Ashpak Sheikh
NETWORK ANALYSIS
BATCH : DSWDMOD 020421

Problem Statement: -

There are two datasets consisting of information for the connecting routes and flight halt. Create network analytics models on both the datasets separately and measure degree centrality, degree of closeness centrality, and degree of in-between centrality.

- Create a network using edge list matrix (directed only**).**

About Data: -

We have been given data about connecting routes of flight, start point and stop point

Analysis with Python: -

1) Connecting Routs: -

#importing libraries to read and manpulate detac

import pandas as pd

import numpy as np

import networkx as nx

import matplotlib.pyplot as plt

#loading deta set into python in form of pandas dataframe

routs=pd.read_csv("D:/DataScience/Class/assignment working/Network Analysis/connecting_routes.csv")

routs=routs.sample(frac=0.1)

#initializing empty graph to plot edges and nodes

g=nx.DiGraph()

#filling data to form edges and nodes

g= nx.from_pandas_edgelist(routs, source="AER", target="KZN", create_using=nx.DiGraph())

#printing info

print(nx.info(g))

```
In [352]: print(nx.info(g))
Name:
Type: DiGraph
Number of nodes: 1981
Number of edges: 6276
Average in degree:   3.1681
Average out degree:   3.1681
```

Ashpak Sheikh
NETWORK ANALYSIS
BATCH : DSWDMOD 020421

#calculating degree of centrality

centrality = nx.degree_centrality(g)

centrality(g)

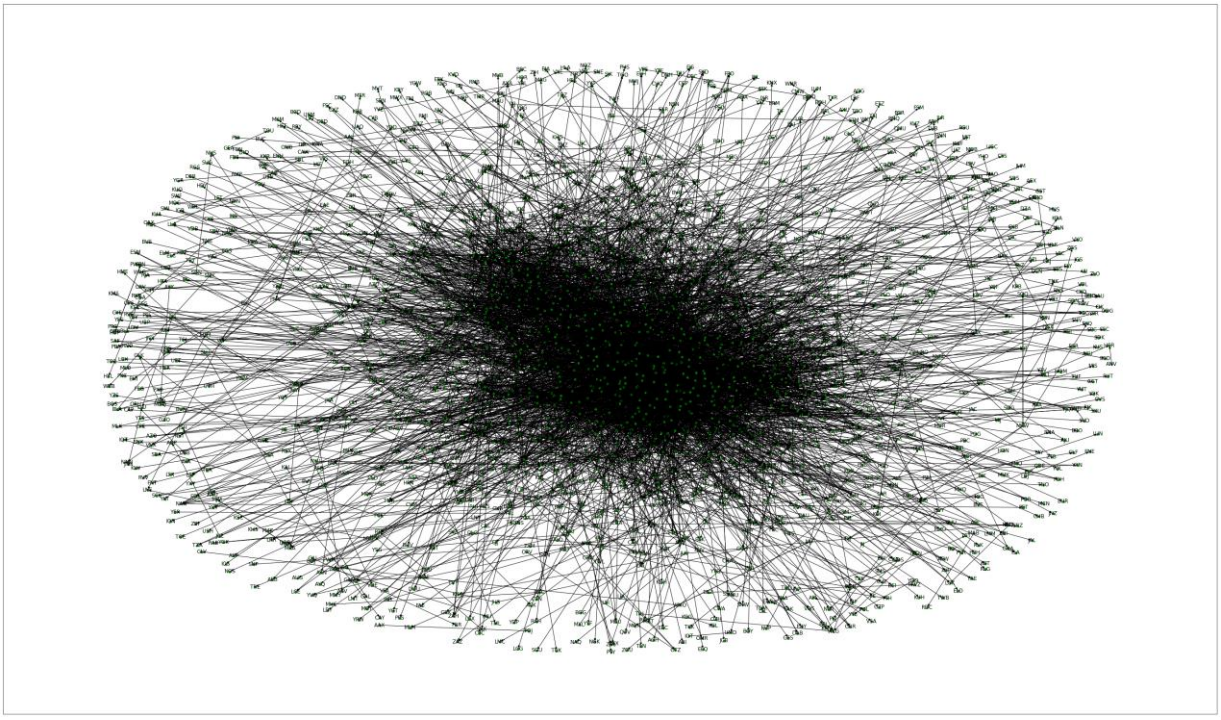
```
In [356]: nx.info(g)
Out[356]: 'Name: \nType: DiGraph\nNumber of nodes: 1981\nNumber of edges:
6276\nAverage in degree:   3.1681\nAverage out degree:   3.1681'
```

#calculating positions

pos = nx.spring_layout(g, k = 0.15)

plt.figure(figsize=(50,30))

nx.draw_networkx(g, pos, node_size = 25, node_color = 'green')



#calculating closeness centrality

closeness = nx.closeness_centrality(g)

print(closeness)

```
## Betweeness Centrality
b = nx.betweenness_centrality(g)
print(b)

## Eigen-Vector Centrality
evg = nx.eigenvector_centrality(g)
print(evg)

# cluster coefficient
cluster_coeff = nx.clustering(g)
print(cluster_coeff)

# Average clustering
cc = nx.average_clustering(g)
print(cc)
```

2) Flight Haults: -

```
#importing required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx

#loading deta set in pandas
haults=pd.read_csv("D:/DataScience/Class/assignment working/flight_hault.csv")
```

```
In [372]: haults.head()
Out[372]:
   1                      Goroka      Goroka.1  ...    10  U  Pacific/Port_Moresby
0  2                      Madang        Madang  ...  10.0  U  Pacific/Port_Moresby
1  3                 Mount Hagen   Mount Hagen  ...  10.0  U  Pacific/Port_Moresby
2  4                      Nadzab        Nadzab  ...  10.0  U  Pacific/Port_Moresby
3  5  Port Moresby Jacksons Intl  Port Moresby  ...  10.0  U  Pacific/Port_Moresby
4  6                   Wewak Intl         Wewak  ...  10.0  U  Pacific/Port_Moresby

[5 rows x 12 columns]
```

#creating empty graph

g=nx.from_pandas_edgelist(haults,source="GKA", target="AYGA", create_using=nx.DiGraph())

#printing info about number rof nodes and edges

print(nx.info(g))

```
In [377]: print(nx.info(g))
Name:
Type: DiGraph
Number of nodes: 12650
Number of edges: 7382
Average in degree:   0.5836
Average out degree:   0.5836
```

#calculating degree of centrality from graph info

centrality=nx.degree_centrality(g)

print(centrality)

#calculating positions for graphs

pos=nx.spring_layout(g,k= 0.15)

#plotting network graph

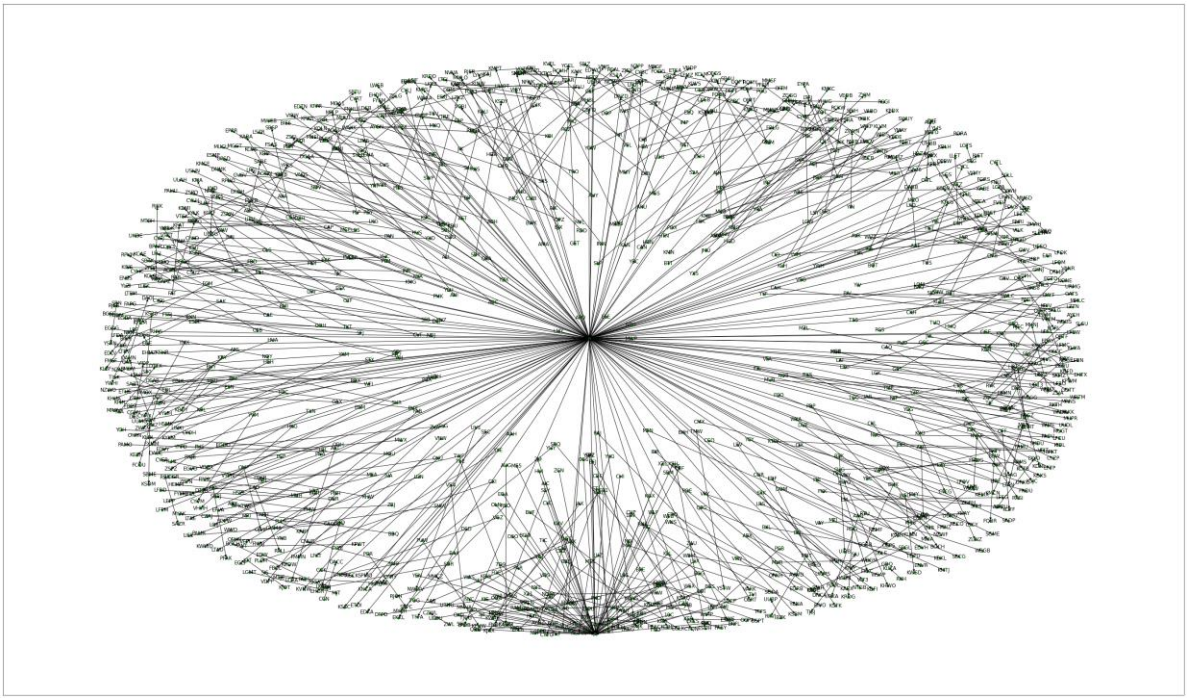
plt.figure(figsize=(50,30))

nx.draw_networkx(g, pos , node_size=25, node_color="green")

Ashpak Sheikh

NETWORK ANALYSIS

BATCH : DSWDMOD 020421



```
#calculating closeness centrality
closeness = nx.closeness_centrality(g)
print(closeness)

## Betweeness Centrality
b = nx.betweenness_centrality(g)
print(b)

## Eigen-Vector Centrality
evg = nx.eigenvector_centrality(g)
print(evg)

# cluster coefficient
cluster_coeff = nx.clustering(g)
print(cluster_coeff)
```

```
# Average clustering

cc = nx.average_clustering(g)

print(cc)
```