Deliverable 2: ML Notebooks for Emission Prediction and Ranking; Documentation; Automated Route Emission Update Pipeline

This deliverable focuses on machine learning for predicting CO_2 emissions and ranking airlines by efficiency, along with an automated pipeline for updates. It utilizes the clean CSV from Deliverable 1. The ML model uses Ordinary Least Squares (OLS) regression for prediction, ranking is performed via groupby/aggregation, and the pipeline is implemented in Streamlit for manual or automated updates.

0.1 Loading and Preparing Data

1. In `train_emission_predictor()`, load `flight_emissions_clean.csv` using Pandas.
2. Drop rows missing `distance_km`, `passenger_capacity`, or `co2_per_passenger_kg`.
3. Define features and target:
    - Features (X): `distance_km`, `passenger_capacity` (add constant for intercept)
    - Target (y): `co2_per_passenger_kg`

0.2 Training ML Model for Prediction

1. Use `statsmodels` OLS:

```
model = sm.OLS(y, X).fit()
```

2. Display summary: coefficients, R-squared, p-values.
3. Provide interactive prediction: User inputs `distance_km` and `passenger_capacity`, then predict using `model.predict()`.
4. Visualization: Scatter plot of actual vs. predicted CO_2.

0.3 Ranking Airlines

1. In `compare_airlines_efficiency()`:

   - Group by `airline_name` and aggregate mean, standard deviation, and count for:
     - `co2_per_passenger_per_km`
     - `distance_km`
     - `co2_per_passenger_kg`
   - Sort by average CO_2/km to determine ranking.
   - Display the DataFrame and bar plot (using Plotly).
   - Benchmark against scraped airline sustainability reports.

0.4 Automated Update Pipeline

1. Function `automated_update_pipeline()`:

   - Initialize calculator
   - Extract and save flights (calls Deliverable 1 methods)
   - Compare efficiencies and train model

2. Streamlit integration:

   - Sidebar for `total_flights` input
   - Button triggers update with spinner for progress
   - Store DataFrame in `session_state`

3. Displays include:

   - Summary statistics (histograms for emissions and distance)
   - Top efficient airlines
   - Efficiency analysis
   - Model prediction results

0.5 Notebooks Structure

- **ML Prediction Notebook** (`emission_prediction.ipynb`):
  - Load data, train OLS, predict interactively, plot actual vs. predicted
- **Ranking Notebook** (`airline_ranking.ipynb`):
  - Load data, group and aggregate, rank/sort airlines, plot bar charts, compare to reports

0.6 Documentation

- **Model**: OLS regression for linear relationship:

$$\mathrm{CO_2} \sim \text{distance} + \text{capacity}$$

- **Ranking**: Based on mean CO_2/km; flight count considered for reliability.
- **Pipeline**: Runs on button click; can be scheduled (e.g., via cron if deployed).
- **Limitations**: Simple linear model; assumes data completeness; no advanced ML methods (e.g., Random Forests).