SOCIAL AND ECONOMIC NETWORK ANALYSIS

Report on Analysis of "Airline Travel Reachability Network" – Project submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

SUBMITTED BY

ADWIN SANJO J (18Z205)
ANANTHAROOBAN B R (18Z205)
ARUN PRASHATH S (18Z208)
JAYABHARTHI A (18Z221)
RIDHI K C (18Z241)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MARCH 2021

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

"AIRLINE TRAVEL REACHABILITY NETWORK" – MINI PROJECT

PROBLEM STATEMENT

The Airline network is one of the largest networks in the world spanning the entire globe. According to ICAO's annual global statistics, 4.5 billion people traveled by air in the year 2019. Hence it is vital to manage and analyze the world airline networks to enhance the travel experience and efficiency of the world air transport. Real-world flight networks can be modeled as a graph where each node represents an airport and a directed edge between two nodes A to B represent the existence of a direct flight from airport A to B. Here, we analyze a sample data set of Airline networks spanning Canada and the USA.

DATASET DESCRIPTION

We have considered a dataset that is an Airline reachability network for cities in Canada and the USA. The edges in the dataset are weighted in a way that there is an edge from city i to city j if airline travel time is less than a threshold value. The travel time includes the stopover delays also. The network is asymmetric because of headwinds. The dataset includes the city metropolitan populations, latitude, and longitude.

Attributes of reachability dataset

- from node: source city id
- to node: destination city id
- weight: Estimated airline travel time based on a threshold

Attributes of reachability-meta dataset

- node id: city id or airport id
- name: city name
- metro pop: Population of a city
- latitude: geographical latitude of the city
- longitude: geographical longitude of city

TOOLS USED

LANGUAGE USED: Python

LIBRARIES USED:

- **networkx** - a software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks in python
- **sklearn** - Contains various classification, regression, and clustering algorithms (k-means)
- **matplotlib** - Matplotlib is a plotting library for python. It provides an object-oriented API for embedding plots into applications
- **pandas** - Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning

CHALLENGES FACED

1.The graph obtained from the dataset is very dense, because of which it is time-consuming

2.The graph obtained is an almost complete graph, so the betweenness measures obtained are more or less the same values

3. Visualization of the dense network is very difficult, so it is time as well as memory consuming

4.Data Preprocessing was not done on the edges because of the large number of redundant edges present in the dataset

CONTRIBUTION OF TEAM MEMBERS

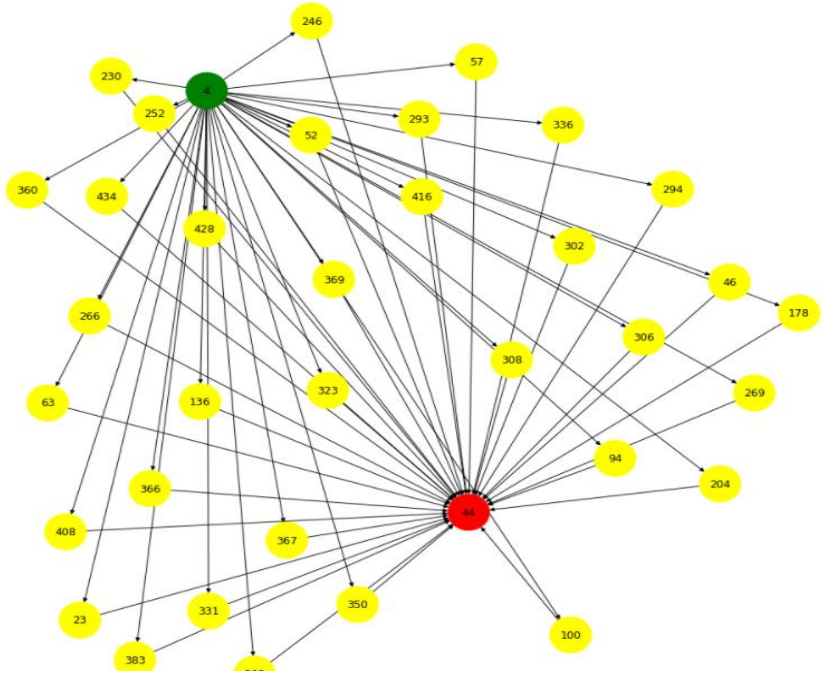
NAME	ROLL NO.	CONTRIBUTION
ANANTHAROOBAN B R JAYABHARATHI A **SET 1**	18Z205 18Z221	✓ Visualization and analysis of our network by 5 centrality measures ✓ Visualizing the path from Source to Destination with given no. of hops ✓ Visualizing the set of nodes that could be reached from a particularly given node with the given hop distance ✓ Visualization of the Degree distribution ✓ Analyzing, if there exists an edge from node A to node B, then whether there exists another edge from node B to node A also
ADWIN SANJO J ARUN PRASHATH S **SET 2**	18Z202 18Z208	✓ Shortest Path from a source to destination ✓ Comparison and Visualization of 4 models, • Erdos Renyi • Watts Strogatz • Barabasi Albert • Power law cluster with the original network
RIDHI K C **SET 3**	18Z241	Clustering between, ✓ Total number of Carrier delay and the total number of delays due to weather ✓ Total number of flights canceled and the total number of flights on time ✓ Total number of flights delayed and the total number of flights on time ✓ Total number flights on time and total number flights

ANNEXURE I: CODE

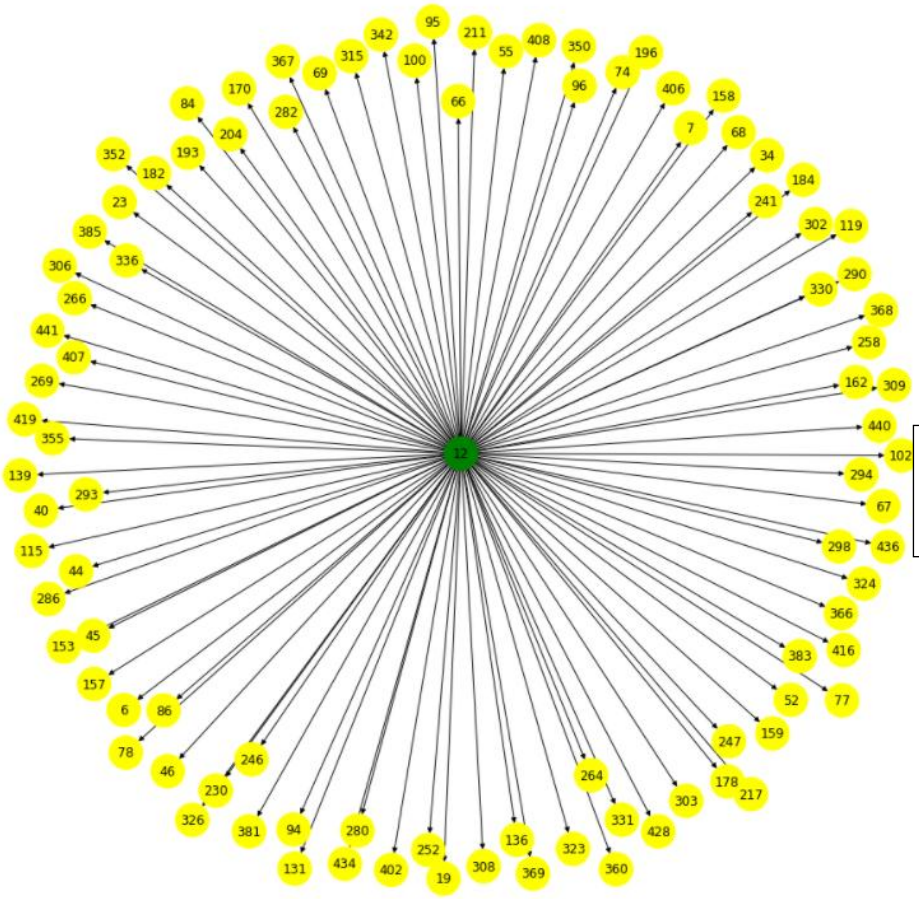
https://github.com/AnantharoobanBR/SENA_Mini_Project

ANNEXURE II: SNAPSHOTS OF THE OUTPUT SET 1 START

CENTRALITY MEASURE	TOP 5 AIRPORTS (Based on Centrality Value)
Betweenness Centrality	Airport Cities which connect Remote Airports ❖ los angeles ca ❖ denver co ❖ new york ny ❖ toronto on ❖ san francisco ca
In-Degree Centrality	Airport Cities with High Arrivals ❖ los angeles ca ❖ san francisco ca ❖ dallas/fort worth tx ❖ chicago il ❖ las vegas nv
Out-Degree Centrality	Airport Cities with High Departures ❖ los angeles ca ❖ las vegas nv ❖ chicago il ❖ san francisco ca ❖ denver co
Closeness Centrality	Airport Cities with High Accessibility ❖ los angeles ca ❖ san francisco ca ❖ dallas/fort worth tx ❖ chicago il ❖ las vegas nv
Eigen Vector Centrality	Airport Cities which connect Important Airports ❖ los angeles ca ❖ san francisco ca ❖ dallas/fort worth tx ❖ las vegas nv ❖ chicago il

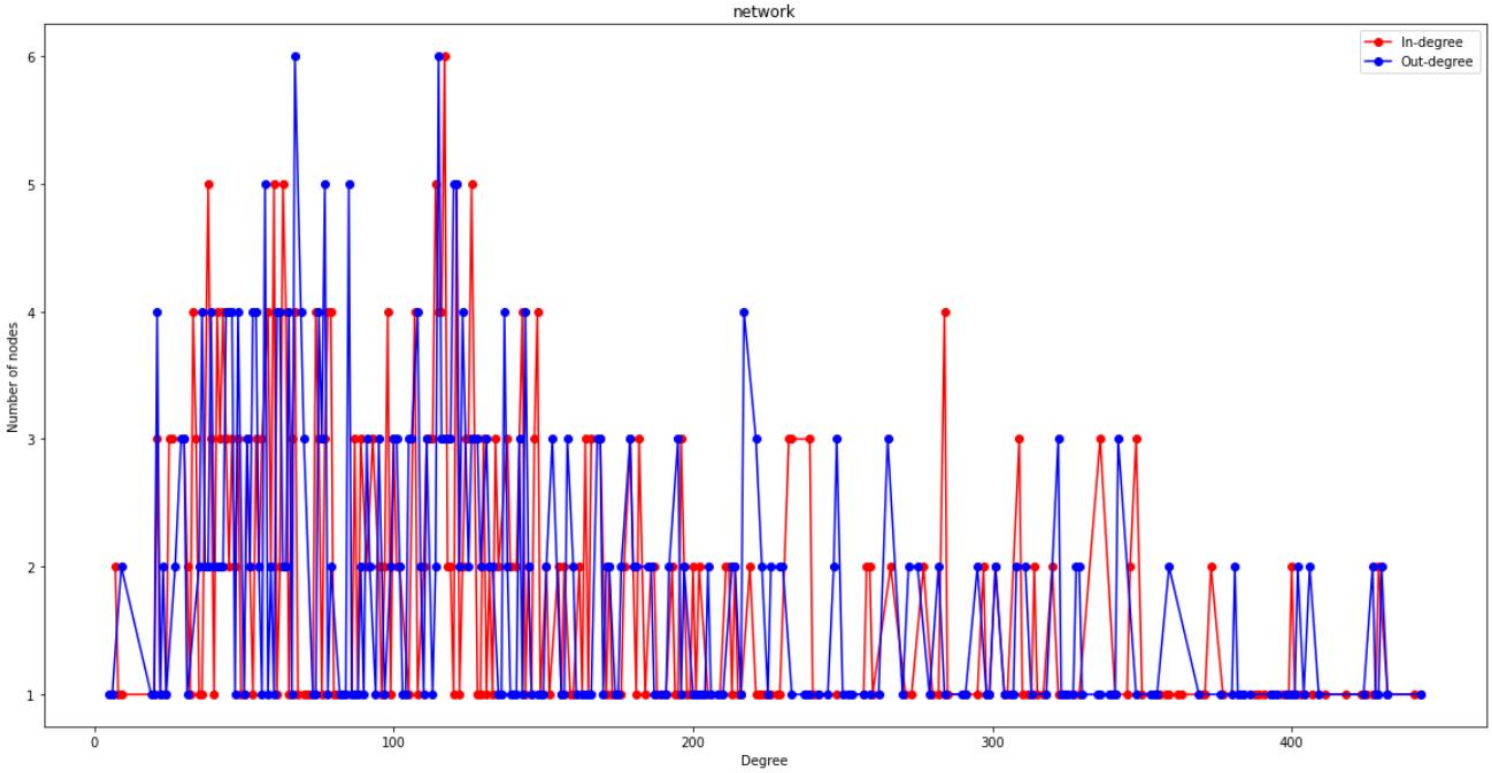


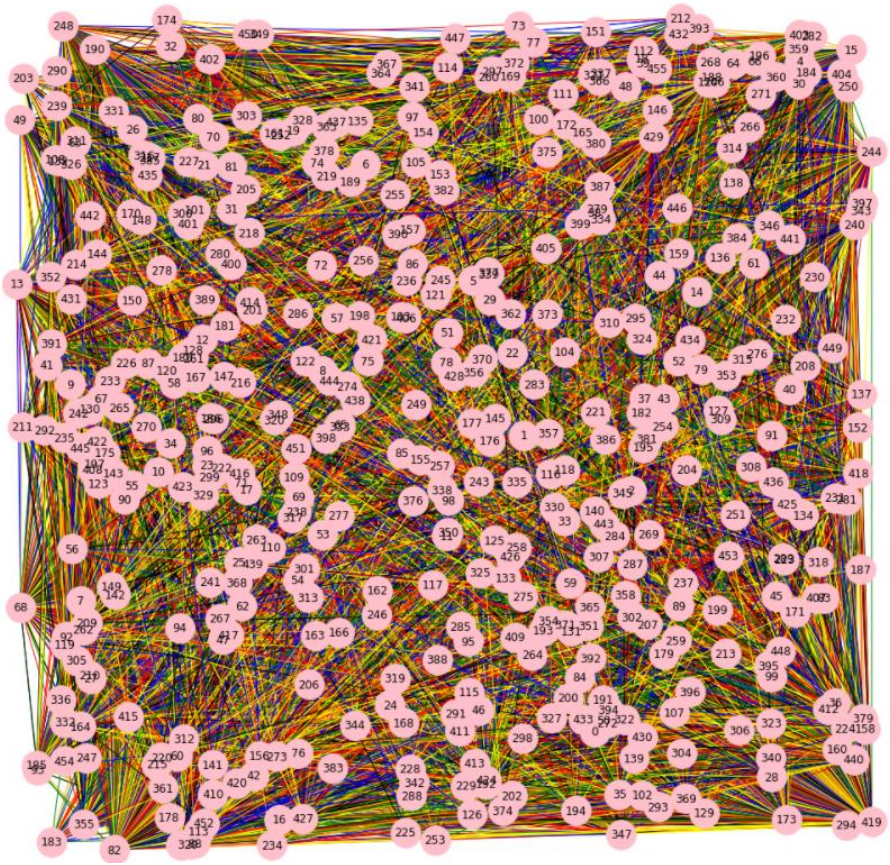
Visualizing the path from Source Airport(4) to Destination Airport(44) with given no. of hops(2)



Visualizing the set of nodes that could be reached from a particular given node(12) with the given hop distance(1)

DEGREE DISTRIBUTION CURVE





Analyzing, if there exists an edge from node A to node B, then whether there exists another edge from node B to node A also.

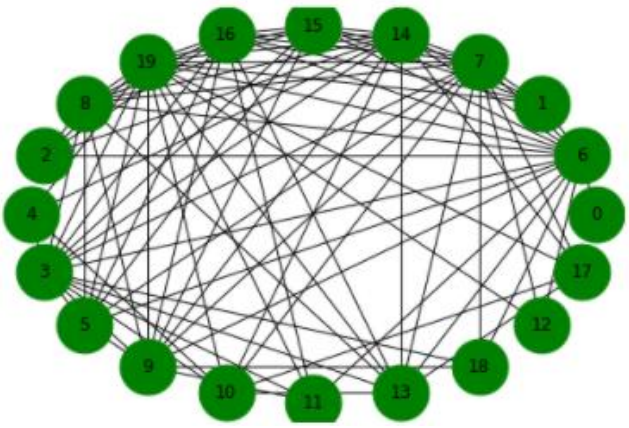
Here, the graph looks a bit clumsy to properly visualize the existence of edges

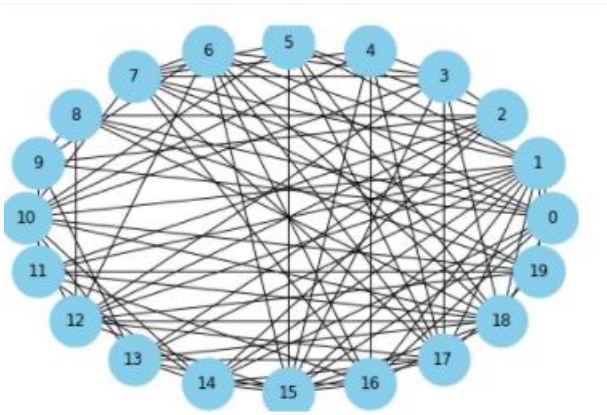
Since, we already explained in one of our challenges – that our graph is almost complete graph; if only one color is used, then the edges will be merged & can't be distinguished

SET 1 END ; SET 2 START

SHORTEST PATH from a source to destination

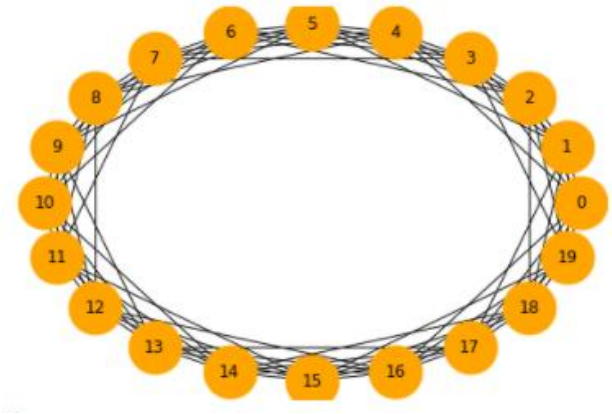
Enter the Source City : abbotsford BC
Enter the destination City :anchorage AK
abbotsford bc -> albany ny -> anchorage ak ->
Enter the city to remove : albany ny
abbotsford bc -> calgary ab -> anchorage ak ->



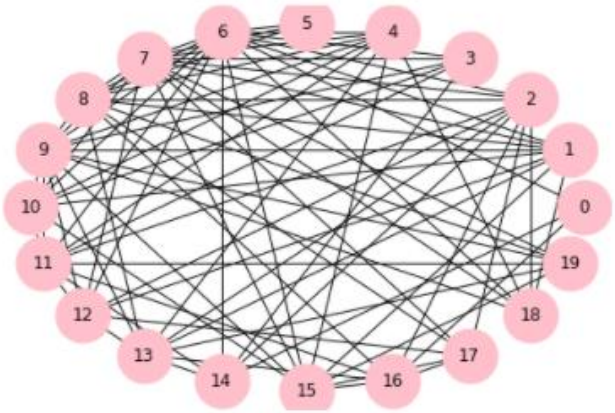


ORIGINAL GRAPH
Number of nodes: 20
Number of edges: 89
Average degree: 8.9000
Average Clustering Coefficient: 0.80522
Transitivity of the Graph: 0.6029
Average Shortest Path Length: 1.635704

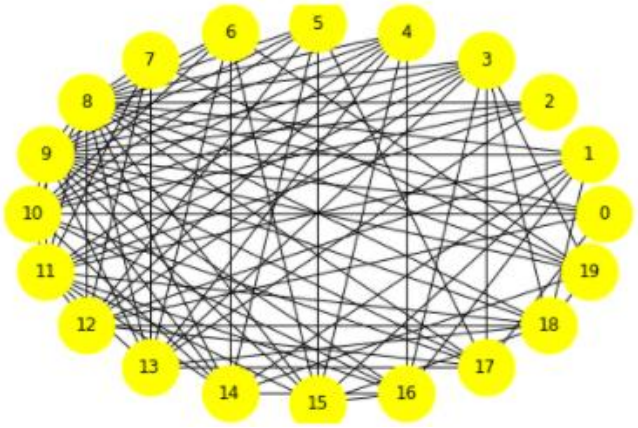
ERDOS RENYI MODEL
Number of nodes: 20
Number of edges: 90
Average degree: 9.0000
Average Clustering Coefficient: 0.47124
Transitivity of the Graph: 0.481723
Average Shortest Path Length: 1.536842



WATTS STROGATZ MODEL
Number of nodes: 20
Number of edges: 80
Average degree: 8.0000
Average Clustering Coefficient: 0.63353
Transitivity of the Graph: 0.631016
Average Shortest Path Length: 1.710526



BARABASI ALBERT MODEL
Number of nodes: 20
Number of edges: 84
Average degree: 8.4000
Average Clustering Coefficient: 0.543368
Transitivity of the Graph: 0.49930
Average Shortest Path Length: 1.56315



POWER LAW CLUSTER MODEL -- REPRESENTS AIR TRANSPORTATION NETWORK
Number of nodes: 20
Number of edges: 93
Average degree: 9.3000
Average Clustering Coefficient: 0.6069758
Transitivity of the Graph: 0.5402684
Average Shortest Path Length: 1.5105263

Comparing Real world flight network with the existing network models
The three most important characteristics of a flight network are,

- ✓ Clustering - HIGH
- ✓ Transitivity - HIGH
- ✓ Average Path Length - LOW

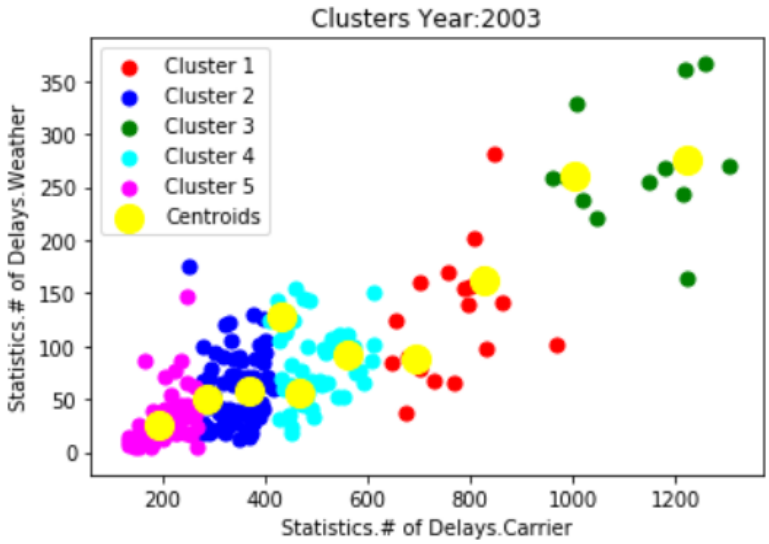
20 nodes are chosen from the dataset and its properties are compared with known models like,

1. Erdos Renyi
    a. Clustering is low
    b. Transitivity is low
    c. Average Path length is similar to the original network
2. Watts-Strogatz
    a. Clustering is similar to the original network
    b. Transitivity is similar to the original network
    c. Average path is length is larger

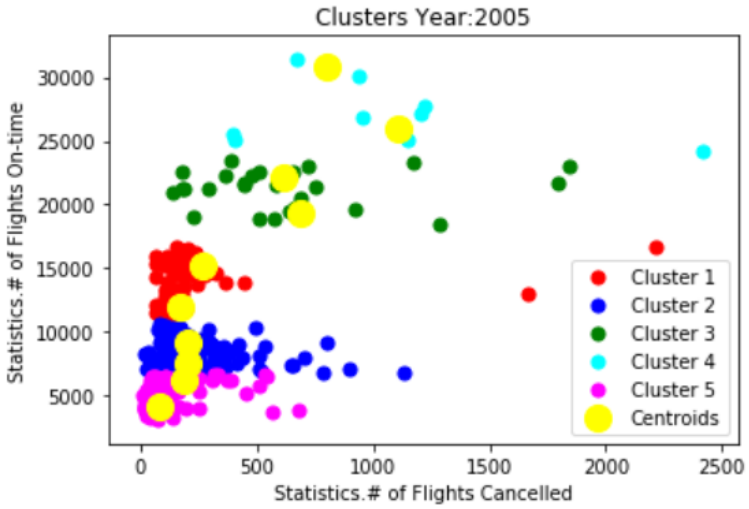
3. Barabasi Albert Model
    a. Average clustering is less
    b. Transitivity is less
    c. Average Shortest Path is similar to the original graph.

Real-life flight network exhibits scale-free property with high clustering. So, a real-life flight network can be represented by the power-law cluster model, which exhibits power-law distribution with high clustering.

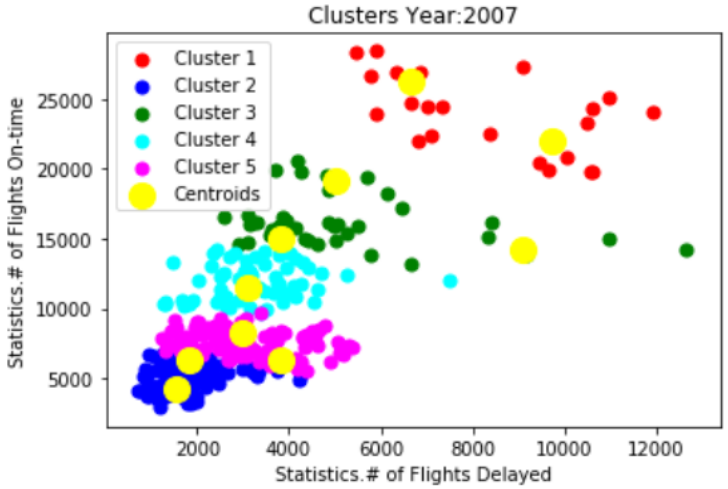
SET 2 END; SET 3 START



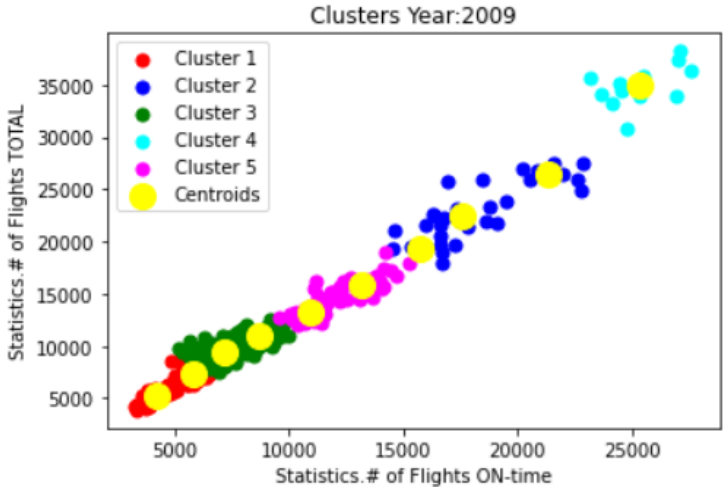
RED → Above medium number of airplane delays and weather delays
BLUE → Intermediate number of airplane delays and weather delays
GREEN → High number of airplane delays and weather delays
CYAN → Medium of airplane delays and weather delays
MAGENTA→ Low number of airplane delays and weather delays



RED → Medium number of flights on time and low flights cancelled
BLUE → Intermediate number of flights on time and low/few medium flights cancelled
GREEN → Above medium number of flights on time and low & medium flights cancelled
CYAN → High number of flights on time and somewhat average flights cancelled
MAGENTA→ Low number of flights on time and low flights cancelled



RED → High number of flights on time and high flights delayed
BLUE → Low number of flights on time and low flights delayed
GREEN → Above medium number of flights on time and medium flights delayed
CYAN → Medium number of flights on time and low flights delayed
MAGENTA→ Intermediate number of flights on time and low/few medium flights delayed



RED → Medium number of flights on time
BLUE → High number of flights on time
GREEN → Low number of flights on time
CYAN → Above medium number of flights on time
MAGENTA→ Intermediate number of flights on time

From the dataset, we take two columns and make them as x and y-axis. After which we invoke the k-means function from the sk_learn library where we use the formula of inertia which will split the resultant output into 5 different clusters. The clusters vary based on the attributes chosen.
The outputs are as follows:
1) Clustering between the total number of Carrier delay and the total number of delays due to weather
2) Clustering between the total number of flights canceled and the total number of flights on time
3) Clustering between the total number of flights delayed and the total number of flights on time
4) Clustering between the total number of flights on time and the total number of flights
SET 3 END

REFERENCES

DATASET LINK: https://snap.stanford.edu/data/reachability.html
https://towardsdatascience.com/learn-python-data-analytics-by-example-airline-arrival-delays-e26356e8ae6b
https://www.geeksforgeeks.org/python-visualize-graphs-generated-in-networkx-using-matplotlib/
https://networkx.org/documentation/stable/reference/algorithms/
https://www.cl.cam.ac.uk/~cm542//teaching/2010/stna-pdfs/stna-lecture8.pdf
https://blog.dominodatalab.com/social-network-analysis-with-networkx/
https://www.cl.cam.ac.uk/teaching/1415/L109/l109-tutorial_2015.pdf
https://corgis-edu.github.io/corgis/csv/airlines/
https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/
https://www.kaggle.com/shrikantuppin/kmeans-hierarchical-clustering-eastwestairlines
https://scikit-learn.org/stable/modules/clustering.html
https://www.tesisenred.net/bitstream/handle/10803/144526/TOLG1de1.pdf?sequence=1&isAllowed=y