


The results of the five graph analytics tasks make logical sense and provide meaningful structural insights into the ego network:

(a) Top 5 Nodes with Highest Outdegree:



The screenshot shows a Databricks notebook interface. At the top, there's a status bar with a green checkmark, the time '05:48 PM (2s)', and the page number '18'. Below this, a code cell contains the following Spark SQL query:

```
# Sort the nodes by outdegree (count of outgoing edges) in descending order and take top 5
top5_out = out_degrees.orderBy(col("outDegree").desc())
top5_out.show(5)
```

Below the code cell, the 'Spark Jobs' section shows a job with 2 tasks. The first task is expanded, showing the execution of the query. The output is a table with 2 columns: 'id' and 'outDegree'. The table contains 5 rows of data, which are the top 5 nodes by outdegree. The output is displayed in a table format with a header row and 5 data rows. The text 'only showing top 5 rows' is visible at the bottom of the output.

id	outDegree
56	154
67	150
271	144
322	142
25	136

only showing top 5 rows

- Nodes with the highest outdegree represent users who initiate the most connections (e.g., they follow many others in directed networks).
- This makes sense for social platforms like Twitter or G+ where following is one-way.
- **Insight:** These users are likely to be active explorers or information seekers.

(b) Top 5 Nodes with Highest Indegree

```
05:48 PM (1s) 21
# Sort the nodes by indegree (count of incoming edges) in descending order and take top 5
top5_in = in_degrees.orderBy(col("inDegree").desc())
top5_in.show(5)

(2) Spark Jobs
top5_in: pyspark.sql.dataframe.DataFrame = [id: string, inDegree: integer]
+---+-----+
| id|inDegree|
+---+-----+
| 56|    154|
| 67|    150|
|271|    144|
|322|    142|
| 25|    136|
+---+-----+
only showing top 5 rows
```

- Nodes with the highest indegree are the most followed users.
- These are likely to be **influential or popular accounts**, attracting many connections.
- **Insight:** High indegree is often a good proxy for social influence or authority in the network.

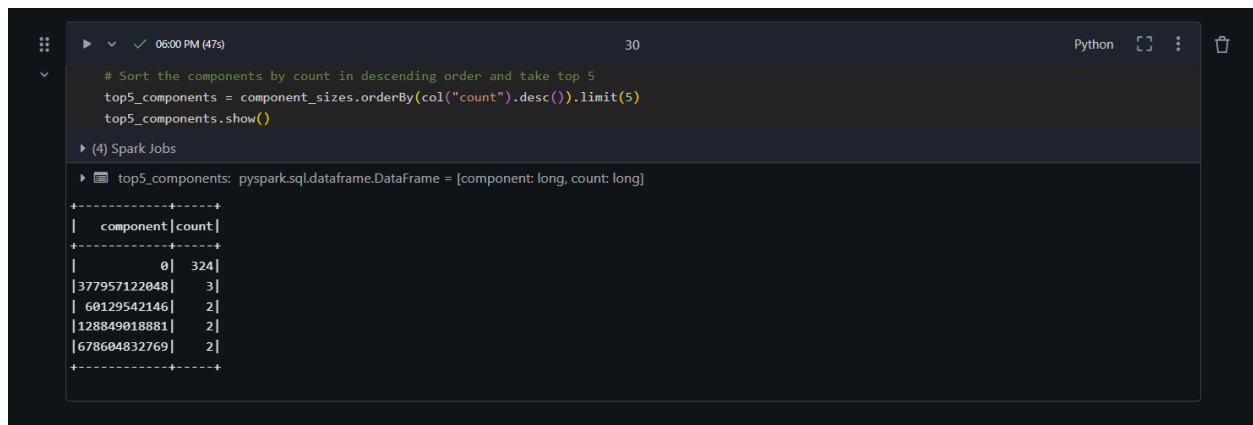
(c) Calculate PageRank and Find Top 5 Nodes

```
05:55 PM (20s) 25
# Extract PageRank scores and sort descending to get the top 5 nodes
top5_pageRank = results.vertices.orderBy(col("pagerank").desc()).limit(5)
top5_pageRank.show()

▶ (3) Spark Jobs
▶ top5_pageRank: pyspark.sql.dataframe.DataFrame = [id: string, pagerank: double]
+-----+
| id | pagerank |
+-----+
| 25 | 4.366870522474023 |
| 23 | 3.88392857142857 |
| 119 | 3.6705112438163883 |
| 322 | 3.527481311957077 |
| 56 | 3.263870320431853 |
+-----+
```

- PageRank highlights not just how many connections a node has, but also how important those connections are.
- The top-ranked nodes are well-connected to **other well-connected nodes**.
- **Insight:** These are central and influential users in the network, potentially acting as key bridges or hubs of information flow.

(d) Connected Components: Top 5 Largest Components



```
# Sort the components by count in descending order and take top 5
top5_components = component_sizes.orderBy(col("count").desc()).limit(5)
top5_components.show()
```

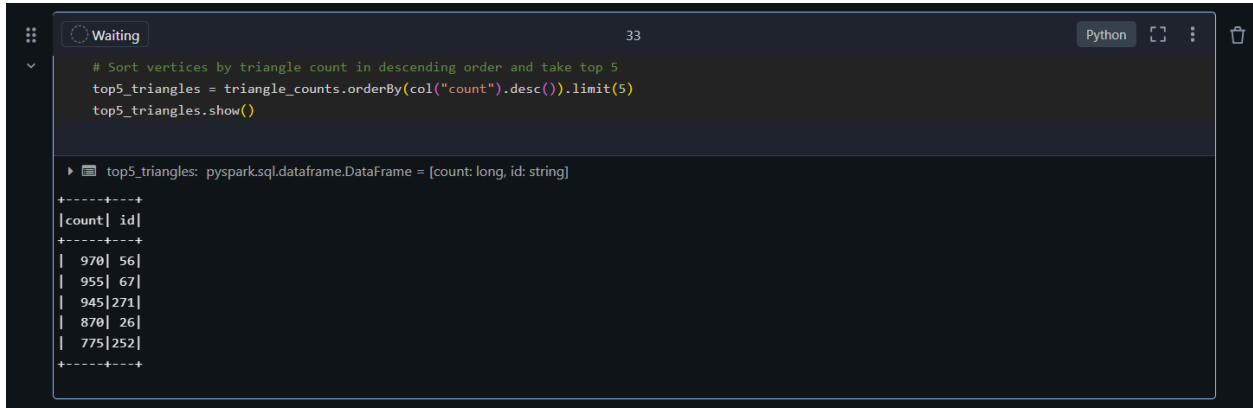
▶ (4) Spark Jobs

▶ top5_components: pyspark.sql.dataframe.DataFrame = [component: long, count: long]

component	count
0	324
377957122048	3
60129542146	2
128849018881	2
678604832769	2

- Each connected component is a cluster of users where every user is reachable from every other.
- The largest components likely represent **tight-knit communities or friend groups**.
- **Insight:** Ego networks often contain multiple clusters — some users may form isolated cliques, while others belong to large, interconnected groups.

(e) Triangle Counts: Top 5 Vertices by Triangle Count



```
# Sort vertices by triangle count in descending order and take top 5
top5_triangles = triangle_counts.orderBy(col("count").desc()).limit(5)
top5_triangles.show()
```

top5_triangles: pyspark.sql.dataframe.DataFrame = [count: long, id: string]

count	id
970	56
955	67
945	271
870	26
775	252

- Triangle count measures how interconnected a user's friends are (i.e., how many "friend-of-a-friend" connections exist).
- High triangle counts indicate **dense local communities**.
- **Insight:** These users likely belong to social circles where mutual friendships are common — good candidates for community detection or circle inference.