Team Name: NoPlan

Team Members:
- Riya Berry | USC ID: 7828642721
- Benjamin Lu | USC ID: 8406742793
- Ryan Tung | USC ID: 2887643877

Github Link:
https://github.com/18rberry/560_labs_no_plan/tree/main/Reddit_Forum_Analysis_Clustering

1) Initial Setup

a) Tools & Libraries

To run our code, please have the following libraries installed:
- **nltk, gensim, sklearn**
- Additionally, please run all import cells at the top of each code file

b) Resource

We used BeautifulSoup4 to scrape Reddit data

2) Data Collection and Storage

As we are all USC students, we **choose the topic r/USC on Reddit** (https://old.reddit.com/r/USC/). We wanted to explore what types of topics and themes students tend to discuss related to USC.

We use **NUM_INPUT** as the number of posts to fetch (inputted from the user) and repeatedly retrieve Reddit posts from r/USC, along with their *title, label, author, number of comments, score, number of likes, and time posted (timestamp).*

3) Data Pre-Processing

We implemented several techniques to pre-process our data including:
- **Text pre-processing:**
  - Lowercase the text, remove punctuation/stopwords, and tokenize the text using NLTK's library
  - removing extra whitespace and special characters
- **encode reddit usernames**:
  - by encoding strings to bytes, applying a hash function, and then converting to integers
  - We did this to protect Reddit users' privacy and anonymity
- **Convert the 'timestamp' column** to the correct datetime format

- **Remove advertisements** or promoted content from Reddit (tagged as 'data-promoted')
    - We did this to ensure all content was relevant to USC and student life
- **Drop null or duplicate rows**
- Image OCR was considered but not implemented
    - The wide majority of the content was text-heavy rather than image-heavy, and image OCR would heavily increase our latency for real-time scraping. However, this is an area of future improvement

4) Forum Analysis & Clustering Methods

Now that we have scraped the content from the Reddit thread about USC, we want to group similar messages together and understand the various clusters/themes represented.

a) Message Content Abstraction

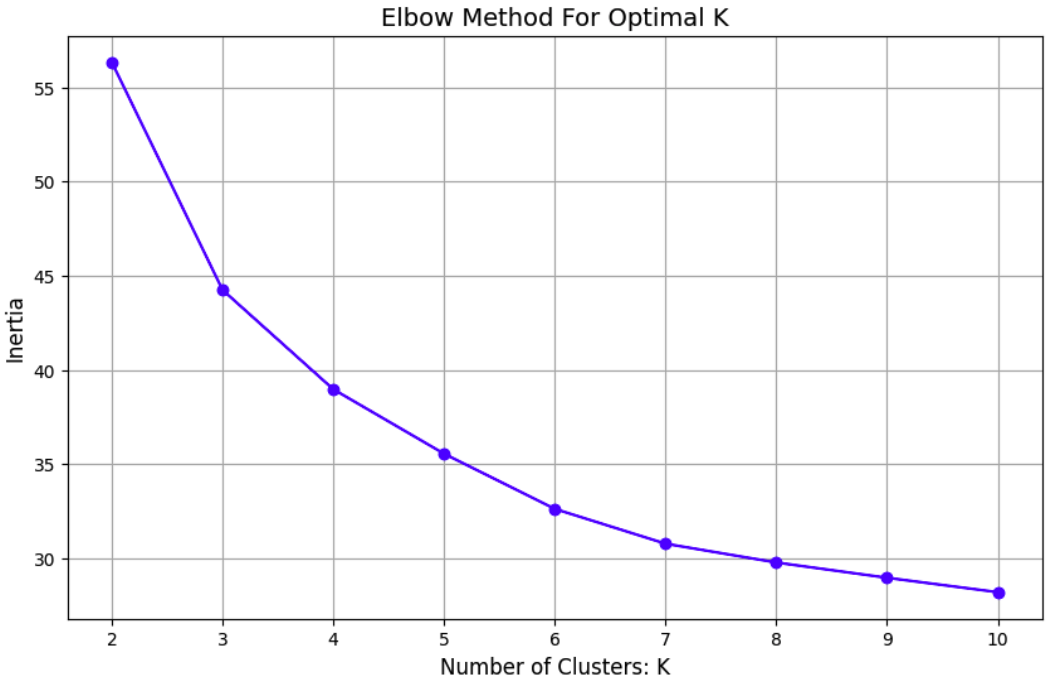
Our first step was to create document embeddings for the Reddit posts – these embeddings semantically represent each post but are easier for the mode to work with rather than raw text data. To do so, we trained a Doc2Vec model on our scraped data, producing an embedding for each post. We used the **following parameters for our Doc2Vec model:**

- *vector_size = 50*: smaller dimension of document vectors works better for shorter text
- *min_count = 1*: this tells us the threshold to ignore words with a lower frequency than. But because the titles are short, we want this threshold to be as low as possible
- *epochs = 100*: add more training epochs
- *dm=0:* Use a distributed bag of words technique (skipgram variant of Word2Vec/ Doc2Vec)
- *workers = 4:* number of threads to use for model training

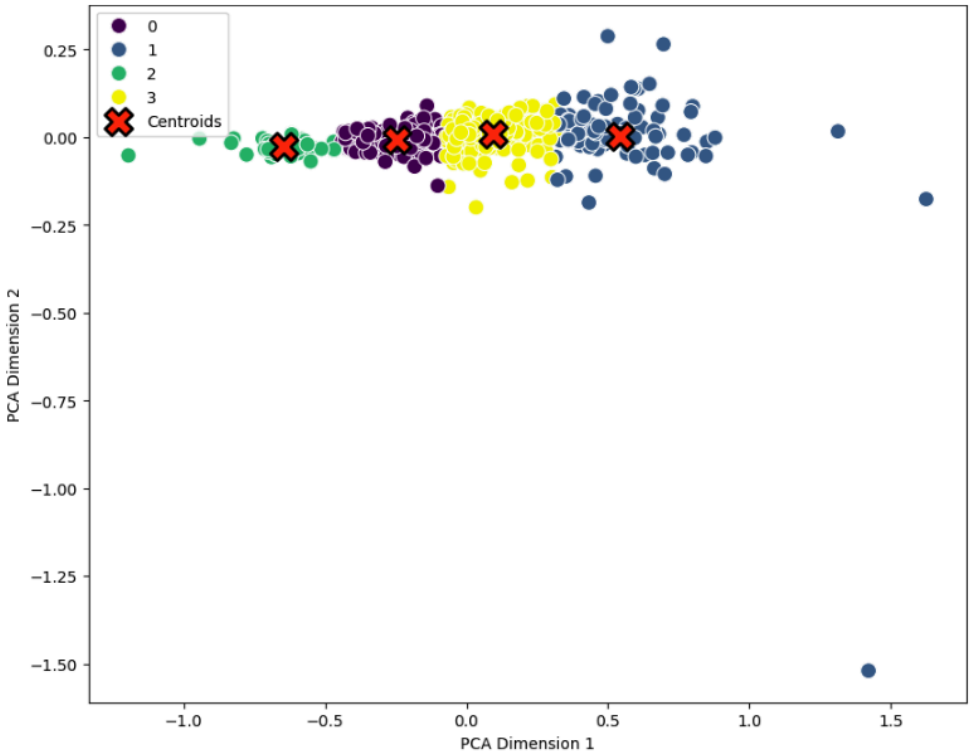
b) Clustering Messages

We wanted to use K-Means clustering to cluster our documents based on the embeddings. However, we first need to find the **optimal K for k-means clustering**. To do so, I plotted the inertia (or Within Cluster Sum of Squares) for each value of K between 2 to 10. My goal was to find the "elbow" point in the curve, or the point where the inertia stops decreasing rapidly and begins to slowly decline – indicating the optimal k value because it's the best balance between maximum cluster cohesion and minimum number of clusters.

Based on the graph below, I found the **optimal number of clusters to be k=4.**

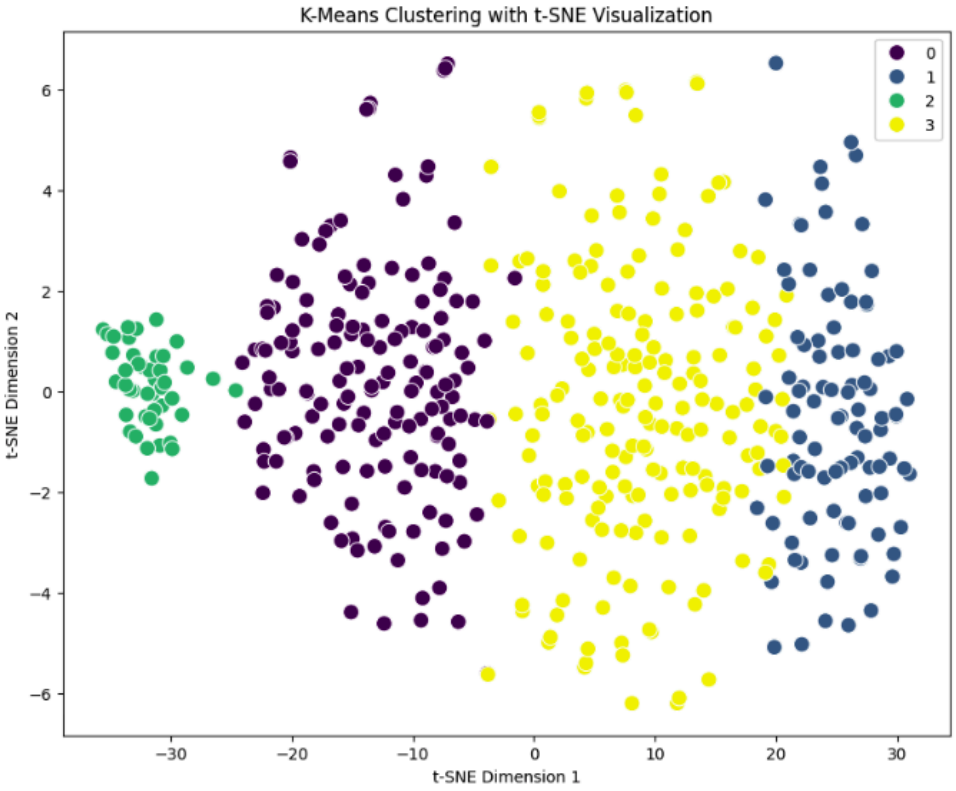


Then, I performed **Principal Component Analysis (PCA)** to reduce the number of dimensions of our document embeddings to 2D (2 dimensions) – so it's easier to plot and visualize our clusters. I also plotted the centroids for each cluster, as you can see below.



We can see that 4 distinct clusters emerge. However, the data is heavily compressed along the second PCA dimension (indicating that most of the variance is captured by the first PCA dimension).

As a result, I decided to use another **visualization method t-SNE** as it tends to give better 2-D visualization results for clustering. T-SNE doesn't have a transform method, so I decided to visualize cluster regions instead.



Now, the data and distribution of clusters looks a lot cleaner! Let's try to delve into the content that makes up each cluster. We want to understand the top 10 keywords summarizing/encompassing each cluster, as well as dig into the content of posts closest to the cluster centroids.

Below, I created a table to summarize the top keywords extracted from each cluster, as well as the top labels (and percentages) for each cluster. The labels are directly scraped from the Reddit website, and can be used to categorize our content – examples including *Academic, Question, Financial Aid, Housing, Clubs & Campus Life.*

Cluster	Total # of posts	Top Keywords and Counts	Top Reddit Labels and Percentages
0	159	• housing: 11 • transfer: 9 • campus: 9 • question: 8 • anyone: 6 • marshall: 6 • students: 6 • get: 6 • best: 5	• Question: 62 (39.0%) • Academic: 38 (23.9%) • Housing: 15 (9.4%)
1	94	• anyone: 6 • like: 6 • would: 5 • students: 5 • uscs: 5 • looking: 4 • long: 4 • campus: 4 • someone: 4	• Question: 39 (41.5%) • Academic: 27 (28.7%) • Other: 7 (7.4%)
2	45	• housing: 8 • transfer: 7 • question: 4 • campus: 4 • study: 2 • students: 2 • freshman: 2 • advice: 2 • job: 2	• Academic: 15 (33.3%) • Question: 10 (22.2%) • Housing: 7 (15.6%)
3	188	• get: 13 • anyone: 9 • campus: 7 • study: 7 • course: 6 • student: 6 • 2026: 6 • advice: 6 • marshall: 5	• Question: 72 (38.3%) • Academic: 57 (30.3%) • Housing: 13 (6.9%)

Then, we wanted to **examine the titles of the posts closest to the cluster centroids.**
Cluster 0:

- “WorkStudy Funds”
- “Chem 105a”
- “Merit Scholarships”

- “Movein tips”
- “stolen scooter”

Cluster 1:

- “tac 460 web dev project”
- “rCFB donation drive will donate on your behalf”
- “Tire Shop Taqueria Closing Time”
- “What counts as community service for Norman Topping Scholarship”
- “Dr Eric Heller ANTH321 ily”

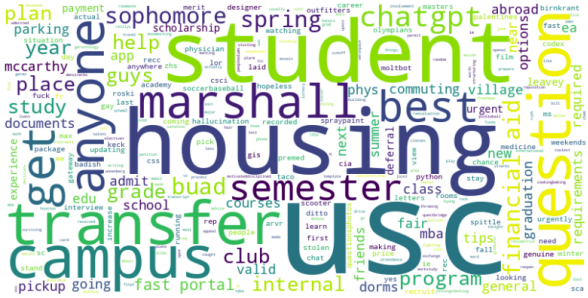
Cluster 2:

- “PHIL260”
- “heytea”
- “frats”
- “Transfer application”
- “This is so funny”

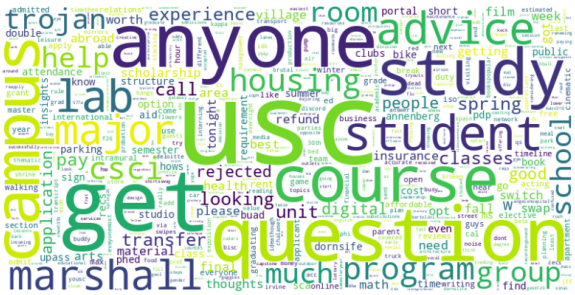
Cluster 3:

- “How do i book a room at JFF”
- “Leisure BooksPopular Reading”
- “Lost hat in lyft”
- “cowlings and ilium aircon”
- “Gateway Smell in Apartment”

Lastly, we wanted to create a WordCloud for each cluster to understand their core themes and keywords. Below are some example WordClouds:



Cluster 0: WordCloud



Cluster 3: WordCloud

Cluster 0 focuses on content about housing (made clear by the words “campus”, “mccarthy”) and it seems to be primarily content about starting USC (as shown by the words “transfer”, “student”, and “sophomore”). The keyword “housing” appears 11 times across all posts in this cluster, and “campus” appears 9 times.

Cluster 1 focuses on mainly academic content (sample post titles include “tac 460 web dev project” and a post related to the course “ANTH321”. Additionally, the Word Cloud for this cluster features keywords such as “student”, “major”, “csci”, “hbio”, “exam”, “prof” (likely reviews about professors), and “switching” (likely in relation to switching classes and majors).

Cluster 2 is the smallest cluster by far (only 45 posts) and tends to focus on **campus activities as well as social and student life.** For example, one of the posts closest to the centroid is about “heytea,” the new boba shop near campus as well as “frats” or greek / fraternity life. Keywords in the WordCloud for this cluster include “protest”, “frats”, “discord.”

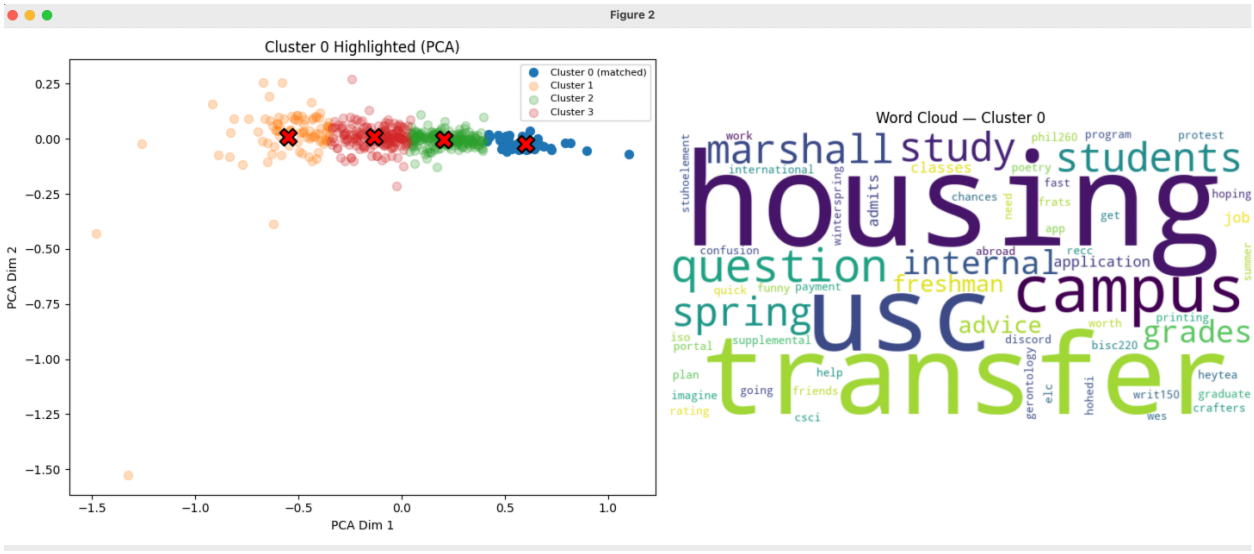
Cluster 3 primarily tends to focus on student questions (72% of the post content has the label “question”). The sample posts titles for posts closest to the cluster centroid include: “How do I book a room at JFF?” Based on the Word Cloud, we can see the main keywords for Cluster 3 are “advice”, “experience”, “classes”, "looking" and “help” – people are likely seeking advice and answers about their personal, academic, and professional lives at USC.

5) Automation

We wrapped all of the work from previous parts in an automation script to upload results to the database. We chose to use SQLite, a local version of SQL writing to a local file. When we extend this concept out to the app/product we build for the final project, we will transition into cloud resources to ensure we have production-level infrastructure. In a similar vein, the “automatic” running of the script is local as well. We must **run the program via python3 pipeline.py [minutes],** and a background daemon thread will be dedicated to “running” the scrape, process, store in SQLite process.

Minutes is a parameter for the specific interval to run the scripts and update the database. This is obviously dependent on someone’s machine constantly being open and running the program, so we will transition this to cloud servers with external schedules (IE cron/Github Actions runners) to run this job 24/7.

Our script also allows users to actively query their local db for the closest clusters to the word they choose. The photos below demonstrate the process (plt.show() is called to have the word cloud and graph pop up for better visualization!)



Finally, proper print statements were used to ensure the CLI provided clear indication to the user of what was happening and **what state of the program they were in.**

```
[SCRAPER] Fetching data from r/USC...
[SCRAPER] Fetched 50 posts.
[PIPELINE] Pre-processing scraped data...
[DATABASE] Updating database...
[DATABASE] Inserted 0 new posts (50 duplicates skipped).
[PIPELINE] Update cycle complete.

[CLUSTERING] Processing data and building clusters...
[CLUSTERING] Cluster distribution:
cluster
0     53
1     88
2    192
```

```
[PIPELINE] Background updates scheduled every 5.0 minute(s).

######################################################################

======================================================================
  INTERACTIVE MODE
  Enter keywords or a message to find the closest cluster.
  Type 'quit' or 'exit' to stop.
======================================================================
```

6) Team Discussions

Meeting Minutes:

Date	Summary	Photo
Tuesday 2/10	We met after class to discuss the division of the work. Based on our interests/time available during the week, we decided that Ryan would do the web scraping, Riya the data processing and clustering, and Ben the automation.	
Thursday 2/12	We again met after class to check in on our work. Ryan discussed challenges encountered during the web scraping and different issues he encountered while trying workarounds. Riya discussed the code she had prepared for the processing and clustering once Ryan finished pulling the data into a CSV. Ben discussed his approach with automation.	
Friday 2/13	We met virtually at 12 PM to check in on our progress. We communicated our remaining work left and estimated time we'd finish to ensure a smooth process before the Saturday deadline.	NA