NYC FLIGHTS – BUSINESS ANALYTICS

TECHNICAL REPORT

Analytics Objective:

We would like to gain insights in the aspect of delays for our airlines operating from the major New York City Airports, thus aiming to boost our overall customer experience.

Tools & Technologies:

1) **Structured Query Language (SQL)** - Pre-processing the dataset
2) **Tableau** - Creating Visual Insights

Pre-Processing the Dataset using SQL:

The data comprises 5 datasets related to air flights, all leaving from one of the NYC airports in 2013. To gain insights from all the aspects of the collected flight data, we are performing merging on several datasets based on the unique and foreign keys.

The SQL query for the same is as follows,

```
PROC SQL;

CREATE TABLE nyc.BigMerge as
SELECT DISTINCT a.carrier as CarrierName, a.name as Airlines,
                ap.name as AirportName,

                f.origin as OriginAirport, f.dest as DestinationAirport,

                f.month as Month, f.day as Day, f.hour as Hour, f.flight as Flight, f.distance as
                TravelDistance,

                f.dep_delay as DepartureDelay, f.arr_delay as ArrivalDelay,

                sum(f.dep_delay) + sum(f.arr_delay)as TotalDelay,

                w.temp as Temperature, w.dewp as DewPoint, w.humid as RelativeHumidity,
                w.wind_dir as WindDirection, w.wind_speed as WindSpeed, w.wind_gust as
                WindGustSpeed, w.precip as Precipitation, w.pressure as Pressure, w.visib as
                Visbility,

                p.year as FlightManufactureYear, p.type as FlightType, p.manufacturer as
                FlightManufacturer, p.model as FlightModel, p.engines as No_Of_Engines,
                p.speed as AvgCruiseSpeed, p.engine as EngineType



FROM nyc.AIRLINES as a, nyc.FLIGHTS as f, NYC.WEATHER as w,
        NYC.AIRPORTS as ap, NYC.PLANES as p
```

```
WHERE a.carrier = f.carrier and
      f.time_hour = w.time_hour and
      f.origin = w.origin and
      ap.faa = f.origin and
      f.tailnum = p.tailnum

GROUP BY 1,2,3,4,5,6,7,8,9

ORDER BY 1;

RUN;
```

Query Logic: Performing Huge Inner Join on All the Datasets

1) We are creating a new dataset in the tabular form
2) Selecting only the desired variables from all the datasets
3) Importing all the datasets to reference it for our new dataset
4) Inner join all the datasets using Unique & Foreign Keys
5) Grouping by major variables in the dataset
6) Ordering the dataset based on Ascending Carrier Name
7) Exporting the resultant huge query into a SAS Data Set

Creating Visual Insights in Tableau:

The data source for our data analysis in Tableau will be the newly exported SAS Dataset. Through intuitive visualizations, we would like to understand how different variables are correlated in terms of the flight delay.

The following are the major relationships discovered in the analysis,

- Express Jet Airlines contributes to the topmost average delay in all the three airports (John F Kennedy - 36.33 Mins; La Guardia – 28.37 Mins; Newark – 37.03 Mins)
- There is no significant relationship between the Delay & Destination travel distance
- The Average delay is predominantly high in June, July & December Months.
    - These might be due to the Repair work due after a hectic winter season
    - A lot of passenger traffic to Europe for summer vacation and NYC airports act as hub for most airlines
    - Hence, Seasonality plays a vital role for delays in these months
- There are peak delays in the Second & Last weeks of all months
- There are peak delays after 6PM and especially more in late night flights
- The worst route in terms of delay would be EWR -> CAE with 78.84 mins average delay
- Predominantly airlines utilizing 4 Cycle Engines are encountering more delays
- Gulfstream Aerospa is the topmost Flight Manufacturer contributing for delay
- There exists a Positive Relationship between the Wind Gust Speed & Delay
- There exists a Negative Relationship between the Visibility & Delay

The visualization for the same has been reported in the Tableau worksheets, dashboards and the story. Please click on the below provided link to access the same,
https://dub01.online.tableau.com/t/venkatstableau/authoring/NYC_Tableau_Dashboards/Story1/Delays%20over%20time#1

Action Plan:

Based on the thorough data analytics, we would like to propose the following recommendations to improve the overall customer experience by improving the delay time,

1) Benchmark Alaska Airlines, which has the maximum negative delay of -4.12 minutes
2) Predict seasonal demands to adjust the flight schedules and frequency accordingly
3) Consider shifting to Turbo Jet engines, which has only had an avg. delay of 13.84 minutes
4) Consider reducing Share of Business (S.O.B) for Gulfstream Aerospa Flight Manufacturer
5) Operate flights majorly based on the weather condition (Wind Gust Speed, Visibility)
6) Consider scheduling departures to off peak hours especially in the morning

Note: The following are the individual SQL queries for all the mandatory questions mentioned in the assignment description,

```
/* 1. Evaluating the delays for the different airlines */

PROC SQL;

drop view nyc.airlines_delay;
CREATE TABLE nyc.airlines_delay as
        SELECT a.carrier as CarrierName, a.name as Airlines, avg(dep_delay) as DepartureDelay,
                avg(arr_delay) as ArrivalDelay, avg(dep_delay) + avg(arr_delay) as TotalDelay
        FROM nyc.AIRLINES as a, nyc.FLIGHTS as f
        WHERE a.carrier = f.carrier
        GROUP BY 1,2
        ORDER BY 5 DESC;

RUN;

/* 2. Evaluating the delays depending on the destination airports and distances */

PROC SQL;

drop view nyc.destination_delay;
CREATE TABLE nyc.destination_delay as
        SELECT DISTINCT a.carrier as CarrierName, a.name as Airlines, f.origin as OriginAirport,
                        f.dest as DestinationAirport, f.distance as TravelDistance, sum(dep_delay)
                        as DepartureDelay, sum(arr_delay) as ArrivalDelay, sum(dep_delay) +
                        sum(arr_delay)as TotalDelay
        FROM nyc.AIRLINES as a, nyc.FLIGHTS as f
        WHERE a.carrier = f.carrier
        GROUP BY 1,2,3,4
        ORDER BY 7 DESC;

RUN;
```

```
/* 3a. Departure - Evaluating reasons for delays */

PROC SQL;

drop view nyc.departure_delay;
CREATE TABLE nyc.departure_delay as
        SELECT a.carrier as CarrierName, a.name as Airlines, w.time_hour as Time, f.origin as Origin,
                ap.name as DepartureAirport, f.dest as Destination,  f.dep_delay as DepartureDelay,
                w.temp as Temperature, w.dewp as DewPoint, w.humid as RelativeHumidity,
                w.wind_dir as WindDirection, w.wind_speed as WindSpeed, w.wind_gust as
                WindGustSpeed, w.precip as Precipitation, w.pressure as Pressure, w.visib as
                Visbility, p.year as FlightManufactureYear, p.type as FlightType, p.manufacturer as
        FlightManufacturer, p.model as FlightModel, p.engines as No_Of_Engines, p.speed as
        AvgCruiseSpeed, p.engine as EngineType

        FROM NYC.AIRLINES as a, NYC.FLIGHTS as f, NYC.WEATHER as w, NYC.AIRPORTS as ap,
              NYC.PLANES as p

        WHERE a.carrier = f.carrier and f.time_hour = w.time_hour and f.origin = w.origin and
                ap.faa = f.origin and
                 f.tailnum = p.tailnum
        GROUP BY 1,2,3,4,5,6
        ORDER BY 7 DESC;

RUN;

/* 3b. Arrival - Evaluating reasons for delays */

PROC SQL;

drop view nyc.arrival_delay;
CREATE TABLE nyc.arrival_delay as
        SELECT a.carrier as CarrierName, a.name as Airlines, w.time_hour as Time, f.origin as Origin,
                f.dest as Destination, ap.name as ArrivalAirport, f.dep_delay as DepartureDelay,
                w.temp as Temperature, w.dewp as DewPoint, w.humid as RelativeHumidity,
                w.wind_dir as WindDirection, w.wind_speed as WindSpeed, w.wind_gust as
                WindGustSpeed, w.precip as Precipitation, w.pressure as Pressure, w.visib as
                Visbility, p.year as FlightManufactureYear, p.type as FlightType, p.manufacturer as
                FlightManufacturer, p.model as FlightModel, p.engines as No_Of_Engines, p.speed as
               AvgCruiseSpeed, p.engine as EngineType

        FROM NYC.AIRLINES as a, NYC.FLIGHTS as f, NYC.WEATHER as w, NYC.AIRPORTS as ap,
                NYC.PLANES as p

        WHERE a.carrier = f.carrier and f.time_hour = w.time_hour and f.origin = w.origin and
                ap.faa = f.dest and f.tailnum = p.tailnum

        GROUP BY 1,2,3,4,5,6
        ORDER BY 7 DESC;

RUN;
```

```
/* 4. Changes in delays over time (Month, Day & Hour) */

PROC SQL;

drop view nyc.delay_over_time;

CREATE TABLE nyc.delay_over_time as

SELECT ap.name as AirportName, a.carrier as CarrierName, a.name as AirlinesName, f.month as
Month, f.day as Day, f.hour as Hour, count(f.flight) as No_Of_Flights, avg(f.dep_delay) as
DepartureDelay, avg(f.arr_delay) as ArrivalDelay,  avg(f.dep_delay) + avg(f.arr_delay) as
AverageDelay

FROM NYC.airports as ap, NYC.AIRLINES as a, NYC.FLIGHTS as f

WHERE a.carrier = f.carrier and ap.faa = f.origin

GROUP BY 1,2,3,4,5,6

ORDER BY 1 ASC, 4 ASC;

RUN;

/* 5. Plot the worst routes (routes with highest delays) */

PROC SQL;

drop view nyc.worst_routes;

CREATE TABLE nyc.worst_routes as

SELECT DISTINCT f.origin as DepartureAirport, f.dest as ArrivalAirport, f.carrier as CarrierName,
count(f.flight) as TotalFlights, avg(f.dep_delay) + avg(f.arr_delay) as AverageDelay,
sum(f.dep_delay) + sum(f.arr_delay) as TotalDelay

FROM NYC.FLIGHTS as f

GROUP BY 1,2

ORDER BY 1,2;

RUN;
```