AirFly Insights: Data Visualization and Analysis of Airline Operations

INTRODUCTION

The objective of this project is to analyse large-scale airline flight data to uncover operational trends, delay patterns, and cancellation reasons using data visualization techniques. The goal is to help understand airline and airport-level performance and contribute to actionable insights using visual analysis.

Week 1: Project Initialization and Dataset Setup

1. Define goals, KPIs, and workflow

- Goal: Analyze flight delays dataset to understand overall punctuality, cancellations, and distance-related insights.
- KPIs:
    - Average arrival delay
    - Average departure delay
    - Total number of cancellations per airline
    - Maximum distance flown per airline
- Workflow: Data loading → Exploration → Cleaning → Analysis → Insights.

2. Explore schema, types, size, and nulls

- Viewed first few rows using df.head() to understand dataset structure.
- Examined last 10 rows with df.tail(10) to verify consistency and completeness.
- Checked dataset size: 14,051,979 elements → gives an idea of data volume.
- Explored dataset shape: 484,551 rows × 29 columns → overview of dimensions.
- Examined data types (df.dtypes) and column names (df.columns) to understand feature nature.
- Used df.describe() and df.info() for statistical summaries, non-null counts, and memory usage.
- Checked missing values with df.isnull().sum() and duplicates with df.duplicated().sum() to ensure data quality.

3. Load CSVs using pandas

- Loaded dataset into a Pandas DataFrame using pd.read_csv("Flight_delay.csv").

4. Perform sampling and memory optimizations

- Calculated min, max, and average values of **Distance** column → understanding range and central tendency.
- Computed average **arrival delay** and **departure delay** → assessing overall flight punctuality.
- Used groupby to:
  - Find maximum distance per airline.
  - Calculate total cancellations per airline.
- These aggregations provided operational insights and helped reduce large data into summarized, manageable form.

WEEK 2: Preprocessing and Feature Engineering

1. Handle nulls in delay and cancellation columns

- Checked for missing values in each column using df.isnull().sum() to identify data quality issues.
- Handled missing values by replacing nulls in the **Org_Airport** and **Dest_Airport** columns with 'unknown', ensuring categorical completeness and avoiding errors in downstream analysis.
- Checked for duplicate rows using df.duplicated().sum() and removed them with df.drop_duplicates(keep='first'), ensuring consistency and preventing redundancy.
- Verified duplicates again to confirm that **0 duplicates** remained.

2. Create derived features: Month, Day of Week, Hour, Route

- Converted the **Date** column into datetime format using pd.to_datetime().
- Extracted new features: **Month**, **DayOfWeek**, and **Hour** to enable trend analysis across time.
- Created a new feature **Route** by combining **Origin** and **Dest**, supporting route-specific flight delay analysis.

3. Format datetime columns

- Ensured the **Date** column was properly formatted as a datetime object to facilitate time-based filtering, grouping, and trend visualization.

4. Save preprocessed data for fast reuse

- Saved the cleaned dataset as **Flight_delay_cleaned.csv**, ensuring a consistent and reusable file for further analysis without repeating preprocessing steps.