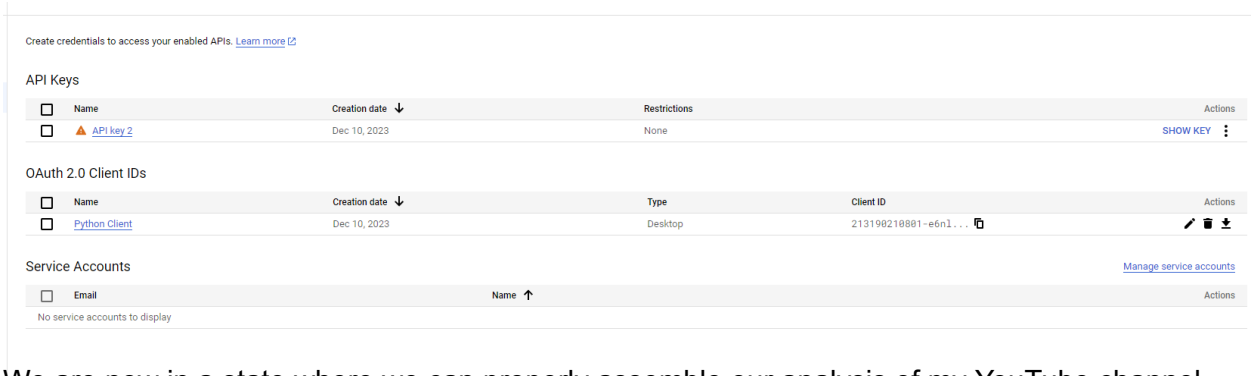
The inspiration to sign up for this class in the first place was to have a greater understanding of the YouTube recommendation system. I have a YouTube channel I run with my friend, we focus primarily on philosophy-based content. One of the biggest challenges we face is determining the connectivity of our videos, and understanding our media network in general. This class has already helped me implement techniques and practices that have informed our decision-making thus far (connecting our pages to each other to form doubly-connected nodes within our Power of Thought Network) but, the recommendation system is by far the most elusive. The motivation to for this project was to provide us with even more of an understanding of how that theoretical network is constructed on YOUTUBE's side of things. We often scratch our heads when a video does not get the proper impressions or engagement that we expected. Through the techniques implemented in this project, I hope to shed some light on how and why our videos are connected to each other, and in what way.

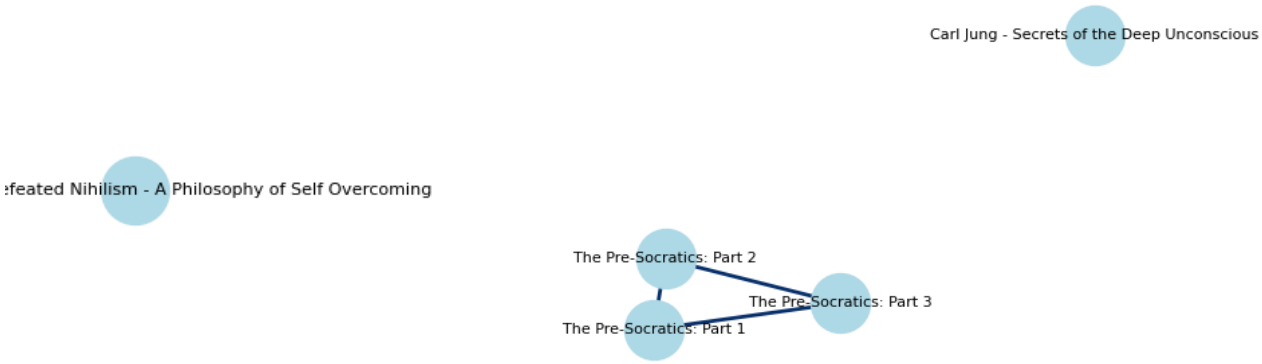
Connect to Youtube API's
Set up OAuth 2.0 Client ID and API key



We are now in a state where we can properly assemble our analysis of my YouTube channel analytics.

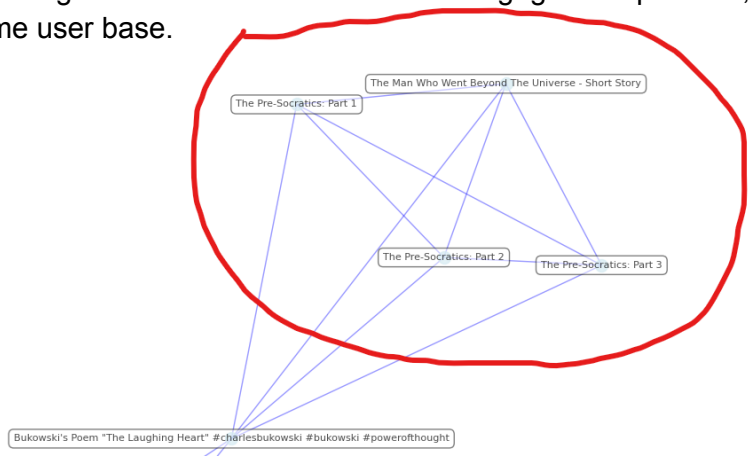
Network.py (CONTENT-BASED SYSTEM)
First, we can model a content-based recommendation system based on the YouTube API data taken from my YouTube channel (Power of Thought). We are using the title, tags, and description, from 10 of my most recent videos. We then use TF-IDF vector to convert the text data into vectors. From here we compute the cosine similarity between each video vector (similar to in class). This allows us to graph the similarities each video has with each other… as we can see here, the two Bukowski videos have the strongest connection to each other in this portion of the sprawling graph.

I find it very interesting how these videos are all connected in some way, except for the expected Bukowski correlation, I assume the tags or the description of these videos must match up in some way to make these correlations. Although this is only half the videos tested…

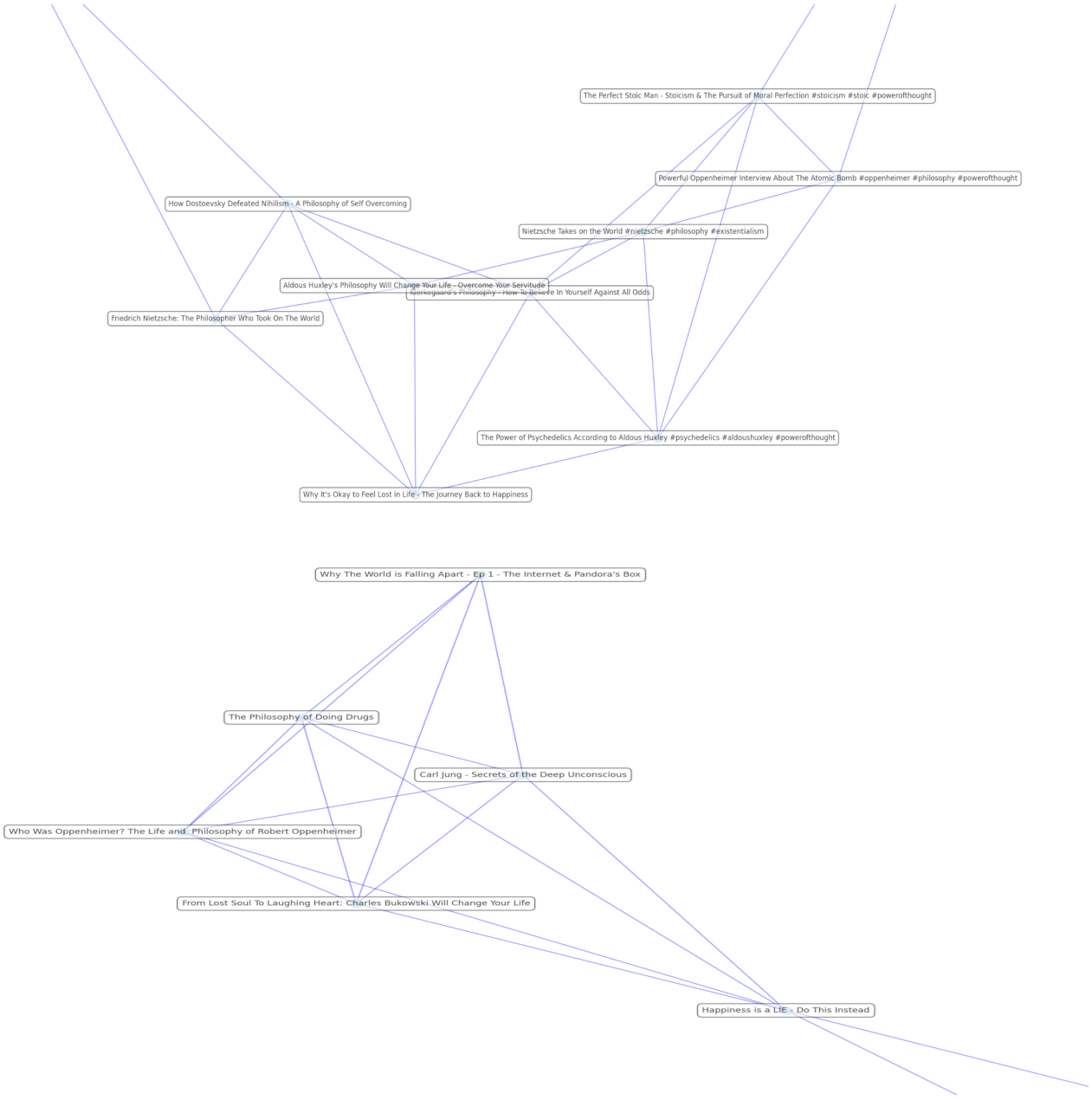


Here we can see the other five videos, the only title cut off here is “How Dostoevsky Defeated Nihilism - A Philosophy of Self Overcoming” (top-left). This and “Carl Jung - Secrets of the Deep Unconscious ” seem to have no significant overlap with any of the other videos. This is extremely helpful for me and my YouTube partner in discerning which of our videos needs more SEO attention. Lastly, the Pre-Socratic series here are perfectly connected, which I expected. These three videos are actually for MEMBERS ONLY, and would not be connected to other videos, so I’m very happy this accurately represents that.

Network.py (COLLABORATIVE FILTERING SYSTEM)
For this system, I used the last 20 videos I uploaded to get a bigger dataset. Similar to last time I placed these all in the same corpus and extracted stats including: id, title, viewCount, likeCount, and commentCount. Using view, like, and comment counts as proxies for user engagement, I implemented the NearestNeighbors algorithm for item-based collaborative filtering. The algorithm uses cosine similarity to measure the engagement profile's likeness between pairs of videos. By normalizing these statistics, we ensure that each metric has an equal impact on the similarity outcomes, despite their differing scales. The brute-force approach of the algorithm was chosen for its precision in calculating the nearest neighbors, especially since our dataset is limited to 20 nodes. This model could serve as the foundation for a recommendation engine, suggesting content with similar audience engagement patterns, thus potentially appealing to the same user base.



Again, it was able to accurately determine our MEMBERSHIP ONLY videos. This makes sense as their engagement and likeness are set only to our paying members.



When paying attention to the clusters, we can see the graph groups similar videos together based on engagement statistics.

To conclude, these graphs and this program only allowed me to pull from video statistics and not individual user statistics. For security and personal information purposes, YouTube does not allow me to pull user data in this way. But, even without individualized user data, we can see the connections made between videos based on many different video statistics.