Python Scraping:

1. Imported relevant libraries, such as the Google API client and Pandas.
2. Created credentials on Google Cloud to obtain the developer key required for accessing the YouTube API and scraping comments.
3. Developed a function to iterate through each page of comments using the pageToken parameter.
4. Converted the scraped data into a Pandas DataFrame for further processing.
5. Conducted minor cleaning and transformation tasks, such as removing the '@' symbol from the Username column.
6. Saved the processed data to the local system using Pandas.

Power BI:

*Power Query*

1. Added a title text box to provide context and clarity to the report.
2. Created a pie chart to visualize the distribution of sentiment scores by sentiment type, and formatted it for clarity.
3. Developed a bubble plot to illustrate sentiment scores against the number of likes, providing insights into comment engagement.
4. Constructed a table visualization displaying comments, sentiment scores, and likes, with conditional formatting applied to the sentiment score column.
5. Implemented a custom Word Cloud visualization on the second page, accompanied by a title text box.
6. Configured Visual settings to exclude common words like "the" and "and" from appearing in the Word Cloud.
7. Added a slicer to filter comments by sentiment type, enhancing user interactivity.
8. Included navigation buttons on both pages for seamless page navigation within the report.
9. Adjusted the layout for optimal viewing on mobile devices.
10. Saved and published the report to the Power BI service.
11. Created a new workspace on the Power BI service to house the uploaded report.
12. Created a new dashboard and pinned the report for live monitoring and analysis.