I grew up obsessively poring over Tollywood, the “Hollywood” equivalent of Telugu cinema from Hyderabad, India. It was no-brainer that a project involving data had to include one of my favorite hobbies, scrutinizing Wikipedia pages and trying to map out all the possible connections between my favorite icons in the industry. I found a comprehensive dataset on Kaggle that included information about Telugu movies spanning from 1940 to 2021, such as title, music composer, genre, director, and cast. I decided to narrow down my queries to music composers, genres, and titles and see if there was a relationship between composers and the genres of movies they tended to compose for. I initially figured to create a graph and compute the centrality between the nodes to determine the most influential relationships within the industry. In a theoretical context, degree centrality can help determine the most influential node(s) by analyzing the incoming and outgoing edges for the nodes. In this case, I am looking to find which composers and genres are most prevalent in the Telugu film industry.

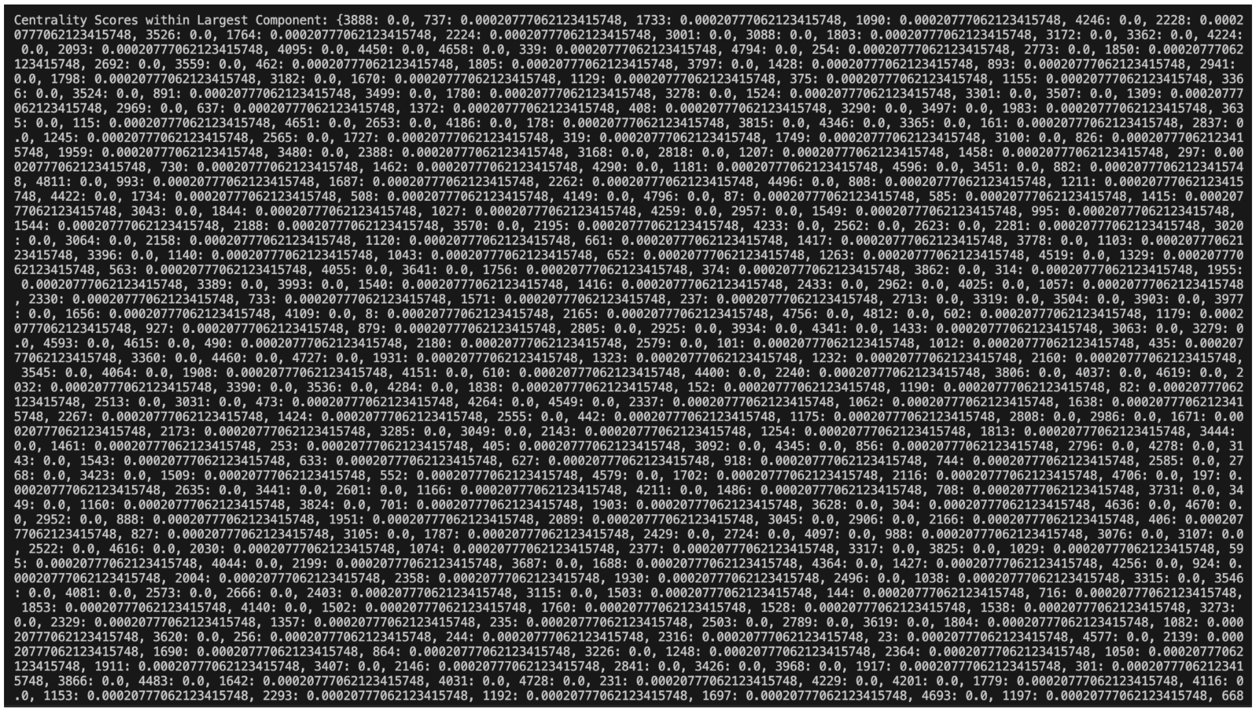
To do this, I first imported my CSV file and made functions that read the CSV file and extracted information about the movie titles, genres, and music composers. All of this information was converted to string format and stored in vectors for graph creation. For the creation of the graph, I denoted the titles to be the vertices that connected the nodes, which were made to be the composers of the film (i.e. “Ilaiyaraaja”, “K. V. Mahadevan”) and the genres (i.e. “Romance”, “Folklore”) of the specific movie. Since I was most interested in exploring the relationship between genres and composers, the aspect connecting the two nodes would be the specific movies, which is why I determined the graph edges to be the movie titles.

I implemented several methods that aided in the creation of my graph, such as one that reversed the edges of the graph, added directed edges, sorted outgoing edges, and created undirected and directed graphs. The graph was primarily composed of adjacency lists as the format for the edges and was created in a separate module titled graph.rs. I iterated through the titles of the films to form edges between the composers and genres, forming the structure of the graph. For example, a potential relationship I explored could have been structured as ("Ilaiyaraaja" -> "Comedy"), analyzing the relationship between the composer and movie genre. When creating the graph, I realized that the possibility of the existence of islands–relationships that have no connections with other nodes–was very possible. After consulting Prof. K, I was advised to run a connected components algorithm on the graph, obtain the largest component, and compute the centrality within that particular component.

To run the connected components algorithm, I adapted DS210's lecture 28 by implementing a breadth-first search algorithm to determine the vertices of the components. This code was separated into a module titled connected_components.rs, where I defined Connected Components to be a structure with vectors for components and largest component vertices. I then implemented this structure to create a new graph with just the connected components of a given graph input. The new graph was then traversed and marked with BFS to determine the vertices of the connected components. Since I was interested in determining the largest connected component, I created a new method that returned a list of vertices of the largest component.

Having obtained my largest component, I proceeded to calculate its degree centrality in a new module titled centrality.rs. The formula for centrality within a graph is to calculate the sum of outgoing and incoming edges and divide this by the total number of nodes in the graph. I created a method that did just this and returned the centrality scores of all of the vertices in the largest component graph stored in a Hashmap. The values of the centrality scores must fall between the values of 0 and 1.

To bring it all together, my main.rs module read the CSV file, created a graph with the appropriate movie information, identified connected components of the graph, created another graph with the largest connected component vertices, and computed the centrality. My output is shown below.



```
Centrality Scores within Largest Component: {3888: 0.0, 737: 0.00020777062123415748, 1733: 0.00020777062123415748, 1090: 0.00020777062123415748, 4246: 0.0, 2228: 0.0002
0777062123415748, 3526: 0.0, 1764: 0.00020777062123415748, 2224: 0.00020777062123415748, 3001: 0.0, 3088: 0.0, 1803: 0.00020777062123415748, 3172: 0.0, 3362: 0.0, 4224:
 0.0, 2093: 0.00020777062123415748, 4095: 0.0, 4450: 0.0, 4658: 0.0, 339: 0.00020777062123415748, 4794: 0.0, 254: 0.00020777062123415748, 2773: 0.0, 1850: 0.00020777062
123415748, 2692: 0.0, 3559: 0.0, 462: 0.00020777062123415748, 1805: 0.00020777062123415748, 3797: 0.0, 1428: 0.00020777062123415748, 893: 0.00020777062123415748, 2941:
0.0, 1798: 0.00020777062123415748, 3182: 0.0, 1670: 0.00020777062123415748, 1129: 0.00020777062123415748, 375: 0.00020777062123415748, 1155: 0.00020777062123415748, 336
6: 0.0, 3524: 0.0, 891: 0.00020777062123415748, 3499: 0.0, 1780: 0.00020777062123415748, 3278: 0.0, 1524: 0.00020777062123415748, 3301: 0.0, 3507: 0.0, 1309: 0.00020777
062123415748, 2969: 0.0, 637: 0.00020777062123415748, 1372: 0.00020777062123415748, 408: 0.00020777062123415748, 3290: 0.0, 3497: 0.0, 1983: 0.00020777062123415748, 363
5: 0.0, 115: 0.00020777062123415748, 4651: 0.0, 2653: 0.0, 4186: 0.0, 178: 0.00020777062123415748, 3815: 0.0, 4346: 0.0, 3365: 0.0, 161: 0.00020777062123415748, 2837: 0
.0, 1245: 0.00020777062123415748, 2565: 0.0, 1727: 0.00020777062123415748, 319: 0.00020777062123415748, 1749: 0.00020777062123415748, 3100: 0.0, 826: 0.0002077706212341
5748, 1959: 0.00020777062123415748, 3480: 0.0, 2388: 0.00020777062123415748, 3168: 0.0, 2818: 0.0, 1207: 0.00020777062123415748, 1458: 0.00020777062123415748, 297: 0.00
020777062123415748, 730: 0.00020777062123415748, 1462: 0.00020777062123415748, 4290: 0.0, 1181: 0.00020777062123415748, 4596: 0.0, 3451: 0.0, 882: 0.0002077706212341574
8, 4811: 0.0, 993: 0.00020777062123415748, 1687: 0.00020777062123415748, 2262: 0.00020777062123415748, 4496: 0.0, 808: 0.00020777062123415748, 1211: 0.00020777062123415
748, 4422: 0.0, 1734: 0.00020777062123415748, 508: 0.00020777062123415748, 4149: 0.0, 4796: 0.0, 87: 0.00020777062123415748, 585: 0.00020777062123415748, 1415: 0.000207
77062123415748, 3043: 0.0, 1844: 0.00020777062123415748, 1027: 0.00020777062123415748, 4259: 0.0, 2957: 0.0, 1549: 0.00020777062123415748, 995: 0.00020777062123415748,
1544: 0.00020777062123415748, 2188: 0.00020777062123415748, 3570: 0.0, 2195: 0.00020777062123415748, 4233: 0.0, 2562: 0.0, 2623: 0.0, 2281: 0.00020777062123415748, 3020
: 0.0, 3064: 0.0, 2158: 0.00020777062123415748, 1120: 0.00020777062123415748, 661: 0.00020777062123415748, 1417: 0.00020777062123415748, 3778: 0.0, 1103: 0.000207770621
23415748, 3396: 0.0, 1140: 0.00020777062123415748, 1043: 0.00020777062123415748, 652: 0.00020777062123415748, 1263: 0.00020777062123415748, 4519: 0.0, 1329: 0.000207770
62123415748, 563: 0.00020777062123415748, 4055: 0.0, 3641: 0.0, 1756: 0.00020777062123415748, 374: 0.00020777062123415748, 3862: 0.0, 314: 0.00020777062123415748, 1955:
 0.00020777062123415748, 3389: 0.0, 3993: 0.0, 1540: 0.00020777062123415748, 1416: 0.00020777062123415748, 2433: 0.0, 2962: 0.0, 4025: 0.0, 1057: 0.00020777062123415748
, 2330: 0.00020777062123415748, 733: 0.00020777062123415748, 1571: 0.00020777062123415748, 237: 0.00020777062123415748, 2713: 0.0, 3319: 0.0, 3504: 0.0, 3903: 0.0, 3977
: 0.0, 1656: 0.00020777062123415748, 4109: 0.0, 8: 0.00020777062123415748, 2165: 0.00020777062123415748, 4756: 0.0, 4812: 0.0, 602: 0.00020777062123415748, 1179: 0.0002
0777062123415748, 927: 0.00020777062123415748, 879: 0.00020777062123415748, 2805: 0.0, 2925: 0.0, 3934: 0.0, 4341: 0.0, 1433: 0.00020777062123415748, 3063: 0.0, 3279: 0
.0, 4593: 0.0, 4615: 0.0, 490: 0.00020777062123415748, 2180: 0.00020777062123415748, 2579: 0.0, 101: 0.00020777062123415748, 1012: 0.00020777062123415748, 435: 0.000207
77062123415748, 3360: 0.0, 4460: 0.0, 4727: 0.0, 1931: 0.00020777062123415748, 1323: 0.00020777062123415748, 1232: 0.00020777062123415748, 2160: 0.00020777062123415748,
 3545: 0.0, 4064: 0.0, 1908: 0.00020777062123415748, 4151: 0.0, 610: 0.00020777062123415748, 4400: 0.0, 2240: 0.00020777062123415748, 3806: 0.0, 4037: 0.0, 4619: 0.0, 2
032: 0.00020777062123415748, 3390: 0.0, 3536: 0.0, 4284: 0.0, 1838: 0.00020777062123415748, 152: 0.00020777062123415748, 1190: 0.00020777062123415748, 82: 0.00020777062
123415748, 2513: 0.0, 3031: 0.0, 473: 0.00020777062123415748, 4264: 0.0, 4549: 0.0, 2337: 0.00020777062123415748, 1062: 0.00020777062123415748, 1638: 0.0002077706212341
5748, 2267: 0.00020777062123415748, 1424: 0.00020777062123415748, 2555: 0.0, 442: 0.00020777062123415748, 1175: 0.00020777062123415748, 2808: 0.0, 2986: 0.0, 1671: 0.00
020777062123415748, 2173: 0.00020777062123415748, 3285: 0.0, 3049: 0.0, 2143: 0.00020777062123415748, 1254: 0.00020777062123415748, 1813: 0.00020777062123415748, 3444:
0.0, 1461: 0.00020777062123415748, 253: 0.00020777062123415748, 405: 0.00020777062123415748, 3092: 0.0, 4345: 0.0, 856: 0.00020777062123415748, 2796: 0.0, 4278: 0.0, 31
43: 0.0, 1543: 0.00020777062123415748, 633: 0.00020777062123415748, 627: 0.00020777062123415748, 918: 0.00020777062123415748, 744: 0.00020777062123415748, 2585: 0.0, 27
68: 0.0, 3423: 0.0, 1509: 0.00020777062123415748, 552: 0.00020777062123415748, 4579: 0.0, 1702: 0.00020777062123415748, 2116: 0.00020777062123415748, 4706: 0.0, 197: 0.
00020777062123415748, 2635: 0.0, 3441: 0.0, 2601: 0.0, 1166: 0.00020777062123415748, 4211: 0.0, 1486: 0.00020777062123415748, 708: 0.00020777062123415748, 3731: 0.0, 34
49: 0.0, 1160: 0.00020777062123415748, 3824: 0.0, 701: 0.00020777062123415748, 1903: 0.00020777062123415748, 3628: 0.0, 304: 0.00020777062123415748, 4636: 0.0, 4670: 0.
0, 2952: 0.0, 888: 0.00020777062123415748, 1951: 0.00020777062123415748, 2089: 0.00020777062123415748, 3045: 0.0, 2906: 0.0, 2166: 0.00020777062123415748, 406: 0.000207
77062123415748, 827: 0.00020777062123415748, 3105: 0.0, 1787: 0.00020777062123415748, 2429: 0.0, 2724: 0.0, 4097: 0.0, 988: 0.00020777062123415748, 3076: 0.0, 3107: 0.0
, 2522: 0.0, 4616: 0.0, 2030: 0.00020777062123415748, 1074: 0.00020777062123415748, 2377: 0.00020777062123415748, 3317: 0.0, 3825: 0.0, 1029: 0.00020777062123415748, 59
5: 0.00020777062123415748, 4044: 0.0, 2199: 0.00020777062123415748, 3687: 0.0, 1688: 0.00020777062123415748, 4364: 0.0, 1427: 0.00020777062123415748, 4256: 0.0, 924: 0.
00020777062123415748, 2004: 0.00020777062123415748, 2358: 0.00020777062123415748, 1930: 0.00020777062123415748, 2496: 0.0, 1038: 0.00020777062123415748, 3315: 0.0, 3546
: 0.0, 4081: 0.0, 2573: 0.0, 2666: 0.0, 2403: 0.00020777062123415748, 3115: 0.0, 1503: 0.00020777062123415748, 144: 0.00020777062123415748, 716: 0.00020777062123415748,
 1853: 0.00020777062123415748, 4140: 0.0, 1502: 0.00020777062123415748, 1760: 0.00020777062123415748, 1528: 0.00020777062123415748, 1538: 0.00020777062123415748, 3273:
0.0, 2329: 0.00020777062123415748, 1357: 0.00020777062123415748, 235: 0.00020777062123415748, 2503: 0.0, 2789: 0.0, 3619: 0.0, 1804: 0.00020777062123415748, 1082: 0.000
20777062123415748, 3620: 0.0, 256: 0.00020777062123415748, 244: 0.00020777062123415748, 2316: 0.00020777062123415748, 23: 0.00020777062123415748, 4577: 0.0, 2139: 0.000
20777062123415748, 1690: 0.00020777062123415748, 864: 0.00020777062123415748, 3226: 0.0, 1248: 0.00020777062123415748, 2364: 0.00020777062123415748, 1050: 0.00020777062
123415748, 1911: 0.00020777062123415748, 3407: 0.0, 2146: 0.00020777062123415748, 2841: 0.0, 3426: 0.0, 3968: 0.0, 1917: 0.00020777062123415748, 301: 0.0002077706212341
5748, 3866: 0.0, 4483: 0.0, 1642: 0.00020777062123415748, 4031: 0.0, 4728: 0.0, 231: 0.00020777062123415748, 4229: 0.0, 4201: 0.0, 1779: 0.00020777062123415748, 4116: 0
.0, 1153: 0.00020777062123415748, 2293: 0.00020777062123415748, 1192: 0.00020777062123415748, 1697: 0.00020777062123415748, 4693: 0.0, 1197: 0.00020777062123415748, 668
```

Having integrated through all 4814 nodes of the largest component graph, the algorithm outputs the centrality score for each vertex. Interestingly, the centrality scores only yielded two values, 0.0 and 0.00020777062123415748. This low value suggests that there is a weak relationship between composers and genres they tend to prefer in the film industry. In conclusion, the low centrality scores, primarily concentrated at 0.0 and 0.00020777062123415748, suggest a weak overall relationship between composers and the genres they tend to prefer in the Telugu film industry. Furthermore, the prevalence of 0.0 scores within connected components implies that certain composers and genres within these subsets lack significant relationships with others. To test whether all the data followed this unique pattern, I created a test function that tested whether every value was either 0.0 or 0.00020777062123415748. It passed with flying colors.

Sources:

- Lecture 28
- Degree Centrality Formula