# Week2_AirFly_Insights

## Objectives

- Load and inspect the raw flight dataset safely from compressed format.

- Handle missing values, data type optimization, and memory efficiency.

- Parse and standardize time columns into proper datetime objects.

- Engineer new features useful for downstream analytics and modeling.

- Perform exploratory summaries: busiest routes, delays, cancellations, seasonal trends.

- Deliver 50 executable code snippets for reproducible, step-by-step data preparation.

## Tasks Completed

Data Loading & Inspection (Steps 1–5)

- Imported libraries (pandas, numpy, datetime), unzipped and inspected CSV contents.

- Loaded sample rows to preview schema, then loaded full dataset with explicit dtype hints to cut memory.

- Profiled missing values and generated descriptive statistics.

Datetime Parsing & Derived Features (Steps 6–14)

- Implemented robust hhmm_to_time parser for HHMM-encoded times.

- Created SCHEDULED_DEP datetime column with safe coercion of invalid entries.

- Derived new columns: month, day_of_week, hour, time_of_day.

- Built route identifiers (route, route_id).

- Processed delays: filled NaNs, created delay flags, computed derived delay minutes using scheduled vs actual times.

- Added is_cancelled and cancellation_code fields.

Feature Engineering & Optimization (Steps 15–20)

- Computed rolling average route delays (route_delay_roll5).

- Frequency encodings and category codes for categorical variables (carrier, origin, destination).

- Optimized memory via downcasting numeric columns.

- Dropped duplicates and checked anomalies in delays/times.

- Prepared 1% sample dataset for fast iteration.

Exploratory Summaries & Checks (Steps 21–50)

- Carrier analysis: average, median, variance of delays; percentage of delayed flights; cancellation rates.

- Route analysis: mean delays, busiest routes, percentage of delayed flights, cancelled routes.

- Airport analysis: busiest origins/destinations.

- Temporal analysis: flight counts by month, day of week, hour; delay averages by hour, weekday, and time-of-day buckets.

- Distance analysis: longest vs shortest flights, average distance by carrier and route.

- Correlation checks: departure vs arrival delay, numeric correlation matrix.

- Extreme cases: flights >5 hr delay, flights arriving >30 min early.

- Final summaries saved in an in-memory checkpoint (Step 50).

## Key Findings

- Time Parsing: HHMM formats contained invalid entries (e.g., 2400), requiring error-tolerant conversion.

- Delays: Strong correlation between departure and arrival delays, confirming compounding effect.

- Carrier performance: Large variation in both mean and variance of delays; some carriers consistently more punctual.

- Cancellations: Non-trivial proportion of cancelled flights, with varying cancellation codes.

- Routes: Certain high-volume routes also showed high delay percentages.

- Seasonality: Clear peaks in flight counts by month/day, supporting later seasonality analysis.

## Challenges Faced

- Schema variability: Columns like CANCELLED vs CANCELED needed defensive coding.

- Null handling: Distinguishing between “no delay” and “cancelled flight” delays was non-trivial. Flags avoided incorrect imputations.

- Memory efficiency: Even with 100k rows, dtype optimization showed clear performance gains — crucial for full datasets.

- Edge cases: Invalid HHMM entries, extreme delay values, and missing scheduled times required careful parsing.

## Learnings

- Building utility functions early (hhmm_to_time, downcast_nums, top_n_counts) saves repeated effort later.

- It's best practice to keep raw, cleaned, and sampled versions of the dataset.

- Rolling metrics (like route delay averages) provide valuable context beyond raw delays.

- Exploratory summaries (steps 21–50) revealed useful patterns that can directly inform visualization design.