

Final Project: Iteration 02 Report

DS5110: Introduction to Data Management and Processing
Fall 2024

Project Scope and Plan: Social Media Analytics

Sanjiv Motilal Choudhari , Shivram Nekkanti

October 14, 2024

Contents

1	Project Scope	2
2	Project Plan	2
2.1	Timeline	2
2.2	Milestones	3
2.3	Team Roles	3
3	Team Discussion Summary	3
3.1	Skills Assessment	3
3.2	Tools & Technologies	3
3.3	Team Responsibilities	3
4	Skills & Tools Assessment	4
4.1	Skills Gaps & Resource Plan	4
4.2	Tools	4
5	Initial Setup Evidence	4
5.1	Project Repository	4
5.2	Setup Proof	4
6	Progress Review	4
6.1	Progress Update	4
6.2	Issues Encountered	4
7	Revised Project Plan	4
7.1	Updated Plan	4
7.2	Justification for Changes	5
8	Appendix	5

1 Project Scope

The project aims to develop a social media analysis tool using static YouTube data to monitor brand performance, conduct sentiment analysis on reviews, and track marketing campaigns. Although the analysis is based on static data, the system will try simulating real-time scenarios to offer actionable insights into customer sentiment, brand mentions, and engagement metrics.

Key deliverables include sentiment analysis reports that highlight customer emotions towards brands, trend detection for monitoring the impact of campaigns, and early detection of crises that could affect brand reputation. Additionally, user behavior insights will support content strategy, helping brands tailor their content to the audience preferences.

The project uses natural language processing (NLP) techniques for sentiment analysis and data visualization tools to present insights effectively. We will utilize the YouTube API for data collection, applying Python and libraries such as Pandas and NLTK for processing and analysis. While campaign tracking and content strategy are significant focus areas, the system will also assist in crisis management by measuring real-time user engagement and campaign effectiveness.

Overall, this project will provide an analytical tool that supports data-driven decision-making in the fast-paced digital landscape, aiding brand monitoring, engagement tracking, and customer retention strategies

The project's main objectives are:

- Monitor brand performance using YouTube API data for sentiment analysis.
- Implement tools for campaign tracking
- Implement tools content strategy optimization and customer retention strategies.
- Provide actionable insights to enable tailored responses to different scenarios

2 Project Plan

2.1 Timeline

The overall timeline for the project is divided into phases:

- **Week 1 (October 7 - October 13):** Define the project scope, finalize the team roles, and conduct a skills assessment to identify relevant tools and technologies for sentiment analysis and data visualization.
- **Week 2 (October 14 - October 20):** Set up the project repository, collect sample static YouTube data, and begin coding basic functionalities for data extraction and pre-processing using the YouTube API.
- **Week 3 (October 21 - October 27):** Continue development by working on the backend system for data storage and processing. Begin implementing sentiment analysis using Natural Language Processing (NLP) techniques.
- **Week 4 (October 28 - November 3):** Complete the backend system integration. Start working on the frontend for visualizing data insights, including sentiment reports and engagement metrics. Draft technical documentation
- **Week 5 (November 4 - November 10):** Finalize the system with all core functionalities in place, including sentiment analysis, trend detection, and crisis management features. Conduct thorough testing and refine the user interface.
- **Week 6 (November 11 - November 17):** Revise and finalize the technical report and Power-Point presentation, showcasing key features such as sentiment analysis, trend detection, and user behavior insights.
- **Week 7 (November 18 - November 28):** Conduct the final presentation, submit the technical report, and close the project with a review of lessons learned and future opportunities.

2.2 Milestones

Key milestones include:

- Project Scope and Plan Finalized (October 7).
- GitHub Repository Setup and Initial Data Collection (October 14).
- Data Processing and Sentiment Analysis Completion (November 10).
- Final System Testing and Report Draft (November 24).
- Final Presentation and Report Submission (November 30).

2.3 Team Roles

- **Shivram Nekkanti:** Responsible for data extraction, data cleaning, data visualization, and overall system architecture. Also responsible for project management using GitHub and Overleaf.
- **Sanjiv Motilal Choudhari:** Implementation of sentiment analysis models, responsible for presenting insights. Responsible for documentation, and report writing.

These roles are flexible and may change as the project progresses. Both team members will collaborate on testing and quality assurance throughout the project.

3 Team Discussion Summary

3.1 Skills Assessment

To successfully complete the social media analytics project, the team requires proficiency in the following skills:

- Data extraction and processing from static datasets (e.g., YouTube).
- Sentiment analysis using Python (libraries like NLTK or Vader).
- Backend development skills in Python with Flask for server-side logic.
- Data visualization using Tableau or Power BI.
- Familiarity with GitHub for version control and collaboration.
- Report writing using Overleaf for the final technical documentation.

3.2 Tools & Technologies

The following tools and technologies will be used throughout the project:

- **Programming Languages:** Python.
- **Database:** SQLite or MySQL for storing processed data.
- **Collaboration Tools:** GitHub for version control, Overleaf for report writing.
- **Libraries:** NLTK, or Vader for sentiment analysis; Pandas for data manipulation.
- **Visualization Tools:** Tableau or Power BI for creating data visualization dashboards.

3.3 Team Responsibilities

The responsibilities of the team members are distributed as follows to ensure balanced contribution:

- **Shivram Nekkanti:** Focuses on data extraction from static datasets and data visualization, along with part of the sentiment analysis.
- **Sanjiv Motilal Choudhari:** Works on sentiment analysis, developing visualization dashboard for insights using Tableau/Power BI and contributes documentation.
- Both members will collaborate on system testing, ensuring data accuracy and functionality.

4 Skills & Tools Assessment

4.1 Skills Gaps & Resource Plan

Both team members have prior experience in sentiment analysis and data extraction, but we identified a need to enhance our skills in data visualization and database management and also a possible scenario to address real-time streaming data which requires a thorough knowledge of tools like Apache Kafka and Spark and implementing ETL pipelines. To address these, we plan to follow tutorials on creating dashboards with Tableau/Power BI and work on improving during Weeks 2, 3, and 4.

4.2 Tools

The following tools will be utilized throughout the project:

- Python and its libraries (NLTK) for data processing and analysis.
- Tableau/Power BI for creating interactive data visualization dashboards.
- SQLite or MySQL for database management and storing processed data.
- GitHub for version control, collaboration, and code management.
- Overleaf for writing and maintaining the technical report.

5 Initial Setup Evidence

5.1 Project Repository

The project repository has been created on GitHub, accessible by all team members. The repository can be found at https://github.com/SanjivDS/DS5110_Social_Media_Analytics.

5.2 Setup Proof

The development environment has been successfully set up. Screenshots of the setup process are provided in the Appendix.

6 Progress Review

6.1 Progress Update

The team has successfully completed the initial setup of developing an analysis environment and created a GitHub repository for the final project.

6.2 Issues Encountered

The major issue that we encountered was the YouTube API V3 call limitations. The API is limited to only 10,000 units of quota. The cost of calling the API depends on the API method. Extracting comments costs 1 unit of our quota, limiting us to only 10,000 comments per-day.

7 Revised Project Plan

7.1 Updated Plan

After reviewing our progress, we have decided to shift our focus from real-time data analysis to utilizing static data from the YouTube API. This change is necessary due to time constraints and the non-availability of free APIs for extracting live social media data. Consequently, the timeline has been adjusted to accommodate this new direction.

7.2 Justification for Changes

This adjustment was required due to unforeseen challenges in sourcing reliable real-time data and the limited availability of free APIs for social media analysis. By switching to static data, we can focus on developing robust sentiment analysis and visualization tools within the available timeframe. This strategic pivot will allow us to meet project deadlines while still achieving our primary objectives in brand monitoring and analysis.

8 Appendix

```
import googleapiclient.discovery
import pandas as pd

API_SERVICE_NAME = "youtube"
API_VERSION = "v3"
DEVELOPER_KEY = "AIzaSyAFavjomo08Yqr6xHT1QCwi2uPDWlnSQeQ"

def initialize_youtube_client(api_key):
    return googleapiclient.discovery.build(API_SERVICE_NAME, API_VERSION, developerKey=api_key)

def get_video_comments(youtube_client, video_id, target_comment_count=5000):
    all_comments = []
    page_token = None

    while len(all_comments) < target_comment_count:
        response = fetch_comment_page(youtube_client, video_id, page_token)
        comments_on_page = extract_comments(response)
        all_comments.extend(comments_on_page)
        page_token = response.get('nextPageToken')
        if not page_token:
            break

    return all_comments[:target_comment_count]

def fetch_comment_page(youtube_client, video_id, page_token=None):
    request = youtube_client.commentThreads().list(
        part="snippet",
        videoId=video_id,
```


Figure 1: Initialization of a YouTube API client and the beginning of a function to fetch video comments.

```
def fetch_comment_page(youtube_client, video_id, page_token=None):
    request = youtube_client.commentThreads().list(
        part="snippet",
        videoId=video_id,
        maxResults=100,
        pageToken=page_token
    )
    return request.execute()

def extract_comments(api_response):
    return [
        item['snippet']['topLevelComment']['snippet']['authorDisplayName'],
        item['snippet']['topLevelComment']['snippet']['publishedAt'],
        item['snippet']['topLevelComment']['snippet']['updatedAt'],
        item['snippet']['topLevelComment']['snippet']['likeCount'],
        item['snippet']['topLevelComment']['snippet']['textDisplay']
    ]
    for item in api_response.get('items', []):
        pass

if __name__ == "__main__":
    youtube = initialize_youtube_client(DEVELOPER_KEY)
    video_id = "4Tm6Z1y3h94"
    max_comments_to_fetch = 5000
    comments_data = get_video_comments(youtube, video_id, max_comments_to_fetch)
    comments_df = pd.DataFrame(comments_data, columns=['author', 'published_at', 'updated_at', 'like_count', 'text'])
    print(comments_df.head(10))
```

Figure 2: Continuation of Python code for fetching and extracting YouTube video comments, including the main execution block.





	author	published_at	updated_at	\
0	@PursuitofWonder	2024-10-01T15:01:30Z	2024-10-01T15:01:30Z	
1	@Naratifan	2024-10-13T23:46:30Z	2024-10-13T23:46:30Z	
2	@parvanehalipouralmajavan8234	2024-10-13T23:43:30Z	2024-10-13T23:43:30Z	
3	@AmritBankar	2024-10-13T22:10:35Z	2024-10-13T22:10:35Z	
4	@temetnosce6192	2024-10-12T23:34:36Z	2024-10-12T23:36:05Z	
5	@RichardS-qh8mi	2024-10-12T11:14:48Z	2024-10-12T11:14:48Z	
6	@belowfactual	2024-10-11T08:15:02Z	2024-10-11T08:15:02Z	
7	@RoyKristian-l8i	2024-10-09T08:59:22Z	2024-10-09T08:59:22Z	
8	@Skyler-mw7hw	2024-10-09T08:22:24Z	2024-10-09T08:22:24Z	
9	@dawnfield4713	2024-10-08T21:37:32Z	2024-10-08T21:37:32Z	

	like_count	text
0	8	Thank you for watching. I hope this video help...
1	0	Good video, thank you !
2	0	💖🙏
3	0	I needed to hear this today
4	0	If an atom is the scale... I am a giant!-)
5	0	We've always existed and will forever, just no...
6	0	Yes this rid of my stress and also every othe...
7	1	This will stop my anxiety but soon i will stil...
8	0	Everything and everyone except jesus all their...
9	0	We're All significant in our own ways, and...

Figure 3: DataFrame output displaying the first 10 rows of extracted YouTube comments, showing author names, publication dates, and comment text

```
[ ] comments_df.head()


author      published_at  updated_at  like_count  text
0  @PursuitofWonder  2024-10-01T15:01:30Z  2024-10-01T15:01:30Z      8  Thank you for watching. I hope this video help...
1  @Naratifan        2024-10-13T23:46:30Z  2024-10-13T23:46:30Z      0  Good video, thank you !
2  @parvanehalipouralmajavan8234  2024-10-13T23:43:30Z  2024-10-13T23:43:30Z      0  💖🙏
3  @AmritBankar      2024-10-13T22:10:35Z  2024-10-13T22:10:35Z      0  I needed to hear this today
4  @temetnosce6192   2024-10-12T23:34:36Z  2024-10-12T23:36:05Z      0  If an atom is the scale... I am a giant!-)

[ ] comments_df.shape

(5000, 5)

[ ] comments_df.to_csv('comments.csv', encoding='utf-8', index=False)
```

Figure 4: DataFrame summary showing the first 5 rows of YouTube comments and the total shape of the dataset, followed by code to export the data to a CSV file