Q2- Report

Problem Statement : To determine the most relevant sites in the provided dataset, by applying the PageRank and authority-hub algorithms.

Dataset: Wikivote data, which consists of 7115 nodes and 103689 edges, was utilised for this.

Graph : After importing the provided data using Python's standard file reading function, we divide it into informational and graphical components. The graph connection is in the Graph section, while the Information component offers details and statistics about the data in the variable ‘info’ .

We use Python networkx package, to create graph using dataframe. In order to generate the graph and turn it into a directed graph, we implement the nx.from_pandas_edgelist method.

PageRank :
We initially take equal probabilities of all nodes to be most relevant page. We then initialized a loop for all nodes in which we recalculate the probability of each node by dividing its current probability by the no. of outgoing edges.This value is added to all nodes having 0 score.

$P(a) = \Sigma p(i)/O(i)$
$O(i)$ - number of outgoing edges
$p(i)$ - current pagerank score.

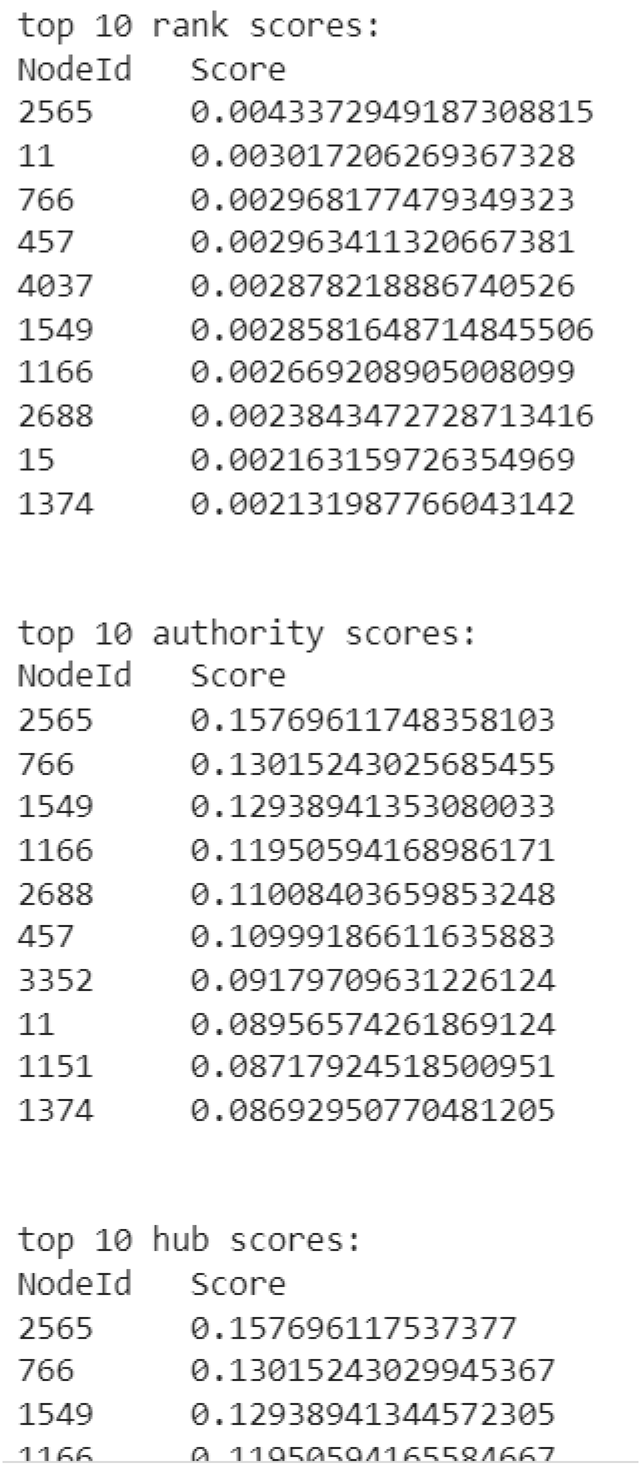
For convergence criteria we use mean squared error and set threshold e=1e-15.

Authority and Hub :
Authority score is for incoming edges while hub score is for outgoing edges.
We take all hub and authority scores = 1 .
We then keep a list of authority scores and hub scores as previous pointers.

Formula :
authority scores : $\text{auth}(p) = \boldsymbol{\Sigma}\ \text{hub}(q)$
hub scores : $\text{hub}(p) = \boldsymbol{\Sigma}\ \text{auth}(q)$

The buffer is updated using MSE errors. Threshold = 1e-20 and no of iter = 100 (max).

Results :



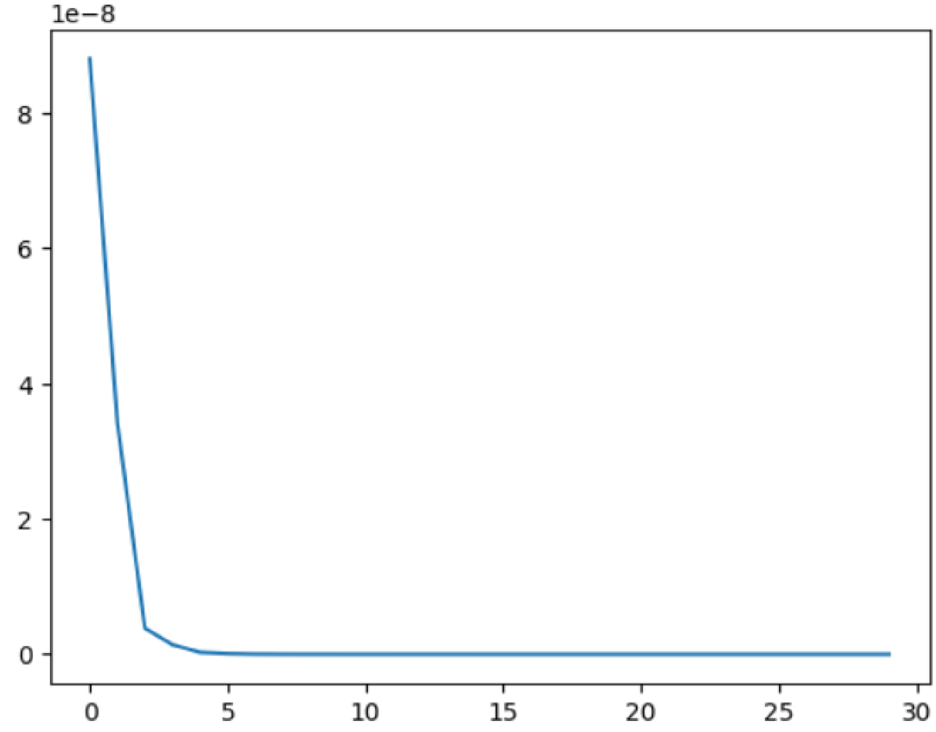
```
top 10 rank scores:
NodeId   Score
2565     0.0043372949187308815
11       0.003017206269367328
766      0.002968177479349323
457      0.002963411320667381
4037     0.002878218886740526
1549     0.0028581648714845506
1166     0.002669208905008099
2688     0.0023843472728713416
15       0.002163159726354969
1374     0.002131987766043142


top 10 authority scores:
NodeId   Score
2565     0.15769611748358103
766      0.13015243025685455
1549     0.12938941353080033
1166     0.11950594168986171
2688     0.11008403659853248
457      0.10999186611635883
3352     0.09179709631226124
11       0.08956574261869124
1151     0.08717924518500951
1374     0.08692950770481205


top 10 hub scores:
NodeId   Score
2565     0.157696117537377
766      0.13015243029945367
1549     0.12938941344572305
1166     0.11950594165584667
```

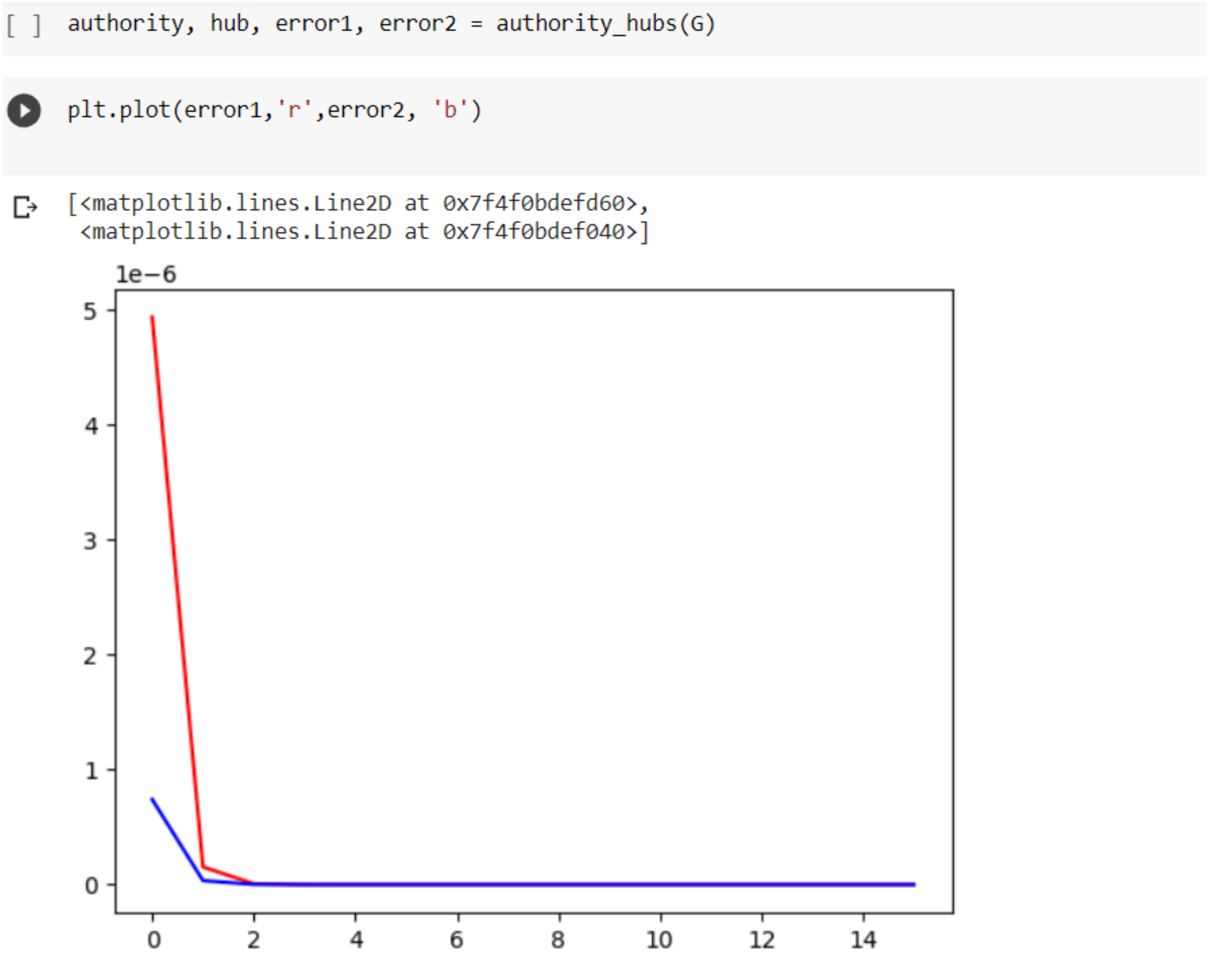
Error plots :

```
] error, rank = PageRank(G)
```

```
] plt.plot(error)
```

```
[<matplotlib.lines.Line2D at 0x7f4f0be26ca0>]
```





```
[ ]  authority, hub, error1, error2 = authority_hubs(G)
```

```
plt.plot(error1,'r',error2, 'b')
```

```
[<matplotlib.lines.Line2D at 0x7f4f0bdefd60>,
 <matplotlib.lines.Line2D at 0x7f4f0bdef040>]
```

Comparison :

By comparing the top nodes identified by each algorithm, we can see that they are not the same. The PageRank algorithm identifies nodes that have a high number of incoming links from other important nodes, while the Authority and Hub algorithm identifies nodes that have both high-quality incoming and outgoing links. However, we can see some overlap between the two algorithms in terms of the top nodes identified, which suggests that they are complementary measures of node importance in the network.