Feb 13th:

- Explored the TikTok API.

Feb 20th:

- Set up the server.
- Explored the data.

Feb 25th:

- Converted all JSON files starting with "mental" into individual dataframes and combined them into a single dataframe, named `combined_df`.
- Created a text column by combining "video_description", "hashtag_names", and "voice_to_text" columns.
- Preprocessed the combined text column and performed topic modeling using CountVectorizer and LDA.
- Created a separate “topic” column and assigned respective topics to each video.

March 4th:

- Combined domain-specific stop words with default stop words to enhance text preprocessing.
- Experimented with different numbers of topics.
- Calculated an importance score for each word by comparing its relative frequency within a specific topic (F_in) against its average frequency in all other topics (F_out). This metric highlights words that are uniquely characteristic of each topic, helping to select more discriminative keywords for improved topic clarity.
- Used that importance score to re-rank and refine topic keywords for improved topic distinctiveness.

March 11th:

- Combined social media-specific stopwords with the existing stopword set to further refine text preprocessing.
- Tried using a TF-IDF vectorizer paired with NMF for topic modeling.
- Experimented with BERTopic to leverage transformer-based embeddings for improved semantic topic extraction.

March 20th:

- Performed descriptive analysis on topics generated by NMF, calculating average like count, average view count, average comment count, unique users, and total videos per topic.
- Visualized the top 5 region codes per topic.
- Cloned the `tiktokresearch` Git repository to extract comments.
- Filtered the `combined_df` for videos with `comment_count > 0`, extracted the unique video IDs from the `id` column of the filtered DataFrame, and compiled them into a list.

- Extracted comments for the first 1500 video IDs, converted them into JSON files, and saved the files.

March 25th:

- Visualized the top 5 region code proportions per topic.
- Extracted comments for the next 1500 video IDs (i.e., up to 3000 video IDs), converted them into JSON files, and saved the files.
- Combined all the comments from these 3000 videos into a single JSON file.
- Converted the JSON file into a DataFrame and performed sentiment analysis on the comments.
- Used VADER's `SentimentIntensityAnalyzer` to compute polarity scores and the `Detoxify` model to obtain toxicity scores.

April 1st:

- Extracted comments for the next 3000 video IDs (i.e., up to 6000 video IDs).
- Utilized previously computed polarity and toxicity scores for each comment, saving this dataset as `sentiment_data`.
- Retrieved topic data from the topic modeling process, where each video was assigned a topic.
- Joined the sentiment_data with the topic data using the `video_id` and aggregated the toxicity and polarity scores at the video level.
- Further aggregated these scores by topic.
- Visualized the following:
    - Average compound score by topic
    - Average toxicity score by topic
    - Distribution of compound scores for videos
    - Distribution of toxicity scores for videos
    - Video level Compound score vs toxicity

April 8th:

- Applied a log odds method on the NMF with TF-IDF vectorizer topics to rearrange the top words. This method recalculates word importance by comparing each word's frequency in a target topic against its frequency in the background corpus. As a result, the method reorders the topic keywords—highlighting more distinctive and discriminative words for each topic.
- Extracted comments for the next 2500 video IDs (i.e., up to 8500 video IDs).
- Combined the new comments from 5000 video IDs with the previous 3000 video IDs DataFrame and performed sentiment analysis as before.
- Visualized the average and distribution of severe toxicity, obscene, threat, insult, and identity attack scores along with the overall toxicity score (as previously visualized).

- For the topic data, topics were assigned using the preprocessed text column, which was created by combining the "video_description", "hashtag_names", and "voice_to_text" columns and then applying cleaning and normalization steps.
- Performed sentiment analysis on this preprocessed text (similar to the analysis done on comments) by obtaining polarity and toxicity scores.
- Visualized the averages of compound score, toxicity, severe toxicity, obscene, threat, insult, and identity attack scores by topic as derived from the topic data.

April 22nd:

- Extracted comments for the next 1500 video IDs (i.e., up to 10,000 video IDs).
- Updated the comments DataFrame with all comments extracted up to 10,000 video IDs and re-ran the same sentiment analysis workflow on the updated dataset.
- Created box plots to visualize:
    - Compound sentiment score distributions by topic (for comments).
    - Toxicity, Severe Toxicity, Obscene, Threat, Insult, and Identity Attack score distributions by topic (for comments).
    - Compound sentiment score distributions by topic (for video-level preprocessed text).
    - Toxicity-related score distributions by topic (for video-level preprocessed text).
- Updated previously created histograms of per-video compound and toxicity-related scores (from comments) by applying log scale to the y-axis.
- Created new histograms (with log-scaled y-axis) for video-level sentiment and toxicity scores, showing the distribution of:
    - Compound sentiment scores
    - Toxicity, Severe Toxicity, Obscene, Threat, Insult, and Identity Attack scores (from preprocessed video text)
- Manually explored sample videos from each of the 8 topics using username and video ID to verify topical relevance, review comment tone, and assess toxicity in context.

April 29th:

- Extracted comments for the next 2000 video IDs (i.e., up to 12,000 video IDs).
- Plotted violin plots instead of box plots, as box plots didn’t reveal much insight. Visualized:
    - Compound Score Distribution by Topic (comments)
    - Toxicity Score Distribution by Topic (comments)
    - Compound Score Distribution by Topic (videos)
    - Toxicity Score Distribution by Topic (videos)
- Plotted histograms for:
    - Compound Score Distribution by Topic (comments)
    - Compound Score Distribution by Topic (videos)
- Uploaded the final code and project presentation to GitHub.