Topic: Network Analytics

Please ensure you update all the details:
Name: ULLI VENKATA SAI KUMAR Batch ID: 04072024HYD10AM

Topic: Network Analytics

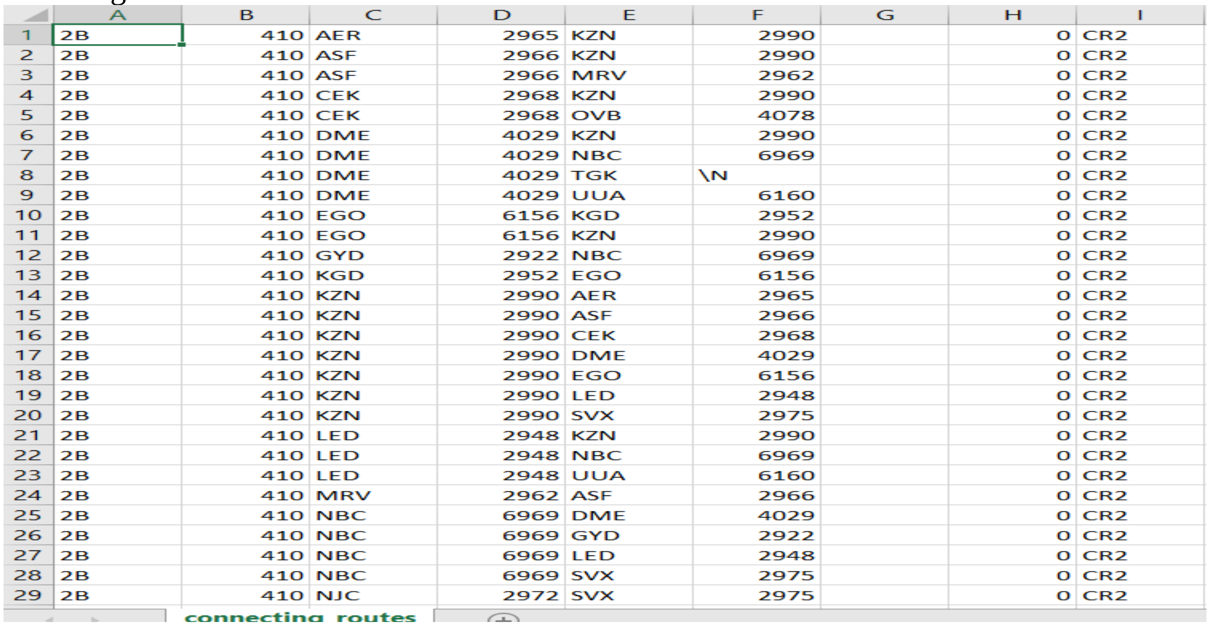
Problem Statement: -
Two datasets consisting of information for the connecting routes and flight halt are provided. Create network analytics models on both datasets separately. Using various network analytics-based measures derive insights for the business to benefit from the data available.

- Create a network using an edge list matrix (directed only).
- Column to be used for respective datasets

Flight_halt = ID", "Name", "City", "Country", "IATA_FAA", "ICAO", "Latitude", "Longitude", "Altitude", "Time", "DST", "Tz database time"

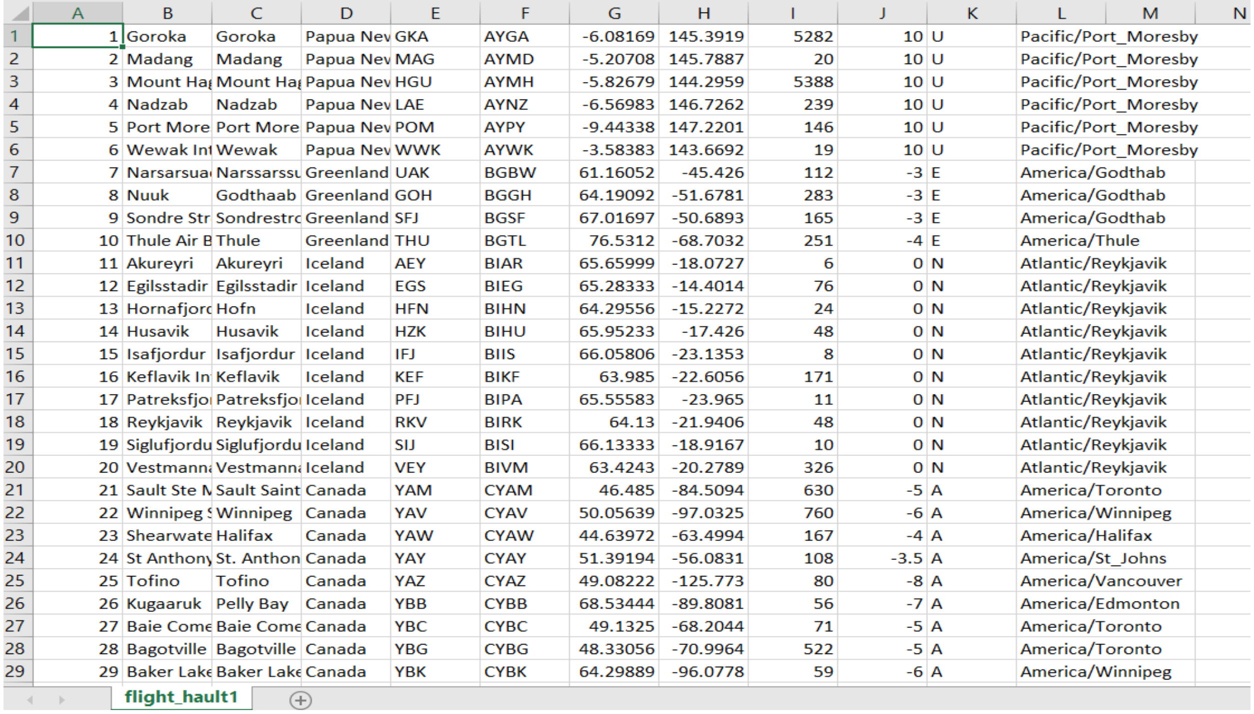
connecting routes = "flights", " ID", "main Airport", "main Airport ID", "Destination ", "DestinationID", "haults", "machinery"

cohaltsng routes



	A	B	C	D	E	F	G	H	I
1	2B	410	AER	2965	KZN	2990		0	CR2
2	2B	410	ASF	2966	KZN	2990		0	CR2
3	2B	410	ASF	2966	MRV	2962		0	CR2
4	2B	410	CEK	2968	KZN	2990		0	CR2
5	2B	410	CEK	2968	OVB	4078		0	CR2
6	2B	410	DME	4029	KZN	2990		0	CR2
7	2B	410	DME	4029	NBC	6969		0	CR2
8	2B	410	DME	4029	TGK	\N		0	CR2
9	2B	410	DME	4029	UUA	6160		0	CR2
10	2B	410	EGO	6156	KGD	2952		0	CR2
11	2B	410	EGO	6156	KZN	2990		0	CR2
12	2B	410	GYD	2922	NBC	6969		0	CR2
13	2B	410	KGD	2952	EGO	6156		0	CR2
14	2B	410	KZN	2990	AER	2965		0	CR2
15	2B	410	KZN	2990	ASF	2966		0	CR2
16	2B	410	KZN	2990	CEK	2968		0	CR2
17	2B	410	KZN	2990	DME	4029		0	CR2
18	2B	410	KZN	2990	EGO	6156		0	CR2
19	2B	410	KZN	2990	LED	2948		0	CR2
20	2B	410	KZN	2990	SVX	2975		0	CR2
21	2B	410	LED	2948	KZN	2990		0	CR2
22	2B	410	LED	2948	NBC	6969		0	CR2
23	2B	410	LED	2948	UUA	6160		0	CR2
24	2B	410	MRV	2962	ASF	2966		0	CR2
25	2B	410	NBC	6969	DME	4029		0	CR2
26	2B	410	NBC	6969	GYD	2922		0	CR2
27	2B	410	NBC	6969	LED	2948		0	CR2
28	2B	410	NBC	6969	SVX	2975		0	CR2
29	2B	410	NJC	2972	SVX	2975		0	CR2

Flight_hault1



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	Goroka	Goroka	Papua Nev	GKA	AYGA	-6.08169	145.3919	5282	10	U	Pacific/Port_Moresby		
2	2	Madang	Madang	Papua Nev	MAG	AYMD	-5.20708	145.7887	20	10	U	Pacific/Port_Moresby		
3	3	Mount Ha	Mount Ha	Papua Nev	HGU	AYMH	-5.82679	144.2959	5388	10	U	Pacific/Port_Moresby		
4	4	Nadzab	Nadzab	Papua Nev	LAE	AYNZ	-6.56983	146.7262	239	10	U	Pacific/Port_Moresby		
5	5	Port More	Port More	Papua Nev	POM	AYPY	-9.44338	147.2201	146	10	U	Pacific/Port_Moresby		
6	6	Wewak Int	Wewak	Papua Nev	WWK	AYWK	-3.58383	143.6692	19	10	U	Pacific/Port_Moresby		
7	7	Narsarsua	Narssarssu	Greenland	UAK	BGBW	61.16052	-45.426	112	-3	E	America/Godthab		
8	8	Nuuk	Godthaab	Greenland	GOH	BGGH	64.19092	-51.6781	283	-3	E	America/Godthab		
9	9	Sondre Str	Sondrestrc	Greenland	SFJ	BGSF	67.01697	-50.6893	165	-3	E	America/Godthab		
10	10	Thule Air B	Thule	Greenland	THU	BGTL	76.5312	-68.7032	251	-4	E	America/Thule		
11	11	Akureyri	Akureyri	Iceland	AEY	BIAR	65.65999	-18.0727	6	0	N	Atlantic/Reykjavik		
12	12	Egilsstadir	Egilsstadir	Iceland	EGS	BIEG	65.28333	-14.4014	76	0	N	Atlantic/Reykjavik		
13	13	Hornafjorc	Hofn	Iceland	HFN	BIHN	64.29556	-15.2272	24	0	N	Atlantic/Reykjavik		
14	14	Husavik	Husavik	Iceland	HZK	BIHU	65.95233	-17.426	48	0	N	Atlantic/Reykjavik		
15	15	Isafjordur	Isafjordur	Iceland	IFJ	BIIS	66.05806	-23.1353	8	0	N	Atlantic/Reykjavik		
16	16	Keflavik In	Keflavik	Iceland	KEF	BIKF	63.985	-22.6056	171	0	N	Atlantic/Reykjavik		
17	17	Patreksfjo	Patreksfjo	Iceland	PFJ	BIPA	65.55583	-23.965	11	0	N	Atlantic/Reykjavik		
18	18	Reykjavik	Reykjavik	Iceland	RKV	BIRK	64.13	-21.9406	48	0	N	Atlantic/Reykjavik		
19	19	Siglufjordu	Siglufjordu	Iceland	SIJ	BISI	66.13333	-18.9167	10	0	N	Atlantic/Reykjavik		
20	20	Vestmanna	Vestmanna	Iceland	VEY	BIVM	63.4243	-20.2789	326	0	N	Atlantic/Reykjavik		
21	21	Sault Ste M	Sault Saint	Canada	YAM	CYAM	46.485	-84.5094	630	-5	A	America/Toronto		
22	22	Winnipeg S	Winnipeg	Canada	YAV	CYAV	50.05639	-97.0325	760	-6	A	America/Winnipeg		
23	23	Shearwate	Halifax	Canada	YAW	CYAW	44.63972	-63.4994	167	-4	A	America/Halifax		
24	24	St Anthony	St. Anthon	Canada	YAY	CYAY	51.39194	-56.0831	108	-3.5	A	America/St_Johns		
25	25	Tofino	Tofino	Canada	YAZ	CYAZ	49.08222	-125.773	80	-8	A	America/Vancouver		
26	26	Kugaaruk	Pelly Bay	Canada	YBB	CYBB	68.53444	-89.8081	56	-7	A	America/Edmonton		
27	27	Baie Come	Baie Come	Canada	YBC	CYBC	49.1325	-68.2044	71	-5	A	America/Toronto		
28	28	Bagotville	Bagotville	Canada	YBG	CYBG	48.33056	-70.9964	522	-5	A	America/Toronto		
29	29	Baker Lake	Baker Lake	Canada	YBK	CYBK	64.29889	-96.0778	59	-6	A	America/Winnipeg		

Connecting routes:-

```
# Importing required libraries
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt


# Load the CSV files into DataFrames
connecting_routes = pd.read_csv('connecting_routes.csv')
# Display the first few rows of each dataset to understand the structure
connecting_routes_head = connecting_routes.head()
# Display the first few rows of the DataFrame
connecting_routes_head
# Renaming relevant columns for clarity
connecting_routes.columns = ['airline', 'flight_number', 'source_airport', 'source_airport_id',
'destination_airport', 'destination_airport_id', 'codeshare', 'stops', 'equipment']
# Display the cleaned DataFrame's first few rows
connecting_routes_cleaned = connecting_routes[['source_airport', 'destination_airport']]
connecting_routes_cleaned.head()
# Inspecting the actual column names to identify the mismatch
connecting_routes.columns
```

```
for_g = nx.from_pandas_edgelist(connecting_routes_cleaned,
                              source='source_airport',
                              target='destination_airport',
                              create_using=nx.DiGraph())
degree_centrality = nx.degree_centrality(for_g)
closeness_centrality = nx.closeness_centrality(for_g)
eigenvector_centrality = nx.eigenvector_centrality(for_g)
betweenness_centrality = nx.betweenness_centrality(for_g)
pos = nx.spring_layout(for_g)
nx.draw(for_g, pos, node_size=20, node_color='blue')
plt.show()
```

flight_hault:-

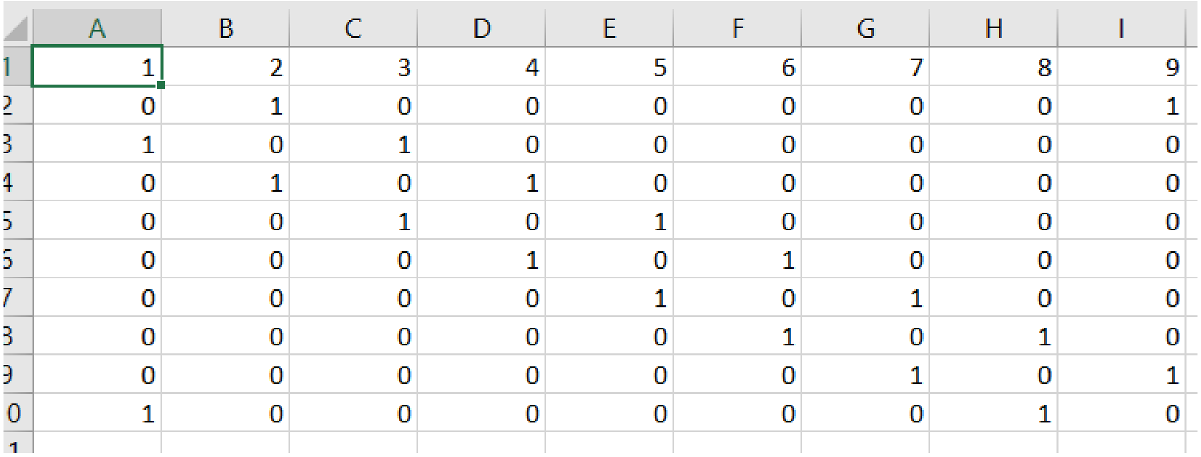
```
# Importing required libraries
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
# Load the CSV files into DataFrames
flight_hault = pd.read_csv('flight_hault.csv')
# Display the first few rows of each dataset to understand the structure
flight_hault_head = flight_hault.head()
# Display the first few rows of the DataFrame
flight_hault_head
# Display the cleaned DataFrame's first few rows
flight_hault_cleaned = flight_hault[['Name', 'City']]
flight_hault_cleaned.head()
for_g = nx.from_pandas_edgelist(flight_hault_cleaned,
                              source='Name',
                              target='City',
                              create_using=nx.DiGraph())
degree_centrality = nx.degree_centrality(for_g)
closeness_centrality = nx.closeness_centrality(for_g)
eigenvector_centrality = nx.eigenvector_centrality(for_g)
betweenness_centrality = nx.betweenness_centrality(for_g)
pos = nx.spring_layout(for_g)
nx.draw(for_g, pos, node_size=20, node_color='green')
plt.show()
```

Problem statement 2:
There are three datasets given (Facebook, Instagram, and LinkedIn). Construct and visualize the following networks:

- circular network for Facebook
- star network for Instagram
- star network for LinkedIn

Create a network using an adjacency matrix (undirected only). The snapshots of those datasets are given below:

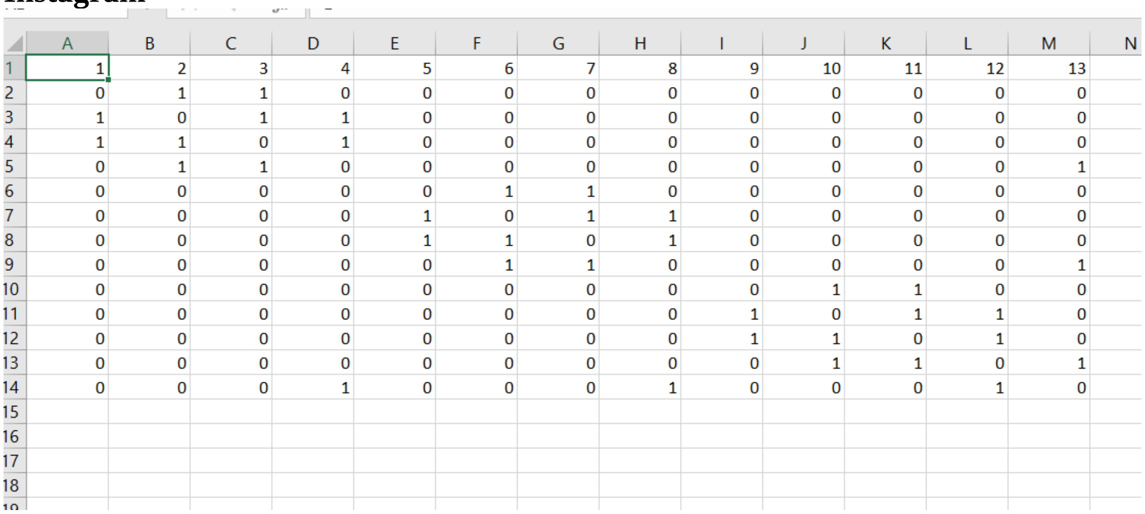
Facebook



	A	B	C	D	E	F	G	H	I
1	1	2	3	4	5	6	7	8	9
2	0	1	0	0	0	0	0	0	1
3	1	0	1	0	0	0	0	0	0
4	0	1	0	1	0	0	0	0	0
5	0	0	1	0	1	0	0	0	0
6	0	0	0	1	0	1	0	0	0
7	0	0	0	0	1	0	1	0	0
8	0	0	0	0	0	1	0	1	0
9	0	0	0	0	0	0	1	0	1
10	1	0	0	0	0	0	0	1	0

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load Facebook data
facebook_data = pd.read_csv("facebook.csv")

# Ensure row and column labels match
facebook_data.columns = facebook_data.index

# Create the undirected graph
facebook_graph = nx.from_pandas_adjacency(facebook_data)

# Plot in a circular layout
plt.figure(figsize=(8, 8))
nx.draw_circular(facebook_graph, with_labels=True, node_size=500, node_color="skyblue",
font_size=10, font_weight="bold")
plt.title("Facebook Circular Network")
plt.show()
```

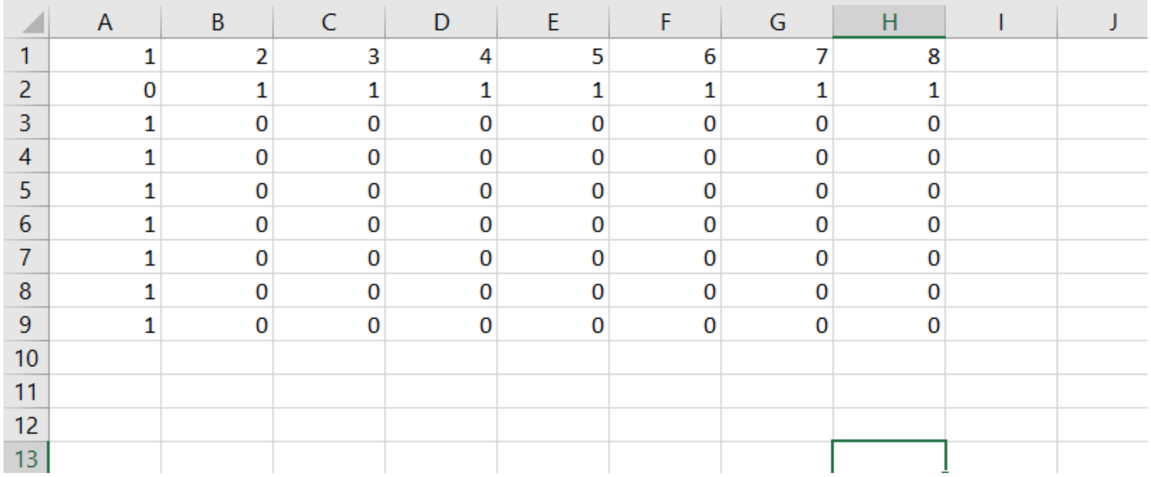
Instagram



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	2	3	4	5	6	7	8	9	10	11	12	13	
2	0	1	1	0	0	0	0	0	0	0	0	0	0	
3	1	0	1	1	0	0	0	0	0	0	0	0	0	
4	1	1	0	1	0	0	0	0	0	0	0	0	0	
5	0	1	1	0	0	0	0	0	0	0	0	0	1	
6	0	0	0	0	0	1	1	0	0	0	0	0	0	
7	0	0	0	0	1	0	1	1	0	0	0	0	0	
8	0	0	0	0	1	1	0	1	0	0	0	0	0	
9	0	0	0	0	0	1	1	0	0	0	0	0	1	
10	0	0	0	0	0	0	0	0	0	1	1	0	0	
11	0	0	0	0	0	0	0	0	1	0	1	1	0	
12	0	0	0	0	0	0	0	0	1	1	0	1	0	
13	0	0	0	0	0	0	0	0	0	1	1	0	1	
14	0	0	0	1	0	0	0	1	0	0	0	1	0	
15														
16														
17														
18														

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load Instagram data
instagram_data = pd.read_csv("instagram.csv")

# Assume row 0 is the central node and connects to all others
instagram_graph = nx.Graph()
central_node = 0
for i in range(1, len(instagram_data)):
    instagram_graph.add_edge(central_node, i)

# Plot in a star layout
plt.figure(figsize=(8, 8))
pos = nx.spring_layout(instagram_graph, center=(0,0))  # Position nodes in a star pattern
nx.draw(instagram_graph, pos, with_labels=True, node_size=500, node_color="lightgreen",
font_size=10, font_weight="bold")
plt.title("Instagram Star Network")
plt.show()
```

LinkedIn



	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8		
2	0	1	1	1	1	1	1	1		
3	1	0	0	0	0	0	0	0		
4	1	0	0	0	0	0	0	0		
5	1	0	0	0	0	0	0	0		
6	1	0	0	0	0	0	0	0		
7	1	0	0	0	0	0	0	0		
8	1	0	0	0	0	0	0	0		
9	1	0	0	0	0	0	0	0		
10										
11										
12										
13										

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load LinkedIn data
linkedin_data = pd.read_csv("linkedin.csv")

# Ensure columns match indices
linkedin_data.columns = linkedin_data.index

# Create a NetworkX graph
linkedin_graph = nx.from_pandas_adjacency(linkedin_data)
```

```
# Circular layout visualization
plt.figure(figsize=(8, 8))
nx.draw_circular(linkedin_graph, with_labels=True, node_size=500, node_color="skyblue",
font_size=10)
plt.title("LinkedIn Circular Network")
plt.show()
```