PSG COLLEGE OF TECHNOLOGY, COIMBATORE 641004

Department of Computer Science & Engineering

19ZO02 – SOCIAL AND ECONOMIC

NETWORK ANALYSIS

SOCIAL ECONOMIC NETWORK ANALYSIS OF US PRESIDENTS AND OFFICIALS

By

19Z331 – M.S.Ajith Narayana

19Z333 – Dhanush Reddy

19Z340 - Venkat Ram

19Z360 – Venkata Gowtham

20Z463 - Niranjan

BE CSE G2 (2019 – 2023)

PROBLEM STATEMENT

There are so many presidents for US among them who are most important influential people and what are the important connections between them and how we are going to build the subgroups for the most influential people and how we are going to analyse the network between them

DATASET DESCRIPTION

The dataset used for the application contains US_PRESIDENTS and OFFICIALS csv file that contains various required names. The important persons are the presidents and we connect the relationships and edges based on their friendships and the timeline they have worked in or the same party they have worked with. Visualization of the dataset will be as followed.
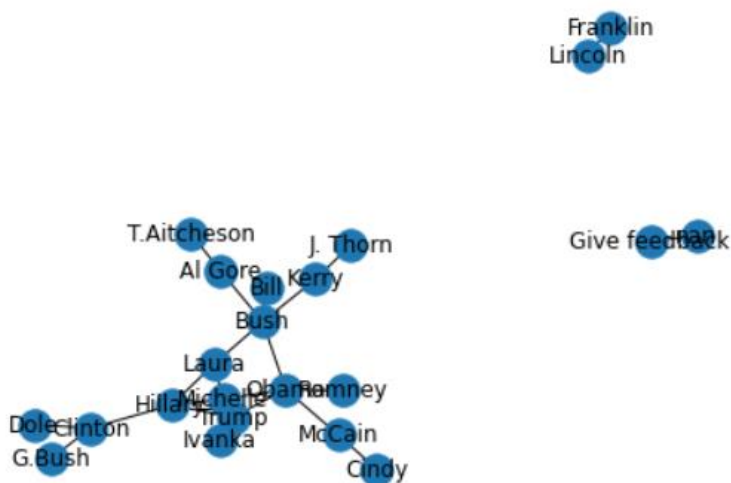
TOOLS USED

NetworkX

NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and function of complex networks. It is used to study large complex networks

represented in the form of graphs with nodes and edges. Using networkx we can load and store complex networks. We can generate many types of random and classic networks, analyse network structure, build network models, design new network algorithms and draw networks.

Matplotlib.pyplot:

Matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels and we can visualize the graph through this for our network and also it is useful in identifying the structures

Pandas:

Pandas is a Python library used for working with data sets.It has functions for analyzing, cleaning, exploring, and manipulating data.Pandas allows us to analyze data and make conclusions based on statistical theories.Pandas can clean messy data sets, and make them readable and relevant.

CHALLENGES FACED

- Visualization of the dataset remained a daunting process. Thus, evaluation of various metrics such as the betweenness centrality and eigen vector centrality remained computationally infeasible. Therefore, it required fetching a sub-portion of the dataset without compromising its real world characteristics such as the degree distribution as per the power law.
- Coding is the most challenging part we have faced in this project as we have to learn some of the concepts from scratch and execute it in the platforms we have never worked before
- Finding the Suitable Centrality Measures to help us in identifying popularity, most liked, and biggest influencers within the network.
- A picture speaks a thousand words but a graph speaks more than 10 thousand words as well as relationships between them analyzing the graphs with the concepts we have been taught ,like bridges and local bridges and also we have to see the centrality measures like betweenness which means who was acting like a bridge for many connections and closeness centrality to measure how close they are to each other

CONTRIBUTION OF TEAM MEMBERS

Roll No.	Name	Contribution
19Z331	Ajith Narayana	Report , Performance Analysis of Recommendation Model and Coding
19Z333	Dhanush Reddy	Analysis of Recommendation model and dataset creation and visualization
19Z340	Venkat Ram	Coding And Analysis of Recommendation Model
19Z360	Venkata Gowtham	Coding , Report & Performance Analysis of Recommendation Model
20Z463	Niranjan	Analysis of Recommendation model , dataset creation and Visualization

ANNEXURE I : CODE

Code link:

https://colab.research.google.com/drive/1K_YQO1AJzZl2j6xqClzz1TFY_CILTbtY?usp=sharing

ANNEXURE II : SNAPSHOTS OF OUTPUT

- **Most Influential People**:

```
[ ]  most_influential = nx.degree_centrality(us_graph)
     for w in sorted(most_influential, key=most_influential.get, reverse=True):
         print(w, most_influential[w])

     Bush 0.23809523809523808
     Obama 0.23809523809523808
     Hillary 0.19047619047619047
     Michelle 0.19047619047619047
     Clinton 0.14285714285714285
     Laura 0.14285714285714285
     Trump 0.14285714285714285
     Ivanka 0.09523809523809523
     Al Gore 0.09523809523809523
     Kerry 0.09523809523809523
     McCain 0.09523809523809523
     G.Bush 0.047619047619047616
     Dole 0.047619047619047616
     Romney 0.047619047619047616
     Bill 0.047619047619047616
     J. Thorn 0.047619047619047616
     T.Aitcheson 0.047619047619047616
     Cindy 0.047619047619047616
     Give feedback 0.047619047619047616
     nan 0.047619047619047616
     Franklin 0.047619047619047616
     Lincoln 0.047619047619047616
```
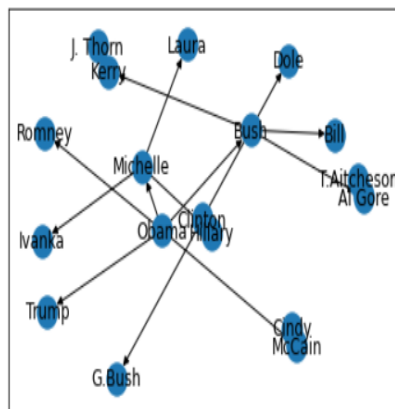
- **Most Important Connections:**

```python
most_important_link = nx.eigenvector_centrality(us_graph)
for w in sorted(most_important_link, key=most_important_link.get, reverse=True):
    print(w, most_important_link[w])
```
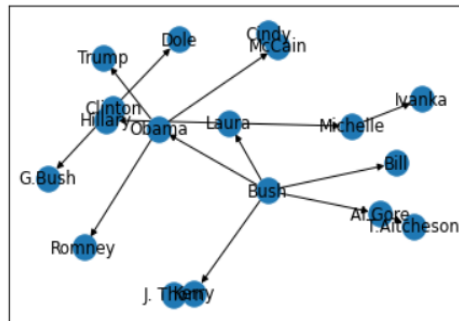
```
Michelle 0.42564995945869316
Obama 0.41701786958849474
Hillary 0.3824034554635519
Laura 0.3540202525602793
Bush 0.34141644782776975
Trump 0.31661678963342654
Ivanka 0.22860648801381267
Clinton 0.1453463495334645
McCain 0.14189557882103113
Romney 0.12843588459321245
Al Gore 0.1161731381116166
Kerry 0.1161731381116166
Bill 0.10515278420191813
G.Bush 0.04476379762447264
Dole 0.04476379762447264
Cindy 0.043702017939385294
J. Thorn 0.03578057533800623
T.Aitcheson 0.03578057533800623
Give feedback 2.0338388966403895e-10
nan 2.0338388966403895e-10
Franklin 2.0338388966403895e-10
Lincoln 2.0338388966403895e-10
```

- **Building subgroups for most influential people using breath first search (BFS):**
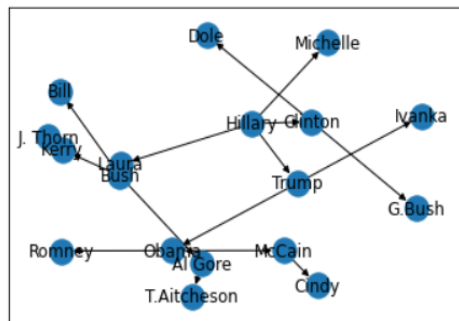
```python
group1 = nx.bfs_tree(us_graph,"Obama")
group2 = nx.bfs_tree(us_graph,"Bush")
group3 = nx.bfs_tree(us_graph,"Hillary")
nx.draw_networkx(group1)
```

```
[ ] nx.draw_networkx(group2)
```



```
[ ] nx.draw_networkx(group3)
```


REFERENCES

- https://www.w3schools.com/python/pandas/default.asp
- https://gephi.org/users/
- https://networkx.org/
- http://snap.stanford.edu/data/
- https://www.youtube.com/watch?v=CPQeSmDGiOQ
- https://matplotlib.org/stable/tutorials/introductory/pyplot.html
- https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/#:~:text=Centrality%20measures%20are%20a%20vital,but%20they%20all%20work%20differently.
- https://www.nvidia.com/en-us/glossary/data-science/graph-analytics/#:~:text=Graph%20analytics%20is%20an%20emerging,%2C%20social%2C%20and%20information%20systems.
- https://medium.com/swlh/a-tutorial-on-networkx-network-analysis-in-python-part-i-43c1d35830b6
- https://www.loc.gov/rr/print/list/057_chron.html

Plagiarism Report:

VeriGuide - Originality Report
Individual Report

<u>Background Information</u>

File Name:	Team-9_Report.pdf
Report Generated On:	11/11/2022, 8:06:41 PM

<u>Similarity Statistics Overview</u>

Similar Sentence(s) Found By VeriGuide:	1 out of 29 sentences = 3.45%
Similar Sentence(s) Filtered by User:	1 out of 29 sentences = 3.45%
Sentence(s) Selected By User To Export:	0

<u>Similarity Statistics for Each Source</u>

Entry	Source	From	Similarity
1	https://gephi.org/	Internet	1 / 29 = 3.45%