A Case Study of International Airline routes analysis

Team members:

DHANVARSHINI G : CB.PS.I5DAS22114

GOPI M : CB.PS.I5DAS22114

VIDHUN KS : CB.PS.I5DAS22162

Name : A Case Study of International Airline analysis Using Airline Dataset

Abstract :

This study analyzes a comprehensive airline dataset using graph theory to analyze airline routes and networks. The dataset contains information on airports, flight connections, and airline operators, forming a weighted graph where airports represent nodes and flight routes act as edges with distances as weights. Various preprocessing steps were applied to clean and structure the data for efficient analysis. Using network analysis algorithms, the shortest path between airports was determined to provide the most efficient travel routes. The study also explored centrality measures to identify key airports and influential hubs within the network. This analysis offers valuable insights into airline connectivity, operational efficiency, and potential optimization strategies for route management. The results can assist airline companies in decision-making and improving passenger experience.

Data Information:

Airline ID: *A unique identifier assigned to each airline.*

Name_Airline: *The official name of the airline.*

Country_Airline: *The country where the airline is registered or based.*

Active*: Indicates whether the airline is currently operating, typically represented as 'Y' for active or 'N' for inactive.*

Source Airport ID*: A unique identifier for the airport where the flight originates.*

Destination Airport ID*: A unique identifier for the airport where the flight terminates****.***

Source_Airport: *The name of the airport where the flight starts.*

City_Source: *The city where the source airport is located.*

Country_Source: *The country where the source airport is located.*

Latitude_Source: *The geographical latitude coordinate of the source airport.*

Longitude_Source: *The geographical longitude coordinate of the source airport.*

Destination_Airport: *The name of the airport where the flight ends.*

City_Destination: *The city where the destination airport is located.*

Country_Destination: *The country where the destination airport is located.*

Latitude_Destination: *The geographical latitude coordinate of the destination airport****.***

Longitude_Destination: *The geographical longitude coordinate of the destination airport.*

Stops: *The number of stops or layovers the flight has between the source and destination airports.*

Equipment: *The aircraft model or type used for the flight route.*

What do we Learn from the Dataset ?

Airline Network Insights:

The dataset provides valuable insights into the global airline network, including how airlines connect various cities and countries. By analyzing the routes, we can identify the major hubs and the most frequently used flight paths. This information helps in understanding airline connectivity patterns and how passengers are transported across different regions.

Flight Connectivity and Route Efficiency:

Using the dataset, we can determine the shortest paths between any two airports. Understanding the shortest route in terms of distance helps in evaluating the operational efficiency of airline networks. It also allows us to optimize travel plans, reduce travel time, and enhance passenger convenience.

Airline Operations and Coverage:

The dataset includes information about airline companies, their country of origin, and their operational status. By analyzing the "Active" status column, we can differentiate between active and inactive airlines, providing insights into which airlines dominate specific regions. Additionally, understanding the types of aircraft (Equipment) used on particular routes gives us a glimpse into airline fleet management and capacity planning.

Geographical Analysis:

With latitude and longitude data available for both source and destination airports, we can perform spatial analysis to visualize the airline routes on a map. This helps in identifying the most connected cities, regions with limited connectivity, and potential opportunities for expanding airline networks.

Route Complexity and Stopovers:

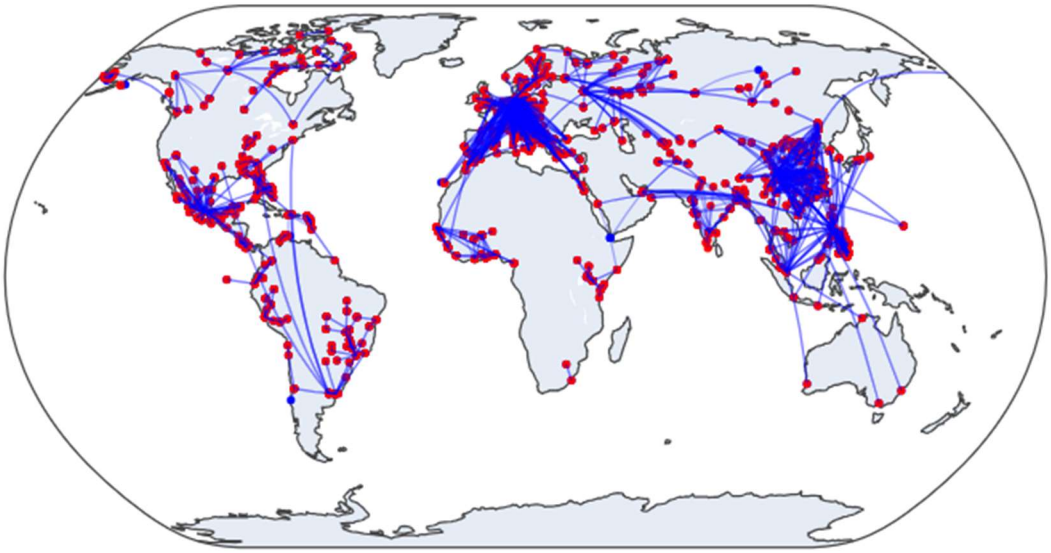
The number of stops on a flight is an essential factor in determining route complexity. By analyzing the "Stops" column, we can identify which routes are direct and which involve layovers. This information is useful for airlines looking to optimize their schedules and reduce unnecessary stopovers.

International and Domestic Travel Trends:

By examining the countries of the source and destination airports, we can study international and domestic travel patterns. This analysis can be valuable for tourism boards, aviation authorities, and airline companies to plan their operations strategically.

Decision Support for Passengers and Airlines:

Passengers can benefit from the shortest path analysis by choosing the most efficient routes with minimal layovers. Airlines, on the other hand, can use this analysis to adjust their network to improve connectivity and reduce operational costs.



Geoplot for all airports with routes

Graph Centralities Analysis using Airline Route Data

Degree Centrality:

Definition*: Degree centrality measures the number of direct connections a node has. Nodes with higher degree centrality are major hubs within the network. Airports with high degree centrality indicate well-connected airports that serve as central transit points. These are typically large international airports or major domestic hubs. Understanding degree centrality helps airlines optimize their route networks and allocate resources effectively.*

Inference:

- *Airports with higher degree centrality act as major hubs.*
- *These airports connect to a large number of destinations.*

- *Identifying these hubs is crucial for optimizing airline operations and increasing connectivity.*

Betweenness Centrality:

Definition*: Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes. Nodes with high betweenness centrality act as critical connectors or bridges. In an airline network, these airports are vital for connecting different regions and ensuring network robustness.*

Inference*:*

- *Airports with high betweenness centrality act as intermediaries.*
- *They play a significant role in maintaining connectivity within the network.*
- *Disruption at these airports can severely impact network flow.*
- *They are often strategic transit hubs for long-haul international flights.*

- 

Closeness Centrality:

Definition*: Closeness centrality calculates how quickly a node can reach all other nodes in the network. It indicates how efficiently a node can spread information. Airports with high closeness centrality are well-positioned for quick access to other airports, making them ideal for short-haul or regional connectivity.*

Inference*:*

- *Airports with high closeness centrality are well-positioned geographically.*
- *They provide efficient access to other airports.*
- *Such airports are suitable candidates for new airline hubs.*
- *High closeness centrality often characterizes central regional airports or logistical hubs.*

Eigenvector Centrality:

Definition: *Eigenvector centrality measures a node's influence based on the importance of its neighbors. High eigenvector centrality indicates that a node is connected to other influential nodes. Airports with high eigenvector centrality are often part of major airline alliances or key transit points in global aviation networks.*

Inference:

- *High eigenvector centrality indicates airports that connect to other influential hubs.*
- *They often facilitate high traffic and serve as major transit points.*
- *These airports enhance the network's efficiency and are essential for international airline operations.*
- *They contribute to economic and logistic growth due to their connectivity*.

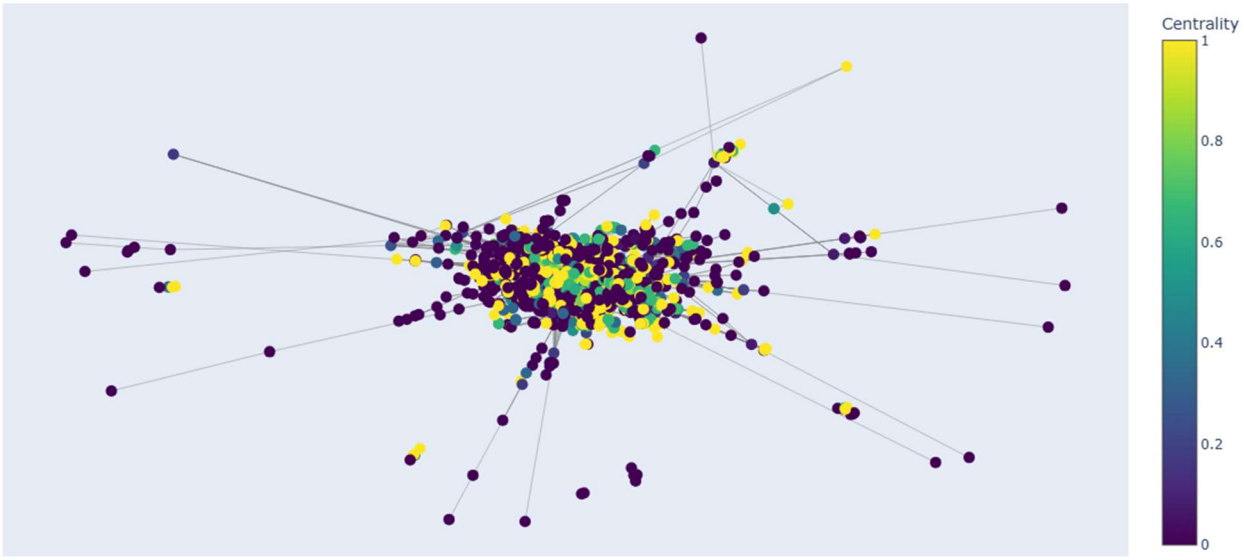
Clustering Coefficient:

Definition: *Clustering coefficient measures the tendency of nodes to form tightly knit groups. It reflects the extent of local connectivity. A high clustering coefficient often indicates the presence of regional airline networks with frequent short-haul flights*.

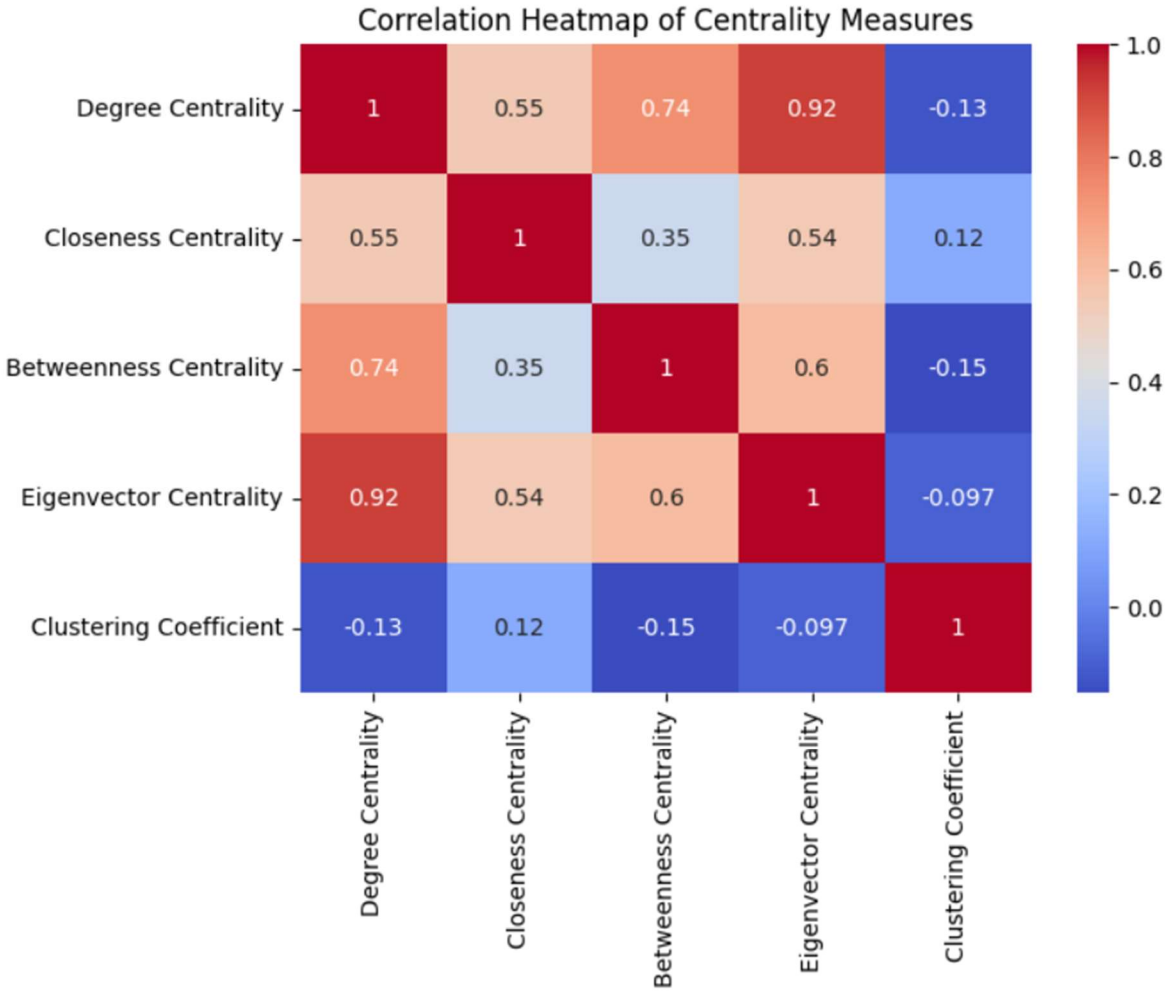
Inference:

- *A high clustering coefficient suggests localized clusters of interconnected airports.*
- *This is common in regional or domestic networks.*
- *Airports in areas with strong economic ties or tourism connections often exhibit high clustering coefficients.*
- *Low clustering coefficient might suggest isolated airports with fewer connections.*

Clustering Coefficient



Correlation Heatmap



Degree Centrality vs. Betweenness Centrality (0.74)

- *A strong positive correlation suggests that airports with a high number of direct connections often act as critical intermediaries in the network. This indicates that major hubs not only serve numerous direct routes but also facilitate long-distance connections.*

Degree Centrality vs. Closeness Centrality (0.55)

- *A moderate positive correlation shows that airports with higher degree centrality tend to have better access to all other airports, reaching destinations efficiently. However, geographic factors may limit the strength of this relationship.*

Degree Centrality vs. Eigenvector Centrality (0.92)

- *A very strong positive correlation implies that well-connected airports are often linked to other influential airports. Major international airports typically exhibit both high degree and eigenvector centrality, forming core components of the global aviation network.*

Betweenness Centrality vs. Closeness Centrality (0.35)

- *A weak correlation suggests that while some airports act as intermediaries, they may not necessarily have the shortest average path to all other nodes. This is typical for airports that serve as regional transfer points rather than direct destination hubs.*

Betweenness Centrality vs. Eigenvector Centrality (0.60)

- *A moderate correlation indicates that some critical connector airports are also highly influential. However, not all influential airports are necessarily the primary transit points, especially in regional clusters.*

Clustering Coefficient Correlations (Negative or Low)

- *The clustering coefficient shows weak or negative correlations with other centralities, particularly with degree centrality (-0.13) and betweenness centrality (-0.15).*
- *This suggests that highly connected hubs typically do not form tightly-knit local clusters, as their connections are often spread globally rather than regionally. Airports in regional networks may show higher clustering coefficients, forming dense clusters with frequent short-haul connections.*

Shortest Path Analysis:

Description: *The shortest path between two airports in the airline network was determined using Dijkstra's algorithm. The distances between airports were calculated using the* ***Haversine formula****, which provides the great-circle distance between two points on the Earth's surface based on their latitude and longitude.*

Methodology:

1. ***Distance Calculation:*** *The Haversine formula was applied to determine the geographic distance between each pair of connected airports.*
2. ***Graph Construction:*** *A directed, weighted graph was created using NetworkX. Airports served as nodes, and flight routes served as edges with weights representing distances.*
3. ***Shortest Path Computation:*** *Dijkstra's algorithm was used to compute the shortest path between the source and destination airports, minimizing the total travel distance.*
4. ***Error Handling:*** *If no path was found, the algorithm returned an appropriate error message.*

Final code:

```
import numpy as np

def haversine(lat1, lon1, lat2, lon2):
    R = 6371  # Earth radius in kilometers
    lat1, lon1, lat2, lon2 = map(np.radians, [lat1, lon1, lat2, lon2])
    dlat = lat2 - lat1
    dlon = lon2 - lon1
    a = np.sin(dlat/2)**2 + np.cos(lat1) * np.cos(lat2) * np.sin(dlon/2)**2
    c = 2 * np.arctan2(np.sqrt(a), np.sqrt(1-a))
    return R * c

# Apply the Haversine function to calculate distances
df['Distance (km)'] = df.apply(lambda x: haversine(x['Latitude_Source'], x['Longitude_Source'],
                                                   x['Latitude_Destination'], x['Longitude_Destination']), axis=1)

print("Distance calculation complete. Sample Data:")
print(df[['source_airport', 'destination_airport', 'Distance (km)']].head(20))
```

Calculate airport distances using Haversine formula for accurate measurements.

```python
import networkx as nx
from networkx.exception import NetworkXNoPath

def search_airport_name(query, graph):
    query = query.strip().lower()
    return [airport for airport in graph.nodes if query in airport.lower()]

source_airport = input("✈️ Enter Source Airport Name: ").strip()
destination_airport = input("🛬 Enter Destination Airport Name: ").strip()

for airport, label in [(source_airport, "Source"), (destination_airport, "Destination")]:
    if airport not in G:
        print(f"⚠️ '{airport}' not found in the graph.")
        matches = search_airport_name(airport, G)
        print("Did you mean one of these?" if matches else "No similar airports found.")
        for match in matches:
            print(f" - {match}")
        break
else:
    try:
        shortest_path = nx.shortest_path(G, source=source_airport, target=destination_airport, weight='weight')
        total_distance = nx.shortest_path_length(G, source=source_airport, target=destination_airport, weight='weight')
        print(f"\n🌐 Shortest Path from {source_airport} to {destination_airport}:\n")
        for i in range(len(shortest_path) - 1):
            u, v = shortest_path[i], shortest_path[i + 1]
            airline_name = G[u][v].get('Name_Airline', 'Unknown Airline')
            print(f"{u} ➡ {v} (Airline: {airline_name})")
        print(f"\nTotal Distance: {total_distance:.2f} km")
    except NetworkXNoPath:
        print(f"❗ No path found between {source_airport} and {destination_airport}.")
```

Find the shortest path between airports using NetworkX with distance-based analysis.

Final example output :

```
✈️ Enter Source Airport Name: Chennai International Airport
🛬 Enter Destination Airport Name: London City Airport
🌐 Shortest Path from Chennai International Airport to London City Airport:

Chennai International Airport ➡ Frankfurt am Main Airport (Airline: Lufthansa)
Frankfurt am Main Airport ➡ London City Airport (Airline: Lufthansa)

Total Distance: 8214.26 km
```

This code calculates the shortest path between two airports using the Haversine formula to measure distances. It models the network as a directed graph using NetworkX, where nodes represent airports and edges represent flight routes with associated airline information and distances.

- *For example: If we search for the shortest route from Chennai International Airport to London City Airport, the output suggests a path via Frankfurt am Main Airport using Lufthansa Airline.*
- *The total travel distance is 8214.26 km, showcasing the efficiency of the shortest path algorithm.*