Amazon Co-purchasing Network Analysis

The project uses the Amazon Product Co-purchasing network dataset to analyze the relationship between each product and their similarity. The first thing is writing a read file function, which reads the data set that contains pairs of size values separated by whitespace and returns them as a vector of tuples. Second, I am trying to find communities, in which I use Tarjan's strongly connected components (SCC) algorithm for detecting communities in an undirected graph. I use the **petgraph** crate's UnGraph and **tarjan_scc** algorithm to find strongly connected components. The algorithm uses a depth-first search with a stack to traverse the graph and identify strongly connected components. The result that I get contains vectors of node indices representing the strongly connected components in the graph. In friend analysis.rs, there are three different functions. The subgraph function takes a vector of edges and a vector of node indices representing a community. It creates an undirected graph (UnGraph) from the input edges, converting node indices to NodeIndex. The calculate_product_similarity Function initializes a HashMap (product_similarity) to store the product similarities between nodes in the community and then iterates over each pair of nodes in the community and calculates the Jaccard similarity based on common neighbors. The analyze_friend_of_friend function takes a vector of edges (edges), a vector of node indices representing a community (community), and a HashMap of product similarities and initializes a HashMap (common_customers) to store the count of common customers between pairs of nodes. Then the function iterates over pairs of nodes in the community and counts the common neighbors. Then, the results are stored in the common_customers HashMap and sorted in descending order based on the count of common customers.

The main function reads the amazon.txt file and stores the result in the edges. It creates a subset of edges (edges_subset) by cloning and taking the first subset_size elements from the edges vector. It uses the find_communities function to detect communities in the graph represented by edges. Then iterates over the detected communities, printing details for each community using the print_community_details function. After that, for each community, it calculates product similarity using the calculate_product_similarity function and analyzes friend-of-friend relationships using the analyze_friend_of_friend function.

Output:

Component 0: [0, 10, 1035, 355, 155, 2838, 4161, 1528, 1705, 16666, 71407, 191581, 26607, 26608, 26606, 122385, 158520, 159994, 307320, 348286, 348285, 381167, 154530, 154538, 163073, 163072, 154537, 154535, 154534, 154533, 154536, 154532, 154531, 151148, 1744, 1946, 1743, 11379, 14862, 44489, 122582, 201211, 224601, 237491, 224600, 224602, 241996, 241995, 304687, 354118, 161233, 247296, 328247]

Friend-of-Friend Analysis in Community 0:

Pair: (10, 15), Common Customers: 1, Jaccard Similarity: 0.25

Pair: (9, 10), Common Customers: 1, Jaccard Similarity: 0.25

Pair: (15, 10), Common Customers: 1, Jaccard Similarity: 0.25

Pair: (0, 10), Common Customers: 1, Jaccard Similarity: 0.25

Pair: (10, 9), Common Customers: 1, Jaccard Similarity: 0.25

Component 1: [1, 3943, 5919, 102346, 121904, 74677, 121903, 93657, 223249, 266551, 326328, 337155, 164250, 229240, 323666, 270056, 270057, 283097, 316322, 316321, 316320, 316319, 316318, 303666, 233742, 344550, 357993, 357992, 46, 112, 184, 2515, 2508, 6384, 6383, 9779, 59033, 150127, 249195, 264908, 265017, 265016, 265015, 339300, 382855, 381334, 374958, 374836, 380743, 383216, 374835, 374957, 374956, 374834, 374833, 374832, 374831, 374829, 366242, 374830, 279778, 249194, 279779, 330965, 356453, 330960, 330963, 347789, 341531, 341532, 340106, 340102, 341530, 341529, 274201, 350779, 368989, 389904, 368988, 387655, 395772, 158281, 294325, 294327, 296116, 295161, 295088, 294463, 294462, 294326, 294324, 294323, 272153, 265700, 158282, 158279, 158278, 70994, 268661, 351112, 384444, 384442, 384443, 384441, 384440, 158276, 245516, 143445, 143447, 174377, 199205, 265014, 339101, 363826, 382801, 380143, 380142, 390808]

Friend-of-Friend Analysis in Community 1:

Pair: (3, 3), Common Customers: 0, Jaccard Similarity: 0.00

Component 2: [2, 118, 189, 246, 1139, 28933, 65061, 17769, 29547, 29549, 39338, 112012, 267536, 290015, 290014, 290013, 290012, 37765, 112011, 244, 13798, 14307, 14306, 164877, 192186, 297436, 239839, 192516, 299996, 358377, 376108, 358376, 358375, 382109, 382108, 382110, 374624, 196032, 239100, 346833, 5, 53, 117, 28927, 257211, 257210, 252125, 185176, 217640, 251126, 324318, 324317, 324316, 324315, 301859, 301856, 315723, 315721, 363049, 363048, 87730, 149745, 261331, 149748, 110280, 211247, 159748, 203100, 282415]

Friend-of-Friend Analysis in Community 2:

Pair: (0, 1), Common Customers: 1, Jaccard Similarity: 0.17

Pair: (1, 0), Common Customers: 1, Jaccard Similarity: 0.17

Pair: (1, 1), Common Customers: 0, Jaccard Similarity: 0.00
Pair: (0, 0), Common Customers: 0, Jaccard Similarity: 0.00
Component 3: [3, 4955, 5543, 12774, 5542, 235, 12469, 33305, 37283, 67779, 57111, 57112, 30748, 30747, 65962, 119130, 255974, 97377, 130690, 222099, 284920, 284923, 339512, 284922, 284921, 250684, 248898, 179320, 248899, 269631, 277644, 333008, 277645, 179919, 146498, 179918, 185783, 45, 1045, 2749, 10308, 23576, 33358, 96718, 113673, 272420, 272421, 272419, 139220, 138470, 304082, 284768, 80408, 113672, 232536, 144707, 221697, 226967, 226968, 144706, 188175, 143361, 143360, 131839, 143359, 131838, 128263, 131840, 126350, 126348, 126349, 80409, 89150, 89148, 65243, 35046, 80410, 190176, 232539, 264512, 312335, 317961, 365521, 380508, 380510, 338643, 380509, 312334, 312333, 239317, 264510, 232537, 232535, 157495, 157493, 159014, 157494, 157492, 157491, 91450, 70047, 54216, 54214, 54215, 54211, 54213, 54217, 54212, 54210, 54209, 22476, 22474, 396077, 379431, 340727, 340726, 54690, 93469, 93466, 94501, 299724, 93468, 93467, 93465, 86012, 211877, 60220, 12111, 29620, 29619, 78105, 78103, 163554, 134383, 177156, 177157, 201194, 194816, 194815, 194814, 215123, 334160, 357987, 348499, 348501, 348500, 245966, 327837, 331580, 358872, 212981, 225591, 307336, 225590, 368598, 368597, 10977, 45527, 153609, 191424, 246939, 115366, 259164, 292405, 293433, 292406, 292404, 293434, 356842, 389565, 389606, 397707]
Friend-of-Friend Analysis in Community 3:
Pair: (8, 8), Common Customers: 0, Jaccard Similarity: 0.00
Pair: (2, 8), Common Customers: 0, Jaccard Similarity: 0.00
Pair: (8, 2), Common Customers: 0, Jaccard Similarity: 0.00
Pair: (2, 2), Common Customers: 0, Jaccard Similarity: 0.00
Component 4: [4, 1033, 1522, 2341, 13740, 106134, 280396, 311622, 311621, 311620, 311619, 17498, 109014, 297567, 359577, 365409, 365758, 371005, 54, 1053, 2750, 45481, 1961, 14013, 52348, 23398, 23399, 37647, 101001, 174415, 211002, 211001, 245772, 245770, 333940, 110039, 110048, 185528, 164345, 164349, 173455, 237334, 262586, 262587, 171655, 170099, 170103, 193792, 331515, 371999, 371998, 371997, 371996, 360352, 279368, 331509, 331514, 331513, 331512, 331511, 331510, 64960, 125883, 113460, 166372, 215924, 215926, 215927, 218683, 296256, 324169, 183905, 131886, 73210, 95575, 95576, 73209, 73208, 73206, 54589, 54588, 54587, 54586, 73207, 54585, 54584, 54583, 324167, 324166, 357876, 245426, 230072, 328401]
…..
In component 0: the list of number are node indices that belong to this community.
For each pair of nodes, the output provides information such as:

- Pair: (node1, node2): The indices of two nodes in the community.

- Common Customers: The number of common customers (neighbors) between the two nodes.
- Jaccard Similarity: The Jaccard similarity coefficient is calculated based on the common neighbors.