

INFORMATION RETRIEVAL ASSIGNMENT 3

Team members:

Shraman Jain (MT22068)

Arshin Jain (MT22094)

Ronit Aryan (2020324)

Q1 Link Analysis

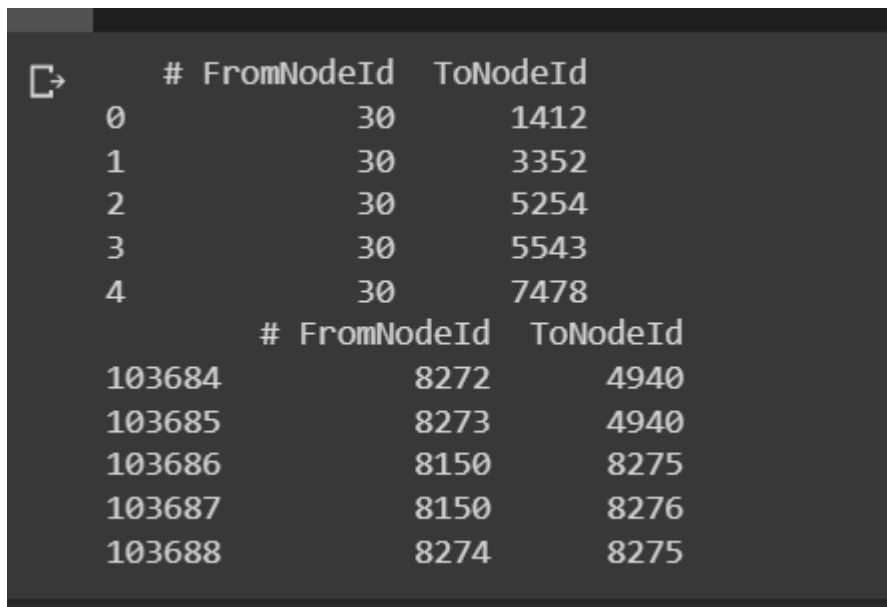
Dataset chosen: Wikipedia Vote Network

Nodes in the network represent Wikipedia users and a directed edge from node i to node j represents that user i voted on user j .

Nodes in the network: Wikipedia users

Edges in the network: user i voted on user j

Dataset was downloaded as a tab-separated txt file; we iterated over each line.



```

# FromNodeId ToNodeId
0 30 1412
1 30 3352
2 30 5254
3 30 5543
4 30 7478
# FromNodeId ToNodeId
103684 8272 4940
103685 8273 4940
103686 8150 8275
103687 8150 8276
103688 8274 8275
```

- **Adjacency Matrix:** First, create a zero-filled matrix, then append it if there is an edge from node 1 to 2, then the `adj_matrix[1][2]=1`; else, it will be 0. In the adjacency matrix, we have stored all the unique nodes.

```
> [[0 0 0 ... 0 0 0]
   [1 0 0 ... 0 0 0]
   [1 1 0 ... 0 0 0]
   ...
   [0 0 0 ... 0 0 0]
   [0 0 0 ... 0 0 0]
   [0 0 0 ... 0 0 0]]
```

- **Edge list:**

```
[(30, 1412), (30, 3352), (30, 5254), (30, 5543), (30, 7478), (3, 28),
(3, 30), (3, 39), (3, 54), (3, 108), (3, 152), (3, 178), (3, 182), (3,
214), (3, 271), (3, 286), (3, 300), (3, 348), (3, 349), (3, 371), (3,
567), (3, 581), (3, 584), (3, 586), (3, 590), (3, 604), (3, 611), (3,
8283), (25, 3), (25, 6), (25, 8), (25, 19), (25, 23), (25, 28), (25,
29), (25, 30), (25, 33), (25, 35), (25, 50), (25, 54), (25, 55), (25,
75), (25, 80),
.....(686, 1533),
(686, 1534), (686, 1538), (686, 1600), (686, 1603), (686, 1658), (686,
1717), (686, 1805), (686, 1833), (686, 8265, 7238), (8266, 7238),
(7637, 7833), (8270, 4940), (8270, 7833), (8271, 7833), (8272, 4940),
(8273, 4940), (8150, 8275), (8150, 8276), (8274, 8275)]
103689
```

Number of Nodes	7115
Number of Edges	103689
Avg In-degree	14.573295853829936
Avg Out-degree	14.573295853829936
Node with Max In-degree	4037
Node with Max Out-degree	2565
The density of the network	0.0020485375110809584

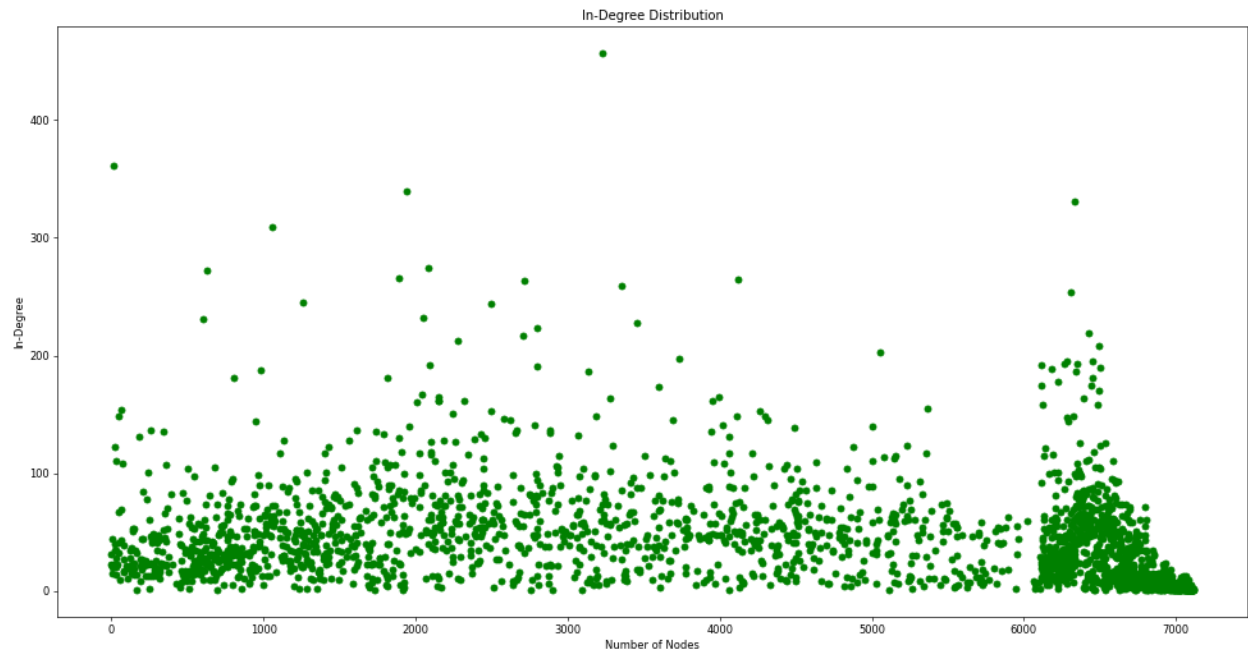
Avg. In-Degree = Sum of the number of incoming edges of every node / total number of nodes

Avg. Out-Degree = Sum of the number of outgoing edges of every node / total number of nodes

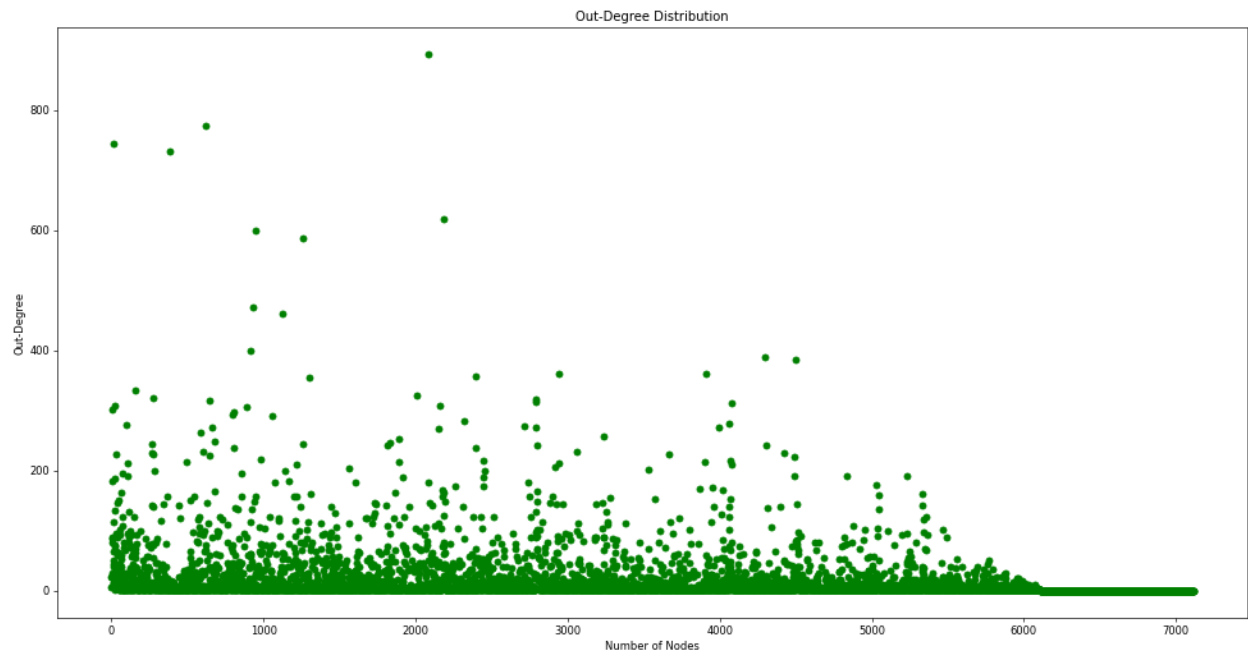
Network Density (D) = $E / (N * (N - 1))$ for directed networks; E = a number of edges, N = a number of nodes

Plotting degree distribution of the network for a directed graph, plot in-degree, and out-degree
Since the graph is a directed graph, we have plotted the scatted graph for both in-degree and out-degree distribution.

In-degree distribution of network:



Out-degree distribution of network:



The local clustering coefficient of each node:

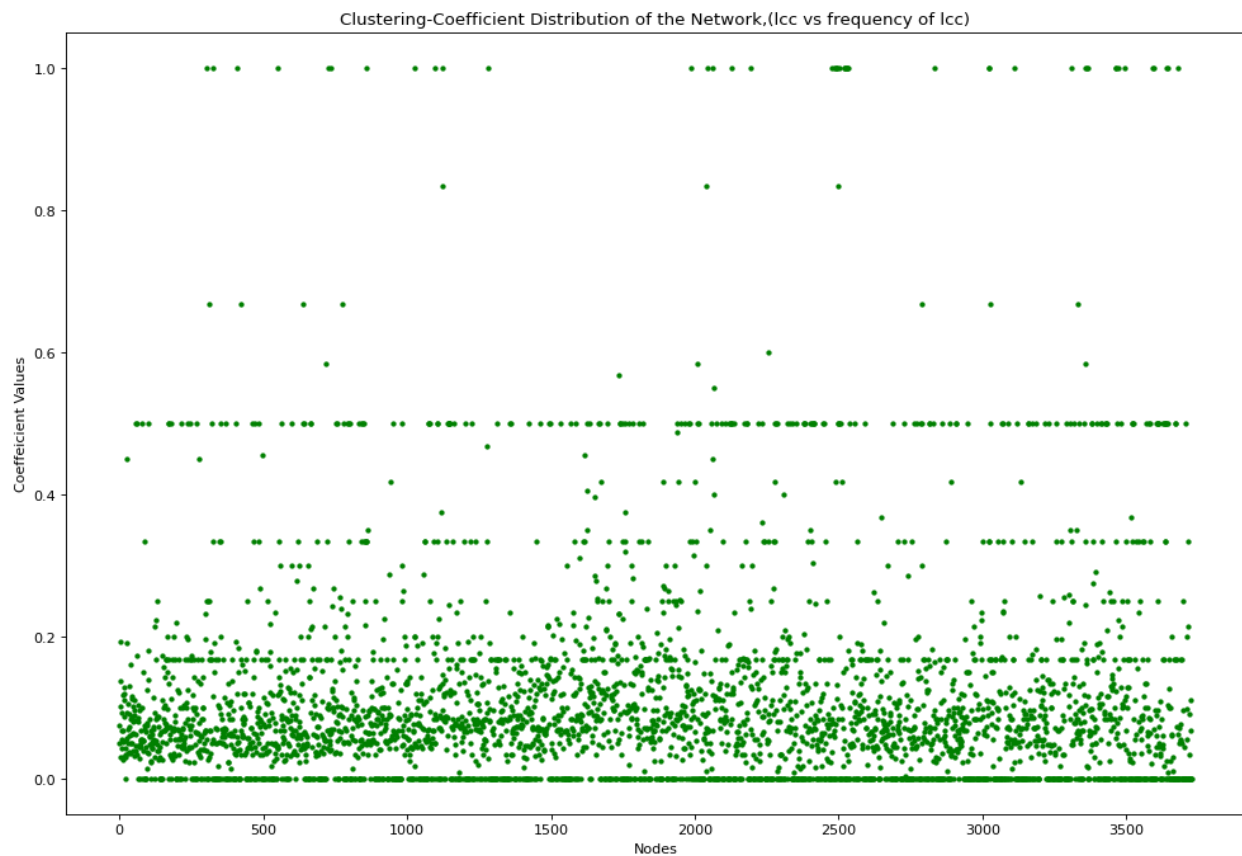
Local Clustering Coefficient of each node = Number of pairs of neighbors of the node that are connected / number of pairs of neighbors of the node

The clustering coefficient lies between 0 and 1. The clustering coefficient is more skewed towards 1, giving higher certainty.

The overall clustering coefficient of the network:

```
[0.05, 0.07509881422924901, 0.10074906367041199, 0.13669950738916256, 0.191699604743083, 0.030659391432531737, 0.0923913043478]
```

Plotting the clustering-coefficient distribution (lcc vs frequency of lcc) of the network.



Ques -2

In this report, we will compare the Pagerank and the HITS algorithms, discussing their similarities and differences, strengths, weaknesses, and application areas.

PageRank

The Pagerank algorithm evaluates the web's link structure and rates each page according to the quantity and quality of links. Higher-scoring pages are considered more reliable and more likely to appear at the top of search engine results pages.

Pros

For large-scale search engines like Google, Pagerank is a very successful technique for ranking web pages. It considers both the quantity and quality of inbound links that point to a web page, which can be used to spot pages with a lot of authority and relevance.

Cons

One of Pagerank's flaws is the absence of consideration for a web page's content. This implies that web pages with high Pagerank ratings may not always be the most pertinent to a user's search query. Additionally, link schemes and other spammy tactics can be used to manipulate Pagerank, resulting in falsely inflated ranks.

Hits

HITS (Hyperlink-Induced Topic Search) is another algorithm used in web search engines to rank web pages based on their relevance and authority. The HITS algorithm works by analyzing the link structure of the web and identifying pages that are both authoritative and relevant to a user's search query.

Pros

The HITS algorithm's ability to consider both a web page's content and link structure is one of its advantages. Accordingly, websites with high HITS scores are more likely to be reputable and pertinent to a user's search query. As it uses both inward and outbound connections to assess a page's authority, HITS is thus less prone to manipulation than Pagerank.

Cons

HITS can be computationally expensive, one of its drawbacks, especially for large-scale search engines. Furthermore, HITS may be vulnerable to spam tactics and link schemes, which may have a negative effect on the authority and relevancy of search results.

Comparison

The time taken for evaluation the scores in HITS algorithm is greater than the time taken in evaluating the scores in Pagerank algorithm. As the HITS creates mutual reinforcement between authority and hub scores and page rank just do it on the basis of authority, the HITS results are less relevance than the page rank scores. This popularity is due to the features like efficiency, feasibility, less query time cost, etc. which are absent in HITS algorithm.

Top results returned by both algorithms

1. PageRank Score

```
✓ 0s ▶ print(pageRankNew[:5])  
[ (4037, 0.0046127158911675415), (15, 0.00368122072952927), (6634, 0.003524813657640256), (2625, 0.003286374369230901), (2398, 0.0026053331717250175)]
```

2. HubScore Rank

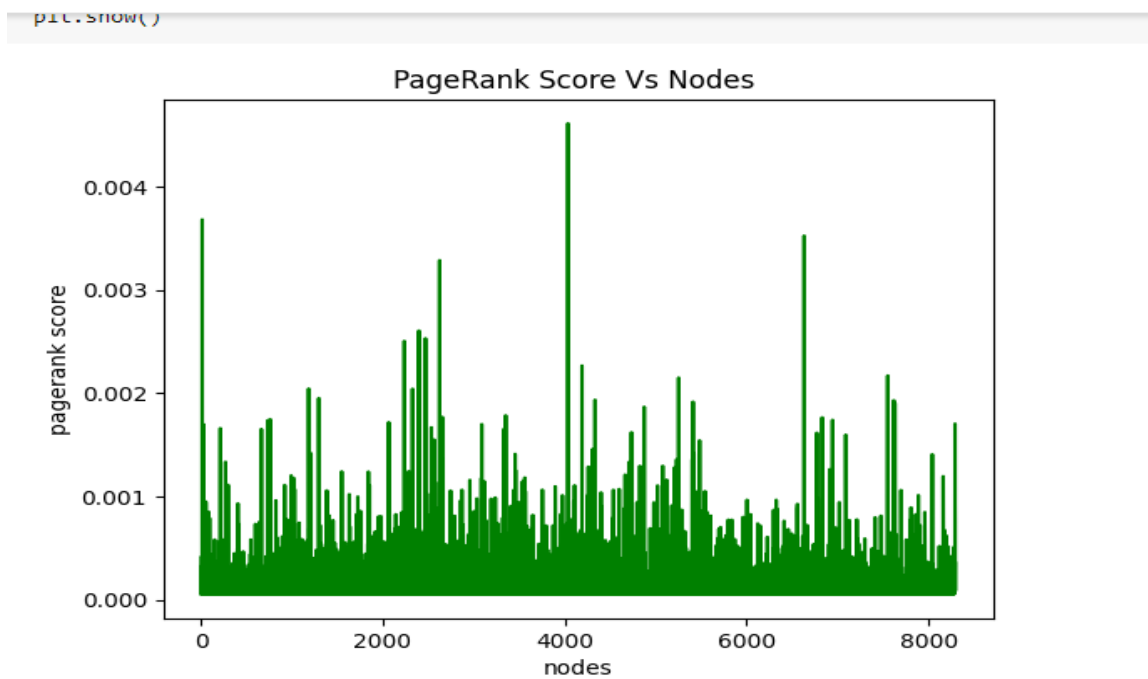
```
✓ 0s [65] print(hubScoreNew[:5])  
[(2565, 0.007940492708143138), (766, 0.007574335297501249), (2688, 0.006440248991029867), (457, 0.006416870490261067), (1166, 0.006010567902411199)]
```

3. AuthScore Rank

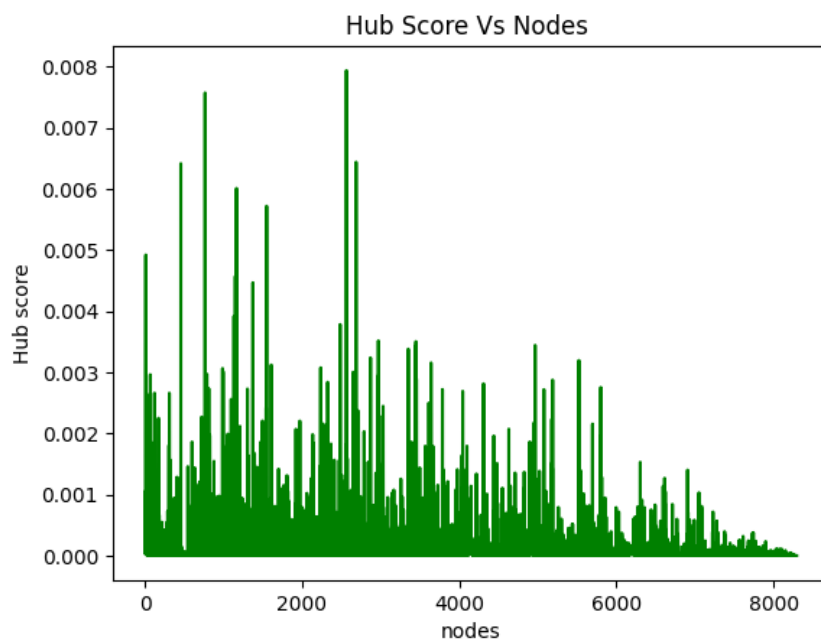
```
✓ 0s [67] print(authScoreNew[:5])  
[(2398, 0.0025801471780088773), (4037, 0.0025732411242297996), (3352, 0.002328415091497685), (1549, 0.00230373148045718), (762, 0.002255874856287141)]
```

Plotting the graph

1. PageRank vs Nodes

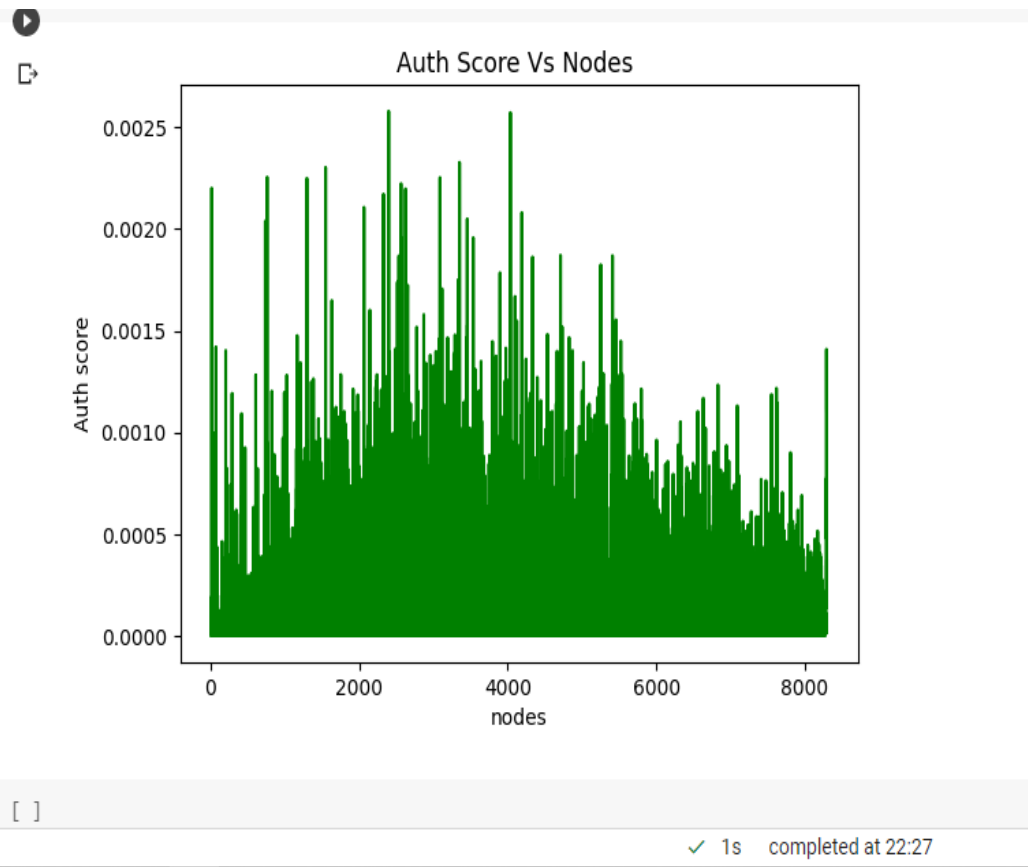


2. HubScore vs Nodes

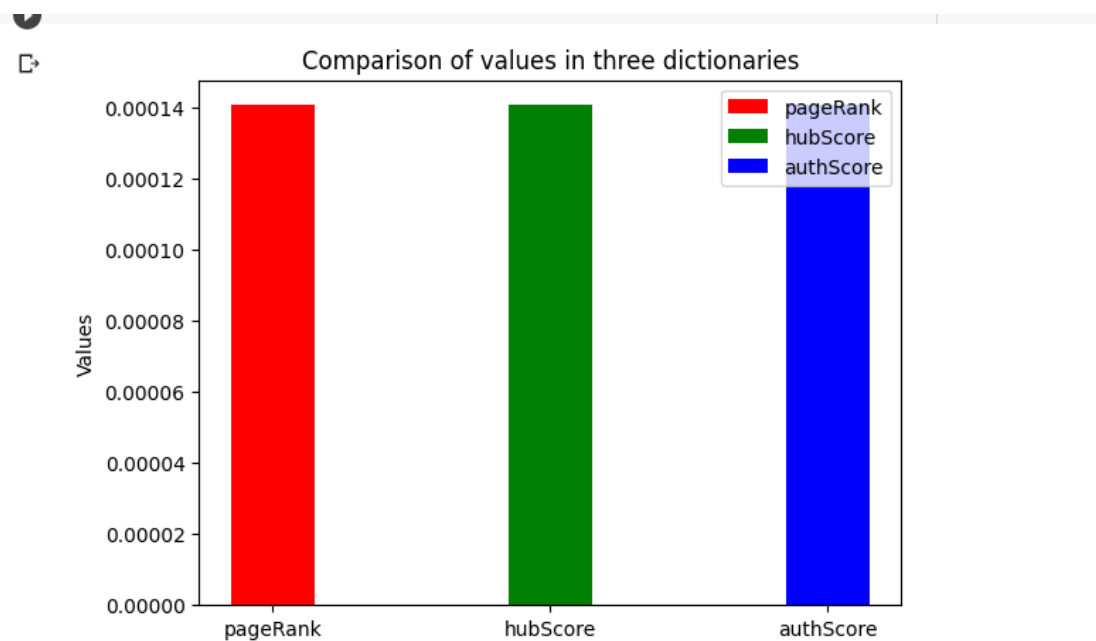


✓ 1s completed at 22:27

3. AuthScore vs Nodes



4. Plotting the average value of all the three scores



5. Comparison of top 5 values from eac of the score dictionaries.

