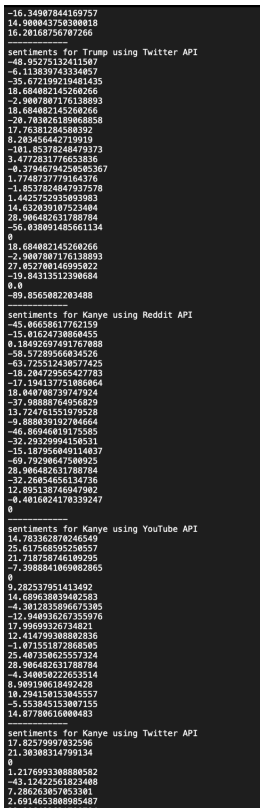
A large problem we faced was with API access. Since our project focuses on social media and there's a lot of controversy about privacy in this field right now, it was difficult to get API access for some platforms. Additionally, we had to deal with rate limiting for many APIs due to being on the lowest tier.

Another issue was authorization. We couldn't find enough API's that gave us the data information we needed so we had to pay for authorization in bigger APIs like twitter to access enough information to get the 3 APIs working in the same way.

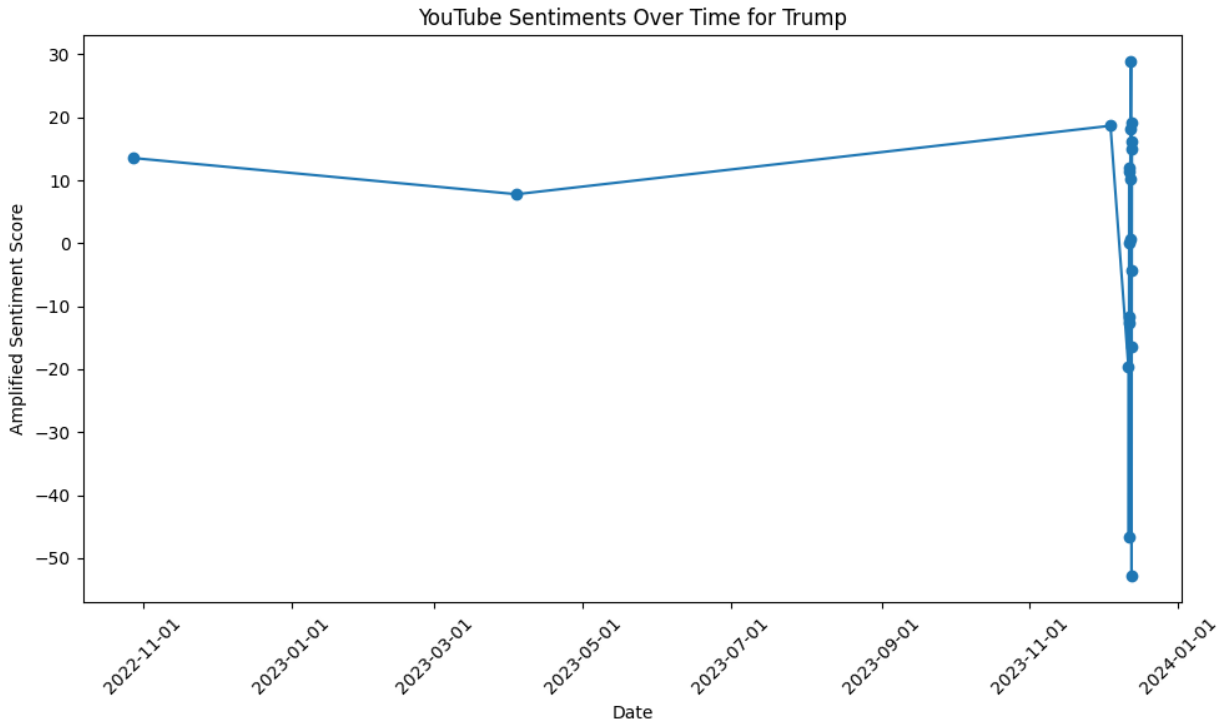
A lot of the data we collected had locked or removed comments sections. We realized this was adding "zero" data that was causing errors in our calculations so we adjusted this by skipping over these data points.

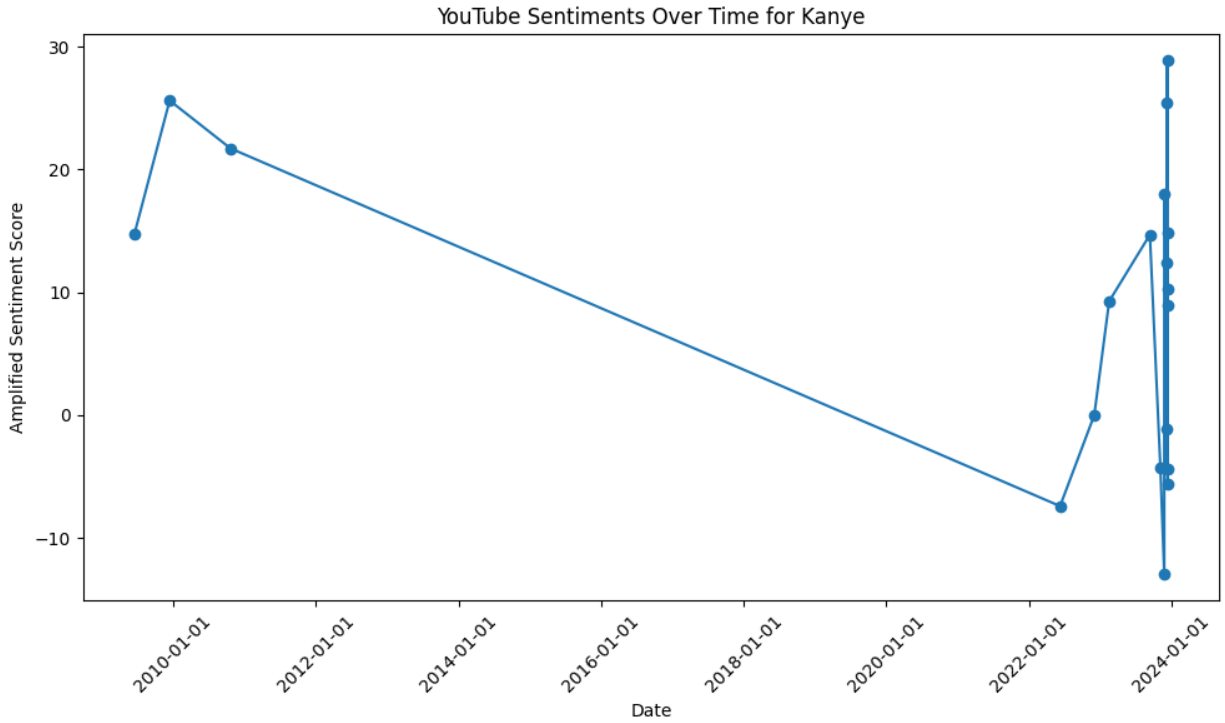
Some of the posts also weren't available in some countries or had safety warnings on twitter so they were locked to a viewer or viewers which we tried to account for in the sentiment analysis.

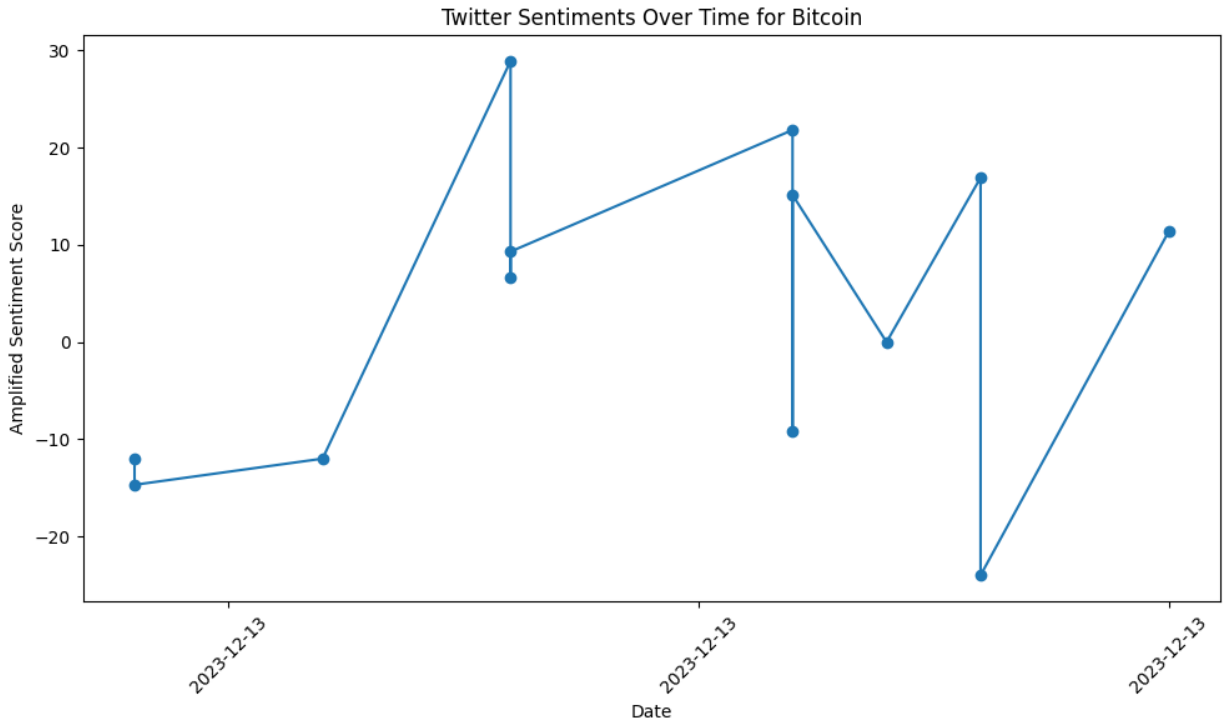
Calculations from Database

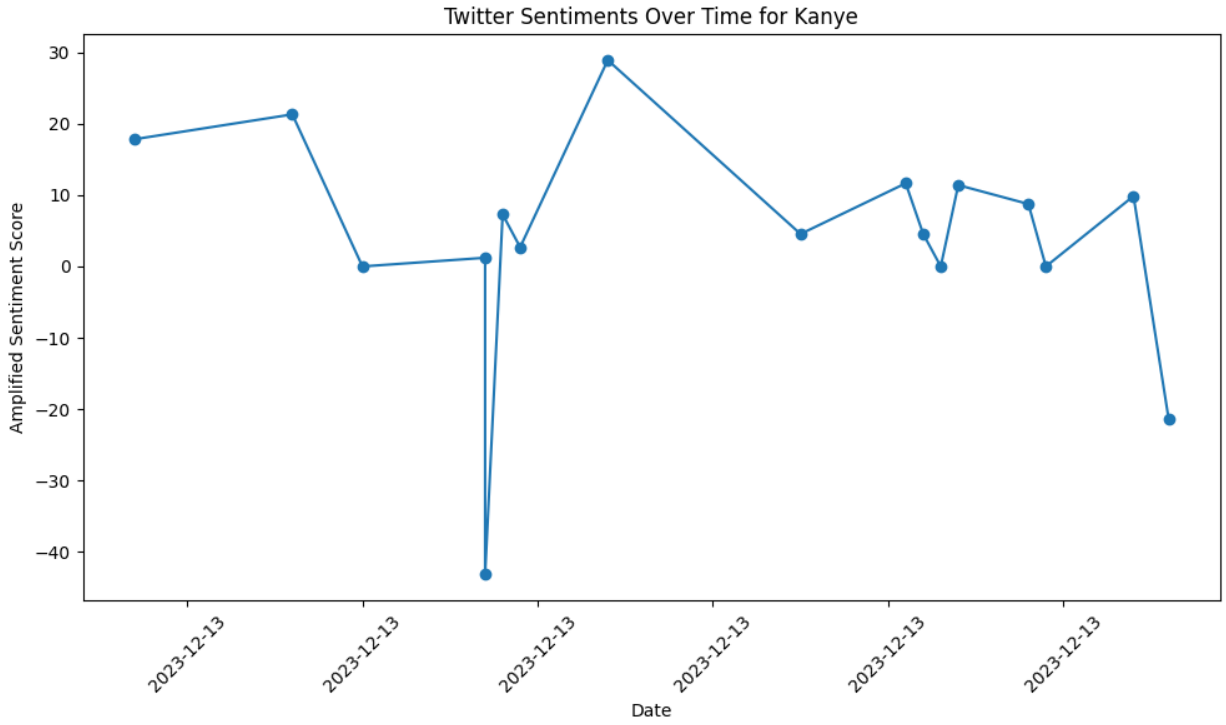


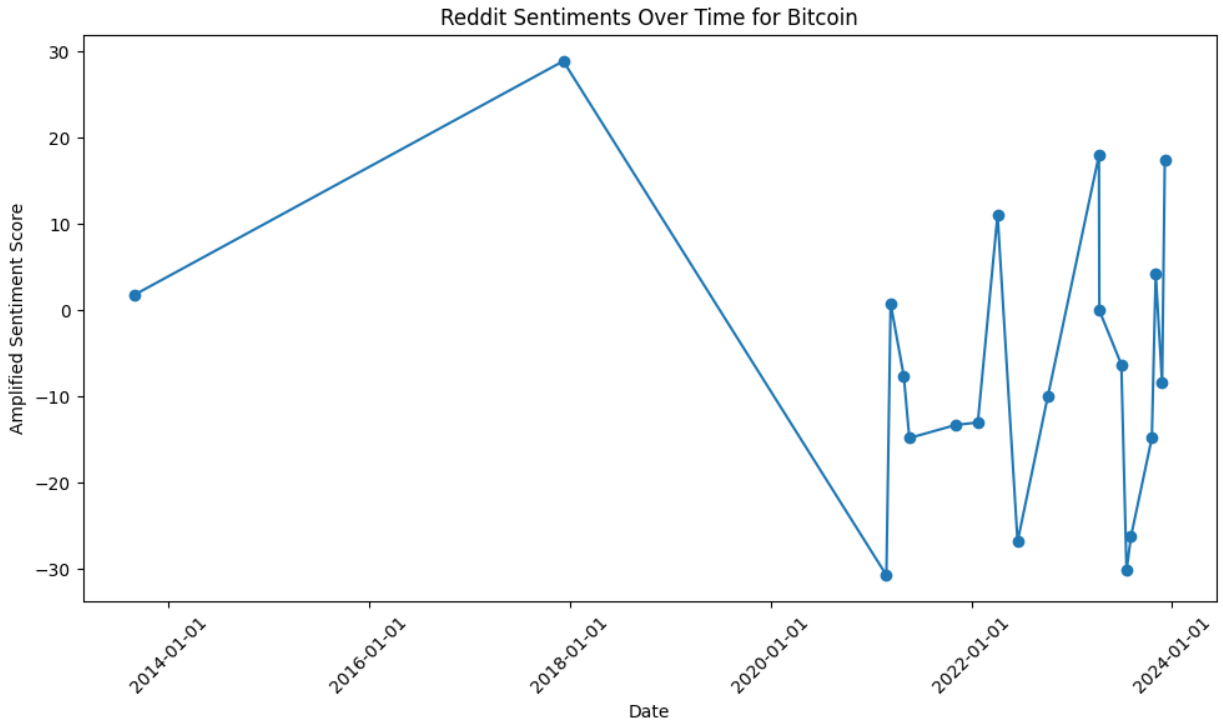
Visualizations

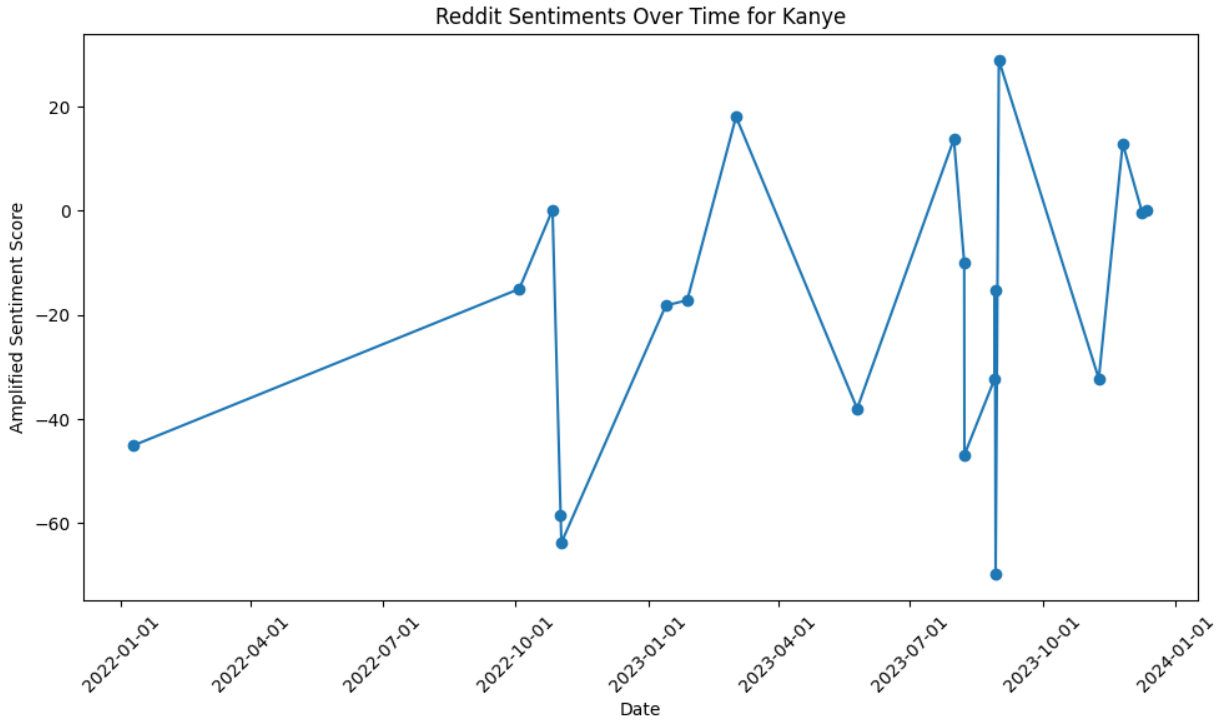


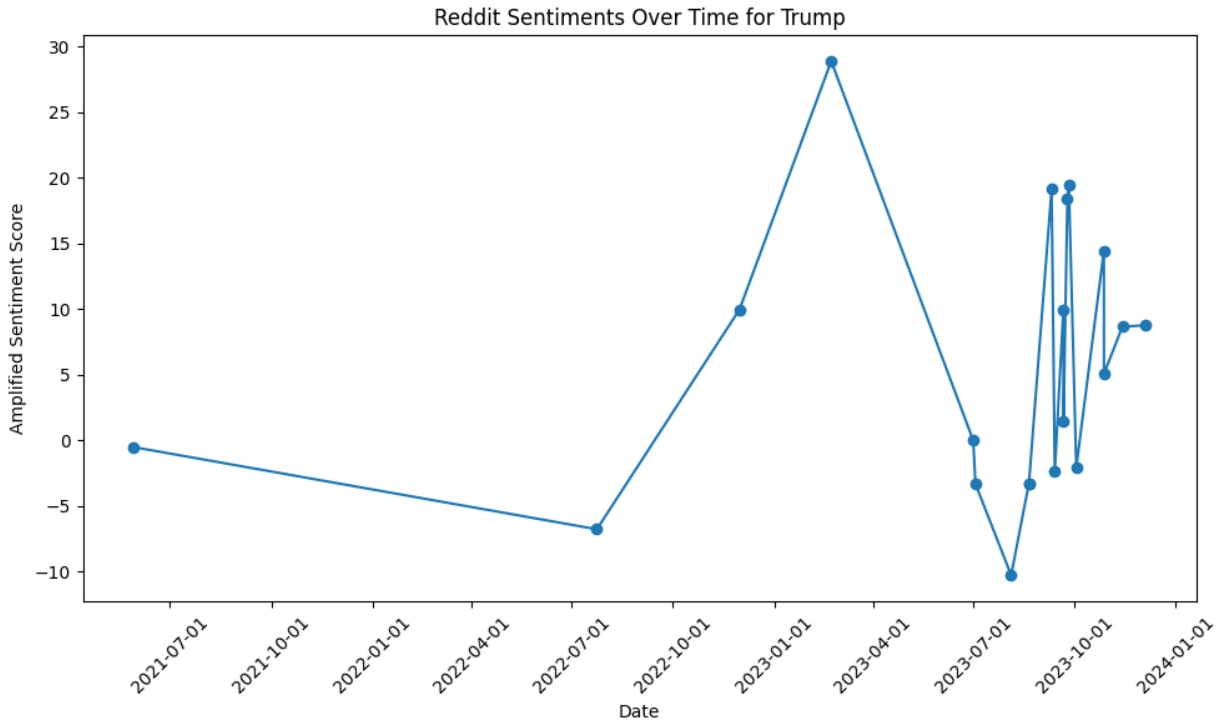


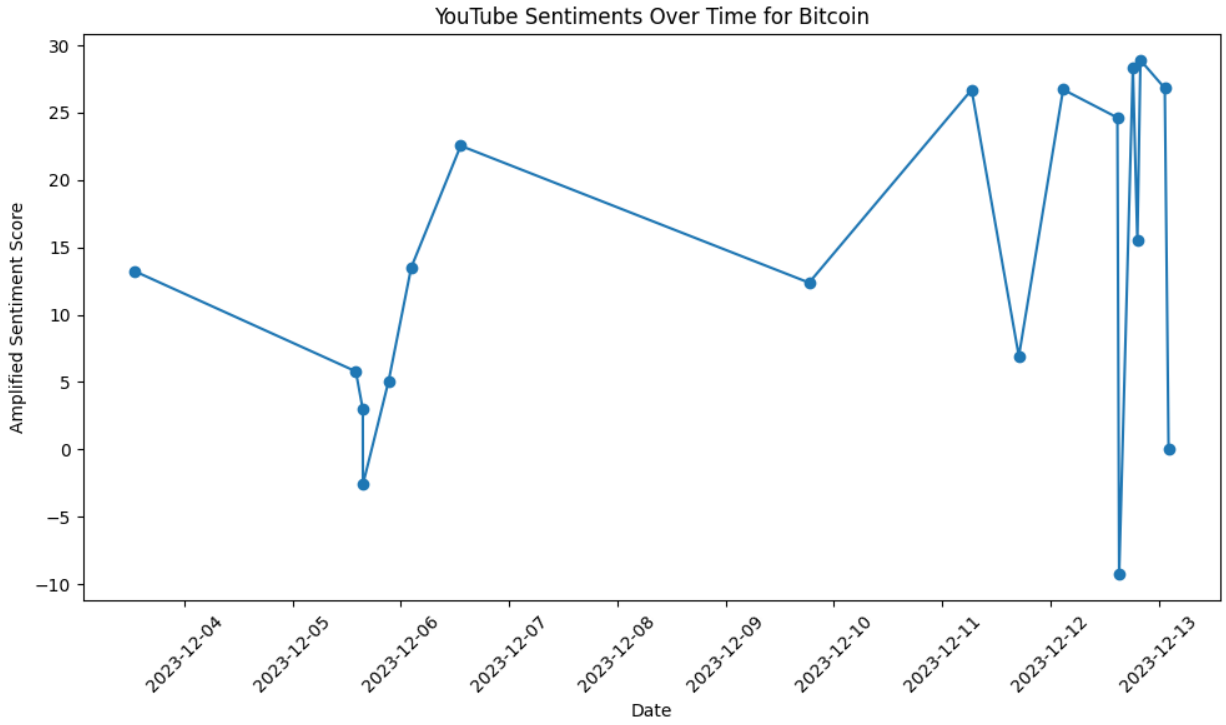

Twitter Sentiments Over Time for Bitcoin
Amplified Sentiment Score
30
20
10
0
−10
−20
2023-12-13
2023-12-13
2023-12-13
Date


Twitter Sentiments Over Time for Kanye
Amplified Sentiment Score
30
20
10
0
−10
−20
−30
−40
2023-12-13
2023-12-13
2023-12-13
2023-12-13
2023-12-13
2023-12-13
Date


Reddit Sentiments Over Time for Bitcoin
Amplified Sentiment Score
30
20
10
0
−10
−20
−30
2014-01-01
2016-01-01
2018-01-01
2020-01-01
2022-01-01
2024-01-01
Date


Reddit Sentiments Over Time for Kanye
Amplified Sentiment Score
20
0
−20
−40
−60
2022-01-01
2022-04-01
2022-07-01
2022-10-01
2023-01-01
2023-04-01
2023-07-01
2023-10-01
2024-01-01
Date


Reddit Sentiments Over Time for Trump
Amplified Sentiment Score
30
25
20
15
10
5
0
−5
−10
2021-07-01
2021-10-01
2022-01-01
2022-04-01
2022-07-01
2022-10-01
2023-01-01
2023-04-01
2023-07-01
2023-10-01
2024-01-01
Date


YouTube Sentiments Over Time for Bitcoin
Amplified Sentiment Score
30
25
20
15
10
5
0
−5
−10
2023-12-04
2023-12-05
2023-12-06
2023-12-07
2023-12-08
2023-12-09
2023-12-10
2023-12-11
2023-12-12
2023-12-13
Date

Instructions for Running The Code

The code is all done in the background. All you have to do is run the mainFile.py file. The code will prompt you with asking for inputs for topics you want to test. after Giving 3 topics, the code will run and after querying the APIs and retrieving from the database, will give you the results in text format in output.txt

Step 1. Clone repository: $ git clone https://github.com/sammalik111/sentimentAnalysis.git

Step 2. Install packages: $ pip3 install praw tweepy google-api-python-client textblob

Step 3. Run program: $ python3 mainFile.py

Step 4. Give 3 topics when prompted

Step 5. Wait for information gathering

Step 6. Observe Data

Documentation for Functions

grabData.py

- **fetch_reddit_data(topic)**: Fetches and analyzes data from Reddit. Input is a topic string, and output is a list of dictionaries containing data for each post.
- **fetch_youtube_data(topic)**: Retrieves and analyzes data from YouTube. Input is a topic string, and output is a list of dictionaries with video data.
- **calculate_comments_sentiment(tweet_data):** Calculates the sentiment score of a tweet. Input is a dictionary containing tweet data, and output is a sentiment score.
- **fetch_twitter_data(topic):** Fetches and analyzes data from Twitter. Input is a topic string, and output is a list of tweet data dictionaries.

mainFile.py

- **calculate_sentiments_and_times(data)**: Extracts sentiment scores and times from data. Input is a list of data items, and output is two lists (sentiments and times).
- **main()**: Coordinates data gathering.Then pushes the information to the database, and grabs everything from the database it needs to output. calculates all the

information needed to be visualized then calls visualize to display everything.

visualize.py

- **amplify_variation(scores, base):** Normalizes and amplifies sentiment scores onto a 0-100 score. Input is a list of scores; output is an amplified list of scores.
- **TimePlot(dates, scores, title):** Creates a time-series plot. Inputs are lists of dates and scores, and a title string.
- **histPlot(scores, color, title):** Generates a histogram. Inputs are a list of scores, color for the plot, and a title string. It then outputs the figure and scores to their respective files.
- **barPlot(topics, avg_sentiments, title):** Creates a bar plot. Inputs are lists of topics and average sentiments, and a title string. It then outputs the figure and scores to their respective files.

DatabaseMaker.py

- **fetch_data_from_database(topic, api_name)**: This function fetches data from the database for a specific topic and API. It takes two parameters, topic (a string) and api_name (a string), and returns a list of dictionaries containing the fetched data.

- **create_and_insert_table(data, topic, api_name):** This function creates and inserts data into a table in the database. It takes three parameters, data, topic, and api_name . It creates a table for the topic and API if it doesn't exist and inserts data into the table while ignoring duplicates.

Resource Documentation

Sentiment Analysis Accuracy

- Date: 12/04/2023
- Issue Description: Challenges in ensuring accuracy of sentiment analysis due to the diverse and ambiguous nature of social media content.
- Location of Resource: Online tutorials, and TextBlob documentation.
- Result: Partially solved. The use of the TextBlob library provided a baseline for

sentiment analysis, but the inherent ambiguities of natural language in social media meant that accuracy was not always optimal.

Handling Inconsistent Time Metrics

- Date: 12/06/2023
- Issue Description: Format of dates was inconsistent across all platforms
- Location of Resource: Data cleaning and preprocessing tutorials, Stack Overflow for Python
- Result: Solved with a lambda function, making sure all data points in visualizations were in order. Implementing data cleaning and preprocessing steps helped in normalizing and sanitizing the data, but some level of data inconsistency remained unavoidable.

Data Consistency Across Platforms

- Date: 12/07/2023
- Issue Description: Difficulty in standardizing and comparing data across different platforms due to varied data structures and metrics.
- Location of Resource: Platform-specific API documentation, data normalization techniques from data science forums and publications.
- Result: Solved to a degree. While each platform's API presented unique data structures, the development of a custom data processing pipeline allowed for the extraction and normalization of key data points (like sentiment scores, timestamps, and popularity metrics) into a consistent format for analysis.

Finding a 3rd API That Allowed for Easy Post Retrieval

- Date: 12/09/2023
- Issue Description: Difficulty in finding free, legal, first-party APIs for post retrieval, particularly for Twitter.
- Location of Resource: Extensive online research, including developer forums and official API documentation.
- Result: Solved. We paid ~$28 for Basic access to Twitter API v2 (paid prorated amount due to already being partially through the month)