CADT
IDT វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល
Institute of Digital Technology

Aircraft Dataset Visualization

DATA SCIENCE -INFORMATION VISUALIZATION USING POWERBI

NOVEMBER 2023

AUTHOR:
SIM THYDATEPIN
PUN SOLITA
SOM DEBORAH
KONG DARACHIN

Table of Contents

List of Figure

List of Tables

I. Introduction

1. Project Objective

This project's main goal is to analyze and visualize an airline dataset in order to get insights about the globally flying business. This involves investigating various types of aircraft, airports, flights, and reservations. We will also look at how flights are distributed among cities and nations. We will also look into trends in flight status, actual departure vs. scheduled departure, and the most prevalent aircraft types utilized on particular routes. This investigation' conclusions will be useful to stakeholders in the aviation sector, such as airlines, airports, and travel agents.

2. Expected Delivery

A comprehensive analysis of the aviation dataset is proposed, aiming to uncover valuable insights into the global flight industry. This in-depth examination will delve into the characteristics of various aircraft, airports, flights, and bookings, shedding light on the patterns and trends shaping the aviation landscape.

Additionally, the distribution of flights across cities and countries will be scrutinized, revealing the geographical dynamics of air travel.

Furthermore, flight status trends, actual departure versus scheduled departure patterns, and the most prevalent aircraft types for different routes will be investigated, providing a nuanced understanding of operational efficiency and fleet utilization.

The findings of this analysis will culminate in a comprehensive report that distills the key takeaways and formulates actionable recommendations for stakeholders in the aviation industry, including airlines, airports, and travel agencies.

II. Data Description

1. Size

travel.sqlite (109.53 MB)

Table	Total Rows	Total Columns
aircrafts_data	9	3
airports_data	104	5
boarding_passes	579686	4
bookings	262788	3
flights	33121	10
seats	1339	3
ticket_flights	1045726	4
tickets	366733	3

Table 1 Size

This figure shows the size of this dataset which is 109.53 MB. It has a total of eight tables. Here are the description of each table:

- Aircrafts_data:

Column Name	Data Type	Description
aircraft_code	character(3)	Code for the aircraft
model	jsonb	Aircraft model in JSON format
range	integer	The range of the aircraft

Table 2 Aircraft data

- Airports_data

Column Name	Data Type	Description
airport_code	character(3)	Code for the airport
airport_name	jsonb	Name of the airport in JSON format
city	jsonb	City where the airport is located
coordinates	point	Geographic coordinates of the airport
timezone	text	Timezone of the airport

Table 3 Airports data

- Boarding_passes

Column Name	Data Type	Description
ticket_no	character(13)	Ticket number
flight_id	integer	ID of the flight
boarding_no	integer	Boarding number
seat_no	character varying(4)	Seat number

Table 4 Boarding Passes

- Bookings

Column Name	Data Type	Description
book_ref	character(6)	Booking reference
book_date	timestamp with time zone	Booking date with timestamp and time zone
total_amount	numeric(10,2)	Total booking amount

Table 5 Booking

- Flights

Column Name	Data Type	Description
flight_id	integer	Flight ID
flight_no	character(6)	Flight number
scheduled_departure	timestamp with time zone	Scheduled departure time with timestamp and time zone
scheduled_arrival	timestamp with time zone	Scheduled arrival time with timestamp and time zone
departure_airport	character(3)	Departure airport code
arrival_airport	character(3)	Arrival airport code
status	character varying(20)	Flight status
aircraft_code	character(3)	Aircraft code
actual_departure	timestamp with time zone	Actual departure time with timestamp and time zone
actual_arrival	timestamp with time zone	Actual arrival time with timestamp and time zone

Table 6 Flight

- Seats

Column Name	Data Type	Description
aircraft_code	character(3)	Aircraft code
seat_no	character varying(4)	Seat number
fare_conditions	character varying(10)	Fare conditions

Table 7 Seats

- Ticket_flights

Column Name	Data Type	Description
ticket_no	character(13)	Ticket number
flight_id	integer	ID of the flight
fare_conditions	character varying(10)	Fare conditions
amount	numeric(10,2)	Ticket amount

Table 8 Ticket flights

- Tickets

Column Name	Data Type	Description
ticket_no	character(13)	Ticket number
flight_id	integer	ID of the flight
fare_conditions	character varying(10)	Fare conditions
amount	numeric(10,2)	Ticket amount

Table 9 Tickets

2. Data Relation

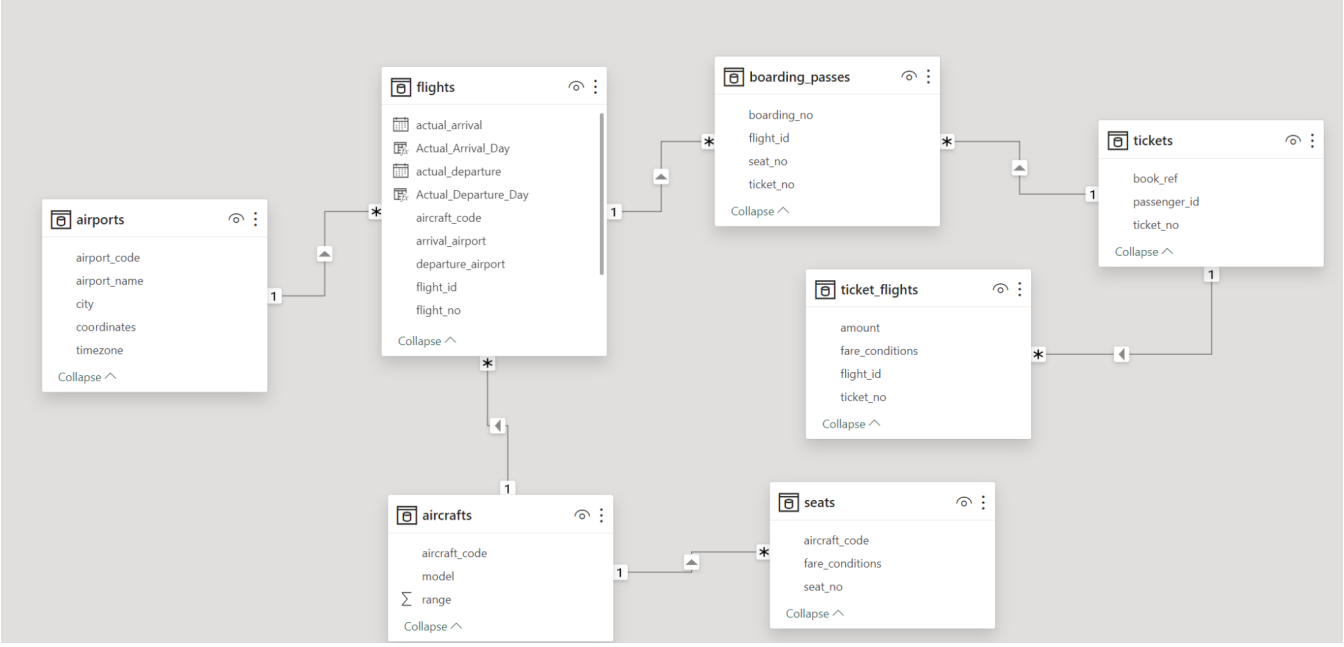

Figure 1 Data Relation

III. Implementation

1. Connecting Dataset to Powerbi

Here are some Steps to connect SQL to Powerbi

1. Open wamp Server

Figure 2 Wamp Server

2. Open Powerbi

Figure 3 Powerbi

3. Click on Get Data

Figure 4 Get Data

4. Choose MySQL Database

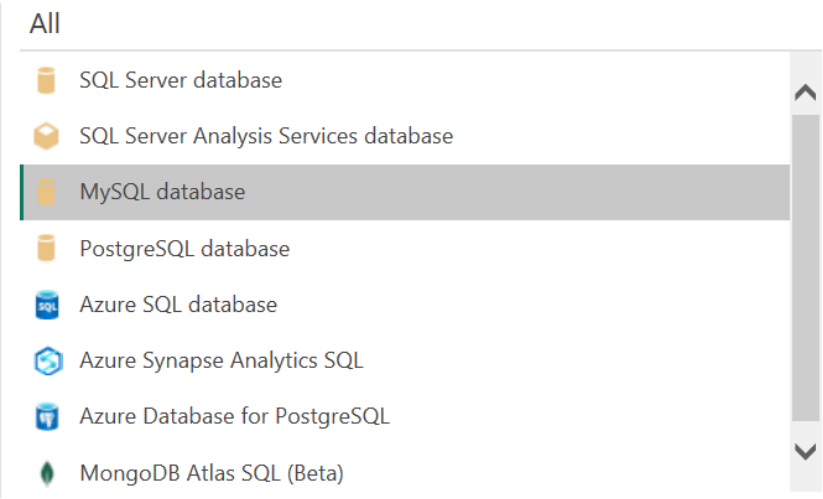

Figure 5 MySql Database

5. Input Server name and Database name

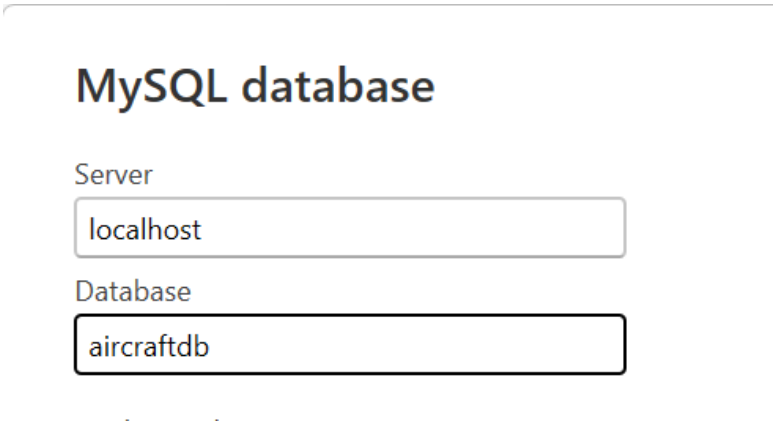

Figure 6 Input Server and Database

6. Then click OK

We have connected the dataset to power bi by using MySQL but in power bi we noted that data has lost a lot. So we have decide to using python code to create APIs and connect with power bi. With these APIs we can get all the data from the data set. Here are the step that we use python code to connect with power bi:

1. Set up project and import library

   The first one we create python code and the dataset file in one folder. And import library as:

```
import pandas as pd
import sqlite3
import json
from flask import Flask, jsonify
from flask_socketio import SocketIO
```

   Which pandas we can used dataframe in python, sqlite3 we used to connect with dataset file, json we used to convert dataframe to json,

   Flash and SocketIO used to create APIs and set routes for each table in the dataset.

2. Convert table in dataset to dataframe

```
app = Flask(__name__)
socketio = SocketIO(app)

# Read sqlite query results into a pandas DataFrame
with sqlite3.connect("travel.sqlite") as con:
    aircrafts_data = pd.read_sql_query("SELECT * from aircrafts_data", con)
    airports_data = pd.read_sql_query("SELECT * from airports_data", con)
    boarding_passes = pd.read_sql_query("SELECT * from boarding_passes", con)
    flights = pd.read_sql_query("SELECT * from flights", con)
    seats = pd.read_sql_query("SELECT * from seats", con)
    ticket_flights = pd.read_sql_query("SELECT * from ticket_flights", con)
    tickets = pd.read_sql_query("SELECT * from tickets", con)
```

The first one we create an app with Flask and set socketio for the app. Then we read the dataset by using sqlite3 library and query to get each table by using pandas and set them to the dataframe.

3. Convert dataframe to json

```
# JSON data for different types
aircrafts_json = aircrafts_data.to_json(orient='records', indent=2)
airports_json = airports_data.to_json(orient='records', indent=2)
boarding_passes_json = boarding_passes.to_json(orient='records', indent=2)
flights_json = flights.to_json(orient='records', indent=2)
seats_json = seats.to_json(orient='records', indent=2)
ticket_flights_json = ticket_flights.to_json(orient='records', indent=2)
tickets_json = tickets.to_json(orient='records', indent=2)
```

We convert dataframe to json by using pandas dataframe. Which syntax is dataframe.to_json().

4. Create routes and get API

```
@app.route('/api/tickets')
def get_tickets_api():
    return jsonify(tickets.to_dict(orient='records'))

if __name__ == '__main__':
    socketio.run(app, debug=True, port=5000, allow_unsafe_werkzeug=True)
```

We create routes with return json in API route for each data that we have converted from dataframe to json format. Then we run an app with socketio in port=5000.

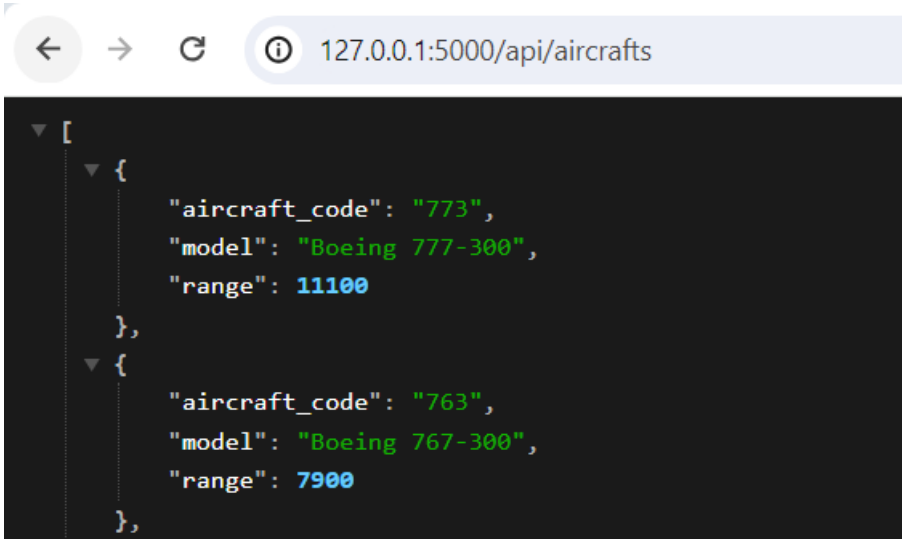

Figure 7 API Route

So the data return as Figure 7.

5. Connect to power bi

We can connect this api to power bi by going to get data and select web and copy our api url to patch in URL text area then click okay.

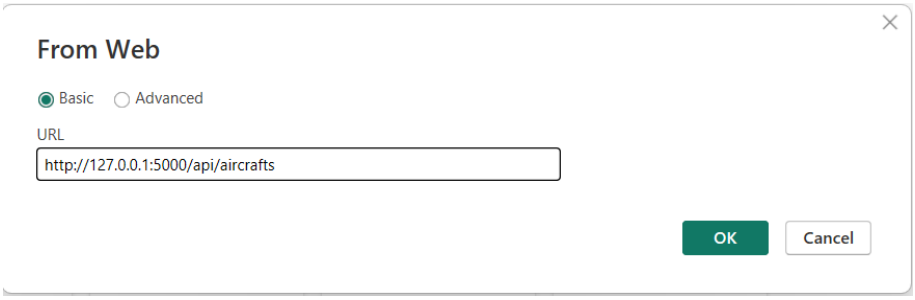

Figure 8 Web to Powerbi

So the data from our API is now in power bi.

	aircraft_code	model	range
1	773	{"en": "Boeing 777-300", "ru": "Боинг 777-300"}	11100
2	763	{"en": "Boeing 767-300", "ru": "Боинг 767-300"}	7900
3	SU9	{"en": "Sukhoi Superjet-100", "ru": "Сухой Суперджет-100"}	3000
4	320	{"en": "Airbus A320-200", "ru": "Аэробус A320-200"}	5700
5	321	{"en": "Airbus A321-200", "ru": "Аэробус A321-200"}	5600
6	319	{"en": "Airbus A319-100", "ru": "Аэробус A319-100"}	6700
7	733	{"en": "Boeing 737-300", "ru": "Боинг 737-300"}	4200
8	CN1	{"en": "Cessna 208 Caravan", "ru": "Сессна 208 Караван"}	1200
9	CR2	{"en": "Bombardier CRJ-200", "ru": "Бомбардье CRJ-200"}	2700

Figure 9 API to Powerbi

2. Clean and Prepare Data for Visualization

```
# Clean dictionary from data
aircrafts_data['model'] = aircrafts_data['model'].apply(lambda x: json.loads(x).g
airports_data['airport_name'] = airports_data['airport_name'].apply(lambda x: jso
airports_data['city'] = airports_data['city'].apply(lambda x: json.loads(x).get('

# Handle the missing with expect values
flights['actual_departure'] = flights.apply(lambda row: row['scheduled_departure'
flights['actual_arrival'] = flights.apply(lambda row: row['scheduled_arrival'] if
```

We observe that our data has one table (flights) missing in a column (actual_departure and actual_arrival) so decide to handle this by replacing the value of (scheduled_departure and scheduled_arrival).

For table aircraft_data we observe the data is returned as a dictionary that has data in different languages which we can't use in power bi so we decide to get only one language in that table.

So now data in all tables can be used in power bi.

	aircraft_code	model	range
1	773	Boeing 777-300	11100
2	763	Boeing 767-300	7900
3	SU9	Sukhoi Superjet-100	3000
4	320	Airbus A320-200	5700
5	321	Airbus A321-200	5600
6	319	Airbus A319-100	6700
7	733	Boeing 737-300	4200
8	CN1	Cessna 208 Caravan	1200
9	CR2	Bombardier CRJ-200	2700

Figure 10 Aircraft Data in Powerbi

3. Arrival City

Figure 11 Arrival City

There are 101 cities on this dataset and each city has one airport except for Moscow and Ulyanovsk which have 3 and 2 airports respectively.

4. Total Destination

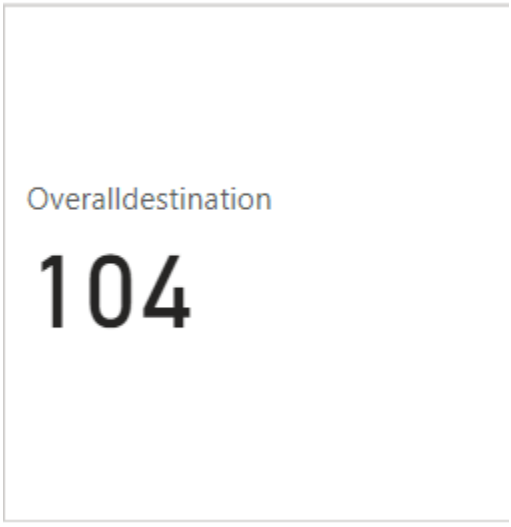

Figure 12 Overall Destination

There are 104 overall Destinations.

We created this measure by using Dax Formula

Syntax: Overalldestination = COUNTROWS(airports)

By using “Card” visualization and choosing this measure as data.

5. Total Number of flights

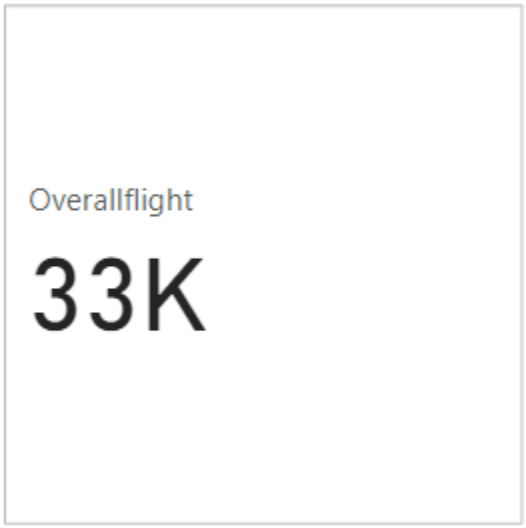

Figure 13 Overall Flight

There are 33k overall flights.

We created this measure by using Dax Formula

Syntax: Overallflight = COUNTROWS(flights)

By using “Card” visualization and choosing this measure as data.

6. Airport that has the most flights

Figure 14 Airport has most flight

On this graph we can say that the airport that has the most flights is **Domodedovo International Airport** has 3217 flights in the overall flight from July to September in 2017 which is located at **MOSCOW** the capital of Russia.

7. Overall Flights each day

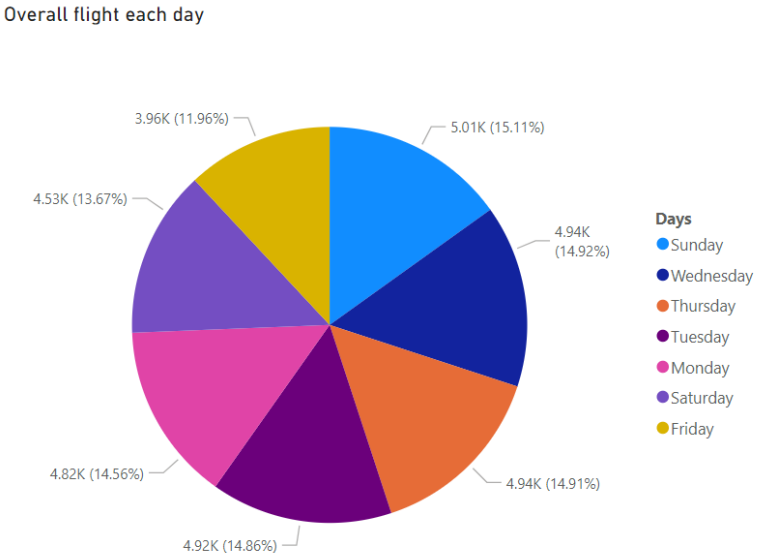

Figure 15 Overall flight each day

Based on this pie chart we can say that the day which has the most flights is on Sunday around 5006 over all flights. If we compare it to Friday, we saw that on Friday there are fewer flights around 3961 flights.

8. City that have most airports

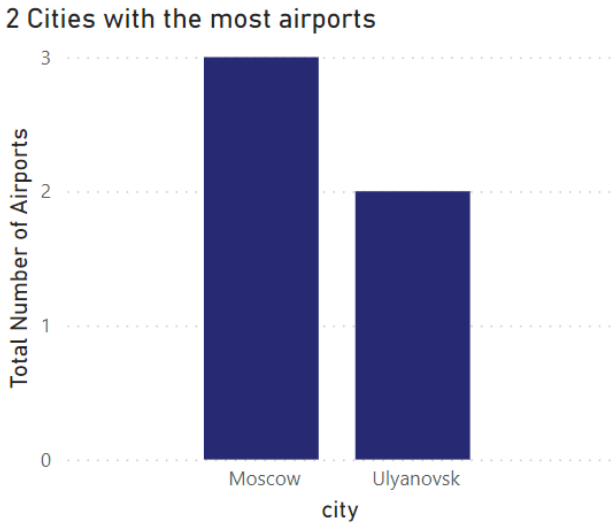

Figure 16 City with most Airports

9. Airplane fly the most each month

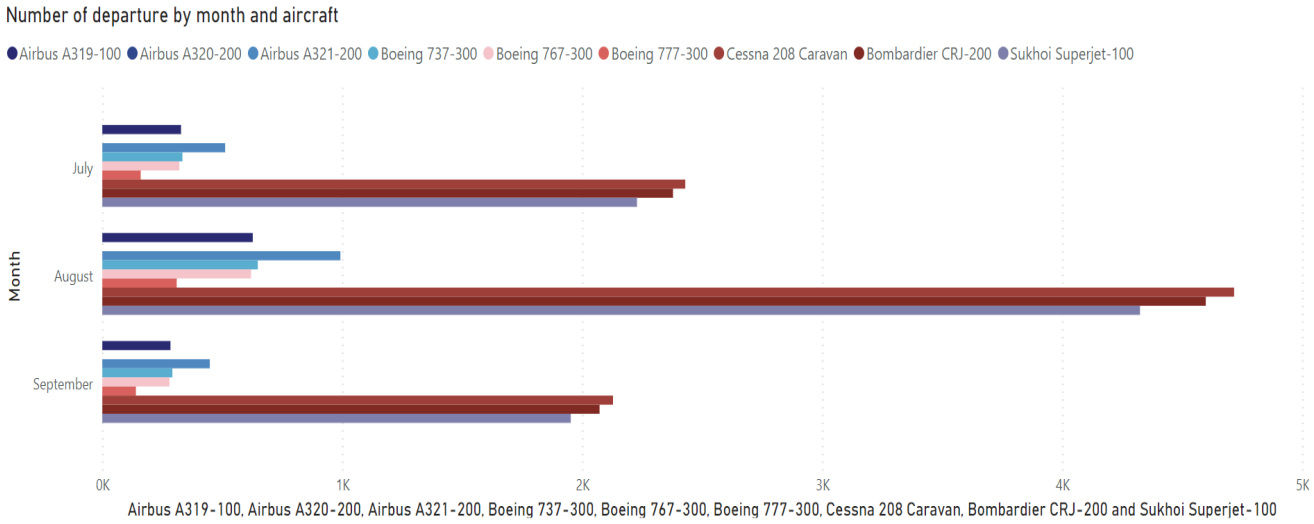

Figure 17 Airplane fly the most each month

In August there are twice as many flights compared to other months, with the Cessna 208 Caravan having the highest number of passengers.

10. The Percentage of Flight Status

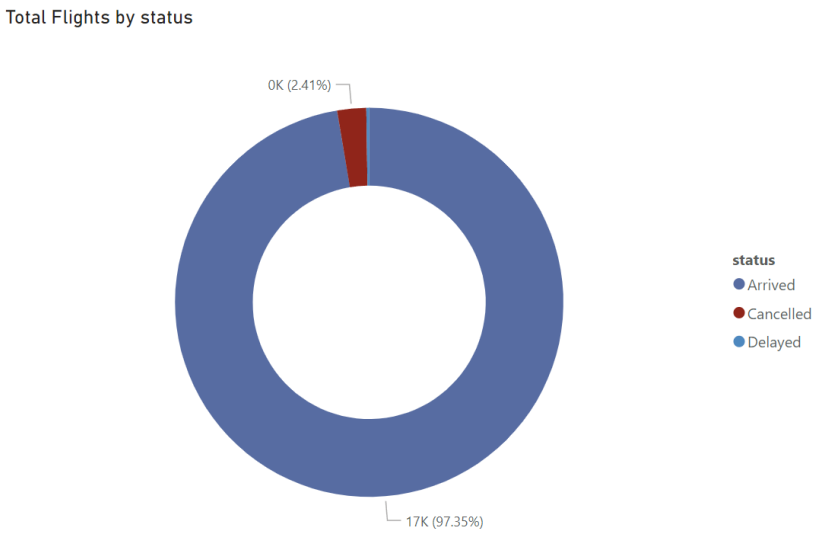

Figure 18 Percentage of Flight Status

Most flights are able to arrive safely and on schedule, only a small percentage of flights get canceled or delayed.

IV. Conclusion and Recommendation

1. Conclusion

In conclusion, from the data given we are able to give general information on the flights' general information such as : number of aircrafts, number of airports, number of cities with airports, number of bookings and types of fare etc. From these we can compare values and give a straightforward and clear summary of the data like which aircraft is most popular, which months have the highest ticket sales and which fare is most bought so that we can take actions according to the data to make the most profit for the company.

2. Recommendation

To enhance operational efficiency and improve customer satisfaction, airlines should consider implementing real-time flight status updates and implementing more proactive measures to mitigate flight delays or cancellations. Additionally, airlines should analyze the most common types of aircraft used for different routes and consider optimizing their fleet composition to better align with passenger demand and route characteristics. Moreover, airports should focus on infrastructure improvements and capacity expansion to accommodate increasing passenger traffic and enhance the overall airport experience. Additionally, airports should collaborate with airlines to streamline check-in and boarding processes to optimize passenger flow and reduce waiting times. Finally, travel agencies should leverage the insights gained from analyzing the aviation dataset to provide more personalized travel recommendations and enhance customer service by offering real-time flight status updates and proactive assistance in case of disruptions.

Reference

https://www.kaggle.com/datasets/saadharoon27/airlines-dataset