1.

1.1 search for the political points of some singers

|  |  |
| --- | --- |
| Left-leaning | Right-leaning |
| Lady Gaga | Kanye West |
| Beyonce | Alice Cooper |
| Tylor Swift | Ted Nugent |
| Billie Eilish | Jessica Simpson |
| Katy Perry | Kid rock |

1.2 search for the most popular 10 songs of each singer on spotify

1.3 search for the official music video of each song on Youtube

1.4 search for 5 left-leaning media and 5 right-leaning media

|  |  |
| --- | --- |
| Left-leaning | Right-leaning |
| MSNBC | Fox news |
| Huffpost | Newswax |
| CNN | New York Post |
| TYT | The Daily Wire |
| CBS | Yahoo News |

For each news media, search 10 most popular videos about Joe Biden and 10 most popular videos about Donald Trump.

----------------------------------------------------------------------------------------------------------------

2. Collect comments of each video (Using R Language)

2.1 Get api keys from google developer platform.

2.2 Using R to scrape comments from videos

# using two packages

# get authentication

#collect data

--------------------------------------------------------------------------------------------------------

3. data cleaning and analysis (Using Python)

3.1

Now we have 5 “left-leaning” singers and 5 “right-leaning” singers, together with their video comments. And we have 10 media and 20 videos of each media, together with their comments. So, we have 20 datasets.

3.2

First, remove emojis. And remove comments replying to other comments to keep initial comments. And choose useful columns which are “Comment”, “User\_name”, “User\_channel\_id” and “video\_id” and add one column called “Video\_author\_id”.

3.3

Find common users between each singer’s dataset and each media’s dataset.

3.4

Based on the number of common users, we build a table containing the singer’s name and the media’s name. Table is shown below.

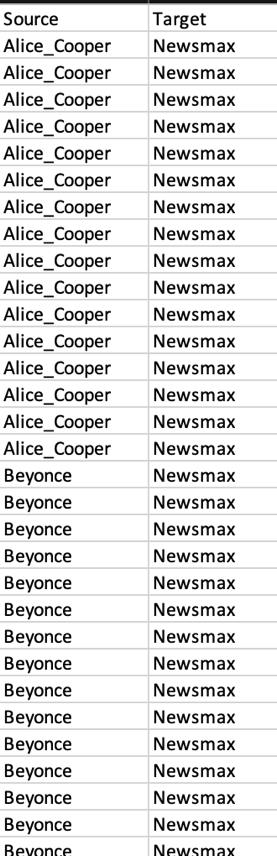
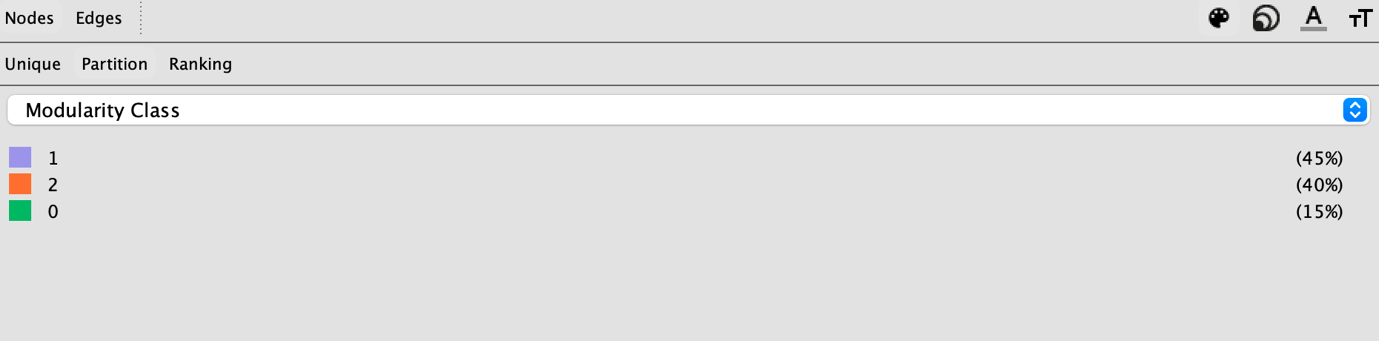


Figure 1

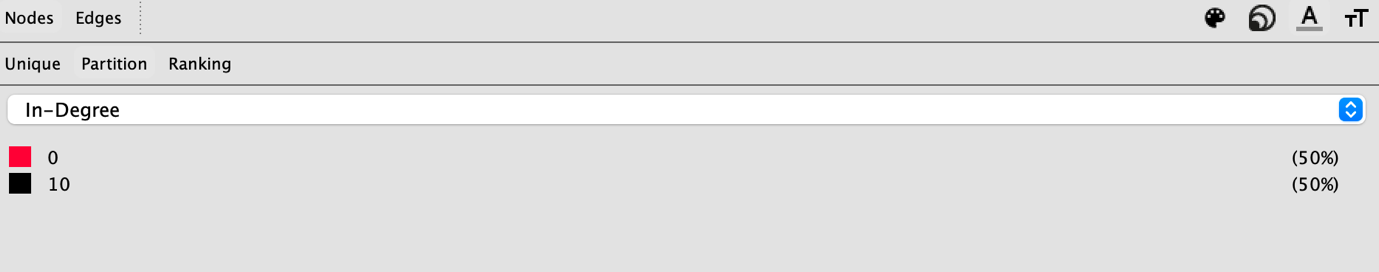
4. Use Gephi to draw a network between singers and media

4.1 Set some parameters:

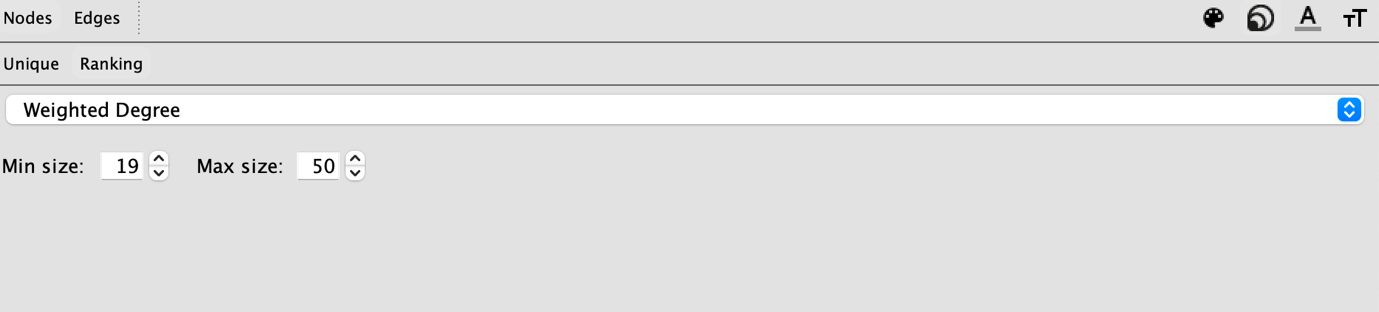
(1) Node -> color -> partition -> modularity class



(2) Node -> Label color -> in-degree



(3) Node -> size -> ranking -> weighted degree



4.2 Render the result

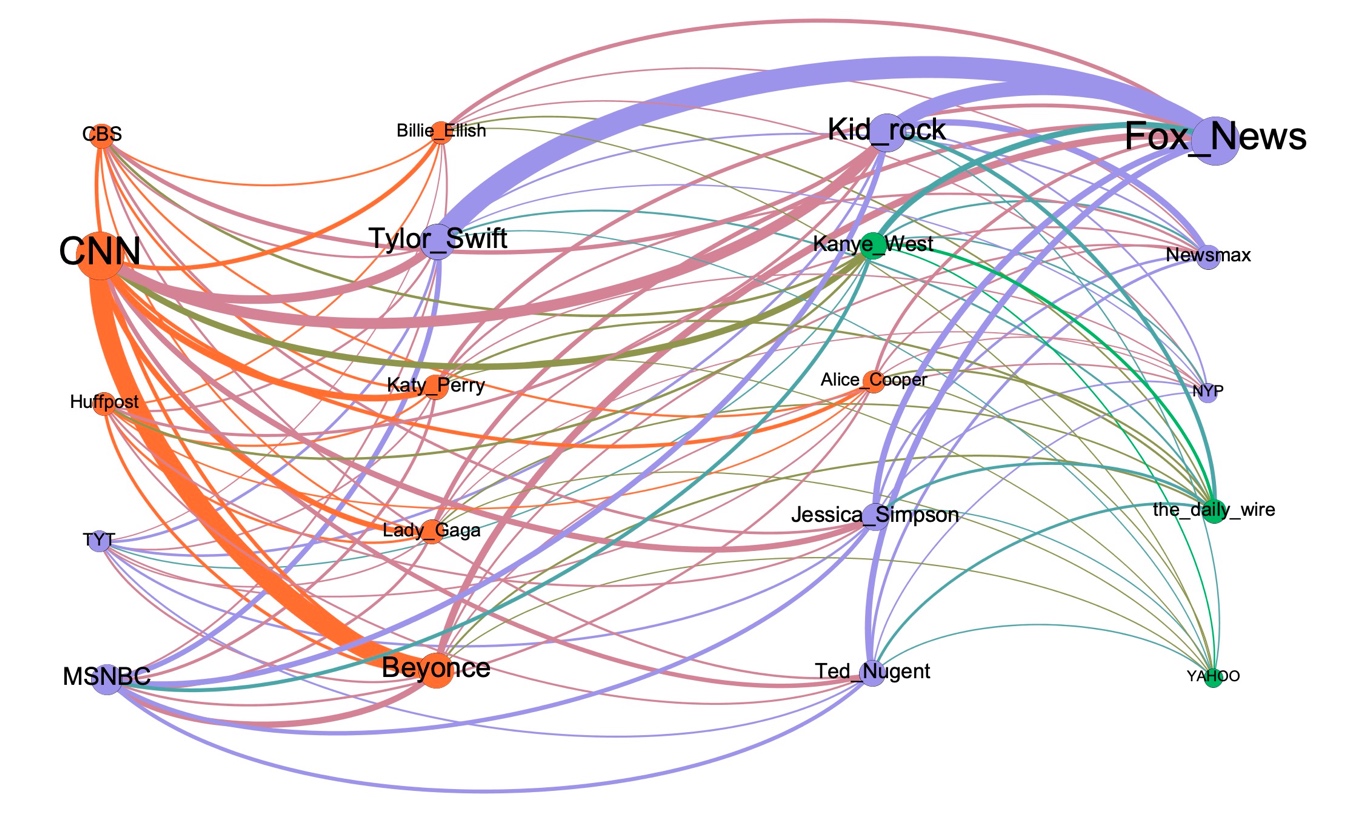


Figure 2

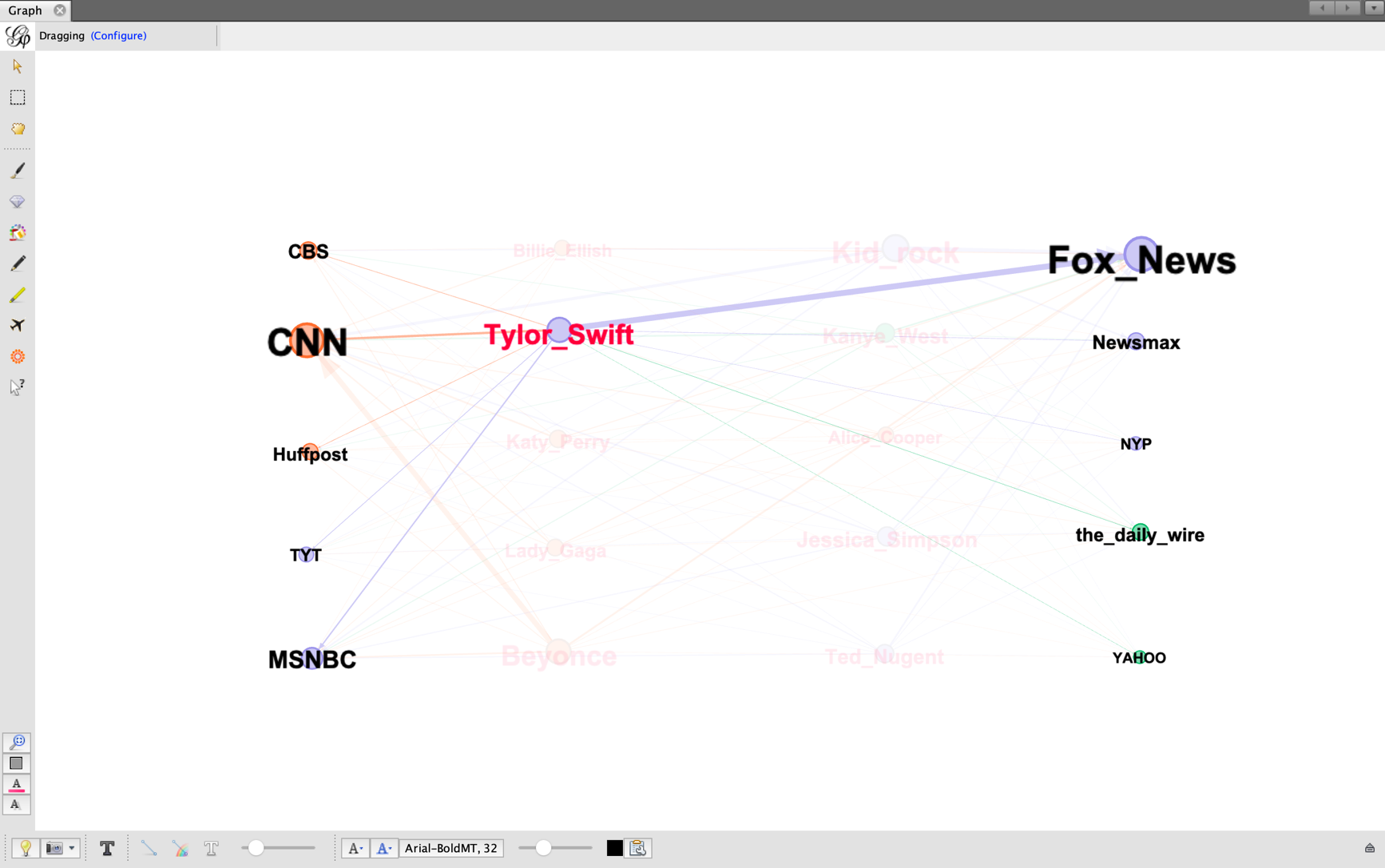


Figure 3

I will illustrate the meaning of nodes and edges respectively. First, as for the nodes, as shown in Figure 2, the color represents the community which each node belongs to. For each dataset(A) which contains the comments of one singer’s 10 videos, if A shares most common users with dataset(B) which contains the comments of 20 videos about two president candidates. Then, A and B will be rendered in one color(community). For example, in Figure 3, the node of “Tylor\_Swift” shares most common users with the node of “Fox\_News”. Then, the nodes are both blue. Moreover, the size of the node represents the popularity of the node. For example, among the 10 medias, the node of “CNN” and the node of “Fox\_News” are the largest, which means they have more YouTube users who left comments on their videos than other medias.

Second, as for the edges, the size of one edge means the number of common users shared between the two nodes. For example, in Figure 3, “Tylor\_Swift” and “Fox\_News” share most common users, so that the size of the edge between the two nodes are the widest.