Dataset Creation Process

1. Initial Setup and Data Sources

The data preparation pipeline utilizes parquet files containing flight trajectory data alongside CSV files with flight metadata. The primary input arguments include:

- **Challenge Set** (--challenge): Loads challenge_set.csv.
- **Submission Set** (--submission): Loads submission_set.csv.
- **Final Submission Set** (--submissionv2): Loads final_submission_set.csv.

Each set provides basic flight metadata (e.g., flight_id, date), and the correct dataset is selected based on these command-line flags.

The main script construct_dataset_v4.py acts as the controller script that includes one single dataset creation run.

The two auxiliary scripts read_parquet_v3.py and total_g_v3.py perform the read in of the parquet files and the calculations necessary to obtaining valuable variables for the dataset respectively.

The calculations were performed using python 3.10 using Pandas and Pyarrow.

2. Folder and Output Configuration

The code ensures the presence of a directory (rip) for storing error logs and outputs. Each dataset generates unique output files for logging and storing extracted metrics. This may be interesting when evaluating data loss along the dataset creation process. Honestly though, we mostly used it to check if our code works and the data is reliable.

3. Parquet Data Extraction

For each flight record, the parquet files contain different information, of which we took:

- **Flight ID**: Unique identifier for the flight.
- **Timestamp, Latitude, Longitude**: Provide temporal and spatial flight trajectory data.
- **Altitude and Groundspeed**: Key metrics for analyzing flight dynamics.
- **Weather Components**: Wind components (u and v), temperature, and specific humidity at different points.

Using specific functions for various calculations, the data extraction involves filtering based on flight_id and date to retrieve only the relevant flight's data.

4. Outlier Removal

To ensure data quality, the remove_outliers function filters out anomalies:

- **Threshold-based Filtering**: Removes sudden changes in altitude (default threshold: 1000 meters).

- **Low Altitude Cutoff**: Identifies the lowest altitude point in the first half of the flight data and discards all data points before this point if deemed an outlier.

5. Feature Calculation

- **Lift-off Calculations (--lift_off)**:
  - Retrieves timestamp, groundspeed, and altitude from the parquet data.
  - **Metrics**:
    - **Time to Lift-off**: Time taken for the aircraft to reach lift-off conditions.
    - **Ground Speed at Lift-off**: Final groundspeed during lift-off.
    - **Ground Speed Delta**: Speed difference from the start to lift-off.
  - Results are appended to ground_speed_data_<set>.csv.

Calculation of this dataset:

The ground_speed_data function calculates metrics related to the aircraft's ground speed and altitude specifically during the take-off phase. This function is essential for understanding the initial momentum and climb characteristics of the aircraft.

a. **Elapsed Time in Seconds**
   - **Calculation**: Converts the timestamp column to datetime format and computes the elapsed time in seconds from the first timestamp.
   - **Purpose**: Provides a time reference to measure duration-related metrics, particularly time-to-lift-off.

b. **Data Filtering for Groundspeed > 0**
   - **Calculation**: The DataFrame is filtered to include only rows where groundspeed is greater than zero, as these indicate motion rather than stationary periods.
   - **Purpose**: Excludes idle or parked phases, isolating data points when the aircraft is actively moving, which is critical for take-off analysis.

c. **Lift-Off Start and End Points**
   - **Lift-Off Start**: Defined as the first point where groundspeed exceeds 20 knots.
   - **Lift-Off End**: Defined by altitude, where the altitude surpasses the initial value by 100 meters (~328 feet).
   - **Purpose**: These thresholds help identify the critical period from initial roll (groundspeed > 20 knots) until the aircraft has lifted sufficiently off the ground, capturing the main take-off period.

d. **Time to Lift-Off**
   - **Calculation**: Time difference in seconds between the lift-off start and end points.
   - **Purpose**: Represents the time taken for the aircraft to reach lift-off, essential for analyzing the efficiency and power of take-off.

e. **Ground Speed at Lift-Off**
   - **Calculation**: Groundspeed at the final timestamp within the lift-off window.
   - **Purpose**: Indicates the speed of the aircraft as it lifts off, offering insights into the momentum needed for lift-off.

f. **Ground Speed Delta**
   - **Calculation**: Difference between the first and last groundspeed values in the lift-off window.

- **Purpose**: Measures acceleration, showing the increase in speed from initial roll to lift-off, a key factor in take-off dynamics.

- Jet Stream Calculations (--jet_stream):
    - Retrieves latitude, longitude, and altitude.
    - Metrics:
        - Average Altitude: Mean altitude over sampled data points.
        - Jet Stream Coefficient: Combines polar and subtropical jet influences (longitude changes). This coefficient is computed by analyzing latitude-based jet stream activity within specific geographic bounds.
    - Results are stored in jet_stream_data_<set>.csv.

Calculation of this dataset:

The jet_stream_data function evaluates atmospheric influences, such as jet stream strength, along the flight path. These calculations help assess external forces affecting the aircraft's trajectory and performance.

a. Average Altitude
    - Calculation: Computes the mean of the altitude column.
    - Purpose: Establishes an altitude reference for jet stream impact assessment since altitude influences jet stream interaction.
b. Data Sampling for Latitude and Longitude Differences
    - Sampling: The DataFrame is sampled to include every 100th data point, reducing data density while preserving spatial coverage.
    - Latitude and Longitude Differences: Computes differences in latitude and longitude between consecutive points.
    - Purpose: The sampling provides manageable data for identifying directional trends (east-west shifts) influenced by jet streams.
c. Vector Construction and Jet Stream Activation
    - Vector Creation: Each sampled point's latitude and longitude differences form a vector.
    - Jet Stream Activation:
        - Polar Jet Stream: Activated for latitudes between 40° and 60° with a coefficient of 1.
        - Subtropical Jet Stream: Activated for latitudes between 20° and 30° with a coefficient of 0.75.
    - Purpose: Identifies areas within typical jet stream latitudes, applying different strength coefficients based on an estimation of the strength of the polar and subtropical jet streams in relation to each other.
d. Jet Stream Coefficient Calculation
    - Calculation: The longitudinal component of the vector is weighted by jet stream coefficients, then summed to form an overall jet stream influence coefficient.

- Weather Calculations (--weather):
    - Extracts weather data (temperature, wind components, and humidity) and altitude.
    - Metrics:
        - Departure and Arrival Temperature: Median temperatures at departure and arrival phases.
        - Wind Components (u, v): Summation of the u and v wind components over the flight.
        - Vertical Ascend and Descend Rates: Median rates for the highest and lowest 100 points in the vertical_rate column.
        - Humidity Difference: Difference between median humidity at high and low points.
    - Results are saved in weather_data_<set>.csv.

    - **Purpose**: Indicates the level of assistance or resistance provided by weather conditions, essential for understanding external influences on ground speed and fuel efficiency.

Calculation of this dataset:

The weather_data function gathers meteorological metrics across various phases of the flight, including temperature, wind, and humidity, which are essential for estimating environmental effects on flight performance.

a. Departure and Arrival Temperatures
    - Calculation:
        - Departure Temperature: Median of the first 60 data points in the temperature column.
        - Arrival Temperature: Median of the last 60 data points.
    - Purpose: These represent the ambient temperature at departure and arrival, which can impact fuel burn, engine performance, and TOW calculations due to varying air density.
b. Wind Components (u and v)
    - Calculation: Summation of the u_component_of_wind and v_component_of_wind columns.
    - Purpose: Provides an aggregate measure of wind forces experienced during the flight, with u and v components representing east-west and north-south wind forces, respectively. High winds can aid or hinder take-off and influence in-flight stability.
c. Vertical Ascend and Descend Rates
    - Calculation:
        - Vertical Ascend: Median of the highest 100 values in vertical_rate.
        - Vertical Descend: Median of the lowest 100 values.
    - Purpose: Reflects the aircraft's climb and descent characteristics under varying atmospheric conditions, revealing potential for smooth or turbulent ascent/descent.
d. Humidity Difference
    - Calculation:
        - Highest Humidity: Median of the top 100 values in specific_humidity.

        - **Lowest Humidity**: Median of the bottom 100 values.
        - **Difference**: Subtraction of the lowest from the highest humidity values.
    - **Purpose**: Indicates the humidity range experienced during flight, which affects air density and engine efficiency. Large humidity changes can impact climb rates, fuel burn, and overall flight efficiency.

6. Error Handling and Logging

If any issues arise during data extraction (e.g., missing files, outliers not handled), the ripperoni files log the flight_id and the error count, allowing for troubleshooting or later retries. In such cases this flight is excluded from the newly created dataset

7. Output

Each CSV output contains calculated metrics for each flight and dataset:

- **Ground Speed Data CSV**: flight_id, time_to_lift_off, ground_speed_at_lift_off, ground_speed_delta.
- **Jet Stream Data CSV**: flight_id, avg_altitude, jet_stream_coeff.
- **Weather Data CSV**: flight_id, temperature at departure and arrival, wind components, vertical ascent/descent, and humidity differences.

8. Final merge

In the final dataset creation process, the script create_dataset_v6.py combines multiple data sources to produce a comprehensive dataset, challenge_set_v6.csv.

a. **Initial Merging**:
    - The primary data, challenge_set.csv, is progressively merged with auxiliary datasets on flight_id, including ground_speed_data_challenge_set.csv, jet_stream_data_challenge_set.csv, and weather_data_challenge_set.csv.
    - After each merge, rows with missing values are dropped to ensure a complete dataset with reliable values for each feature.
b. **Economic and Market Data Integration**:
    - The dataset then incorporates oil_prices_2022.csv and msci_world_2022.csv by merging on date with a nearest match approach, filling any missing dates with the closest available value in the supplementary datasets.
    - The oil prices and MSCI World data were obtained using the `yfinance` library, which downloaded daily historical data for 2022. The scripts `oil.py` and `MSCI_World.py` specifically used the tickers `CL=F` for WTI Crude Oil and `URTH` for MSCI World, saving the data to `oil_prices_2022.csv` and `msci_world_2022.csv`, respectively, for integration into the final dataset.
c. **Final Export**:
    - The fully merged dataset is saved as challenge_set_v6.csv.