Data Cleaning and Merging Workflow

[Import Libraries]

1. **Load and Inspect Datasets**

- [x] ~~Load and inspect `electricity_data_germany.csv` into `electricity_df`~~
- [ ] Observations:
  - [ ] Missing values in `DE_load_forecast_entsoe_transparency`, `DE_solar_generation_actual`, and `DE_wind_generation_actual`
  - [x] ~~Can ignore `CET_timestamp`, use `UCT`~~
  - [ ] `DE_load_forecast_entsoe_transparency` is not a feature (remove)
  - [x] ~~Electricity data starts in 2014, check overlap with weather data~~
  - [x] ~~Identify rows with missing values in `electricity_df`~~
    - [ ] Observations:
      - [ ] Missing solar generation on 2015-02-28
      - [ ] Missing solar and wind data on specific dates in 2016
      - [ ] Missing load forecast on 2018-09-24
      - [ ] Analyze target variable behavior during missing data periods (!)
- [x] ~~Load and inspect `weather_data_germany.csv` into `weather_df`~~
  - [ ] Observation:
    - [ ] No missing values, but 0s in radiation columns.

2. **Check Timestamp Formats**

- [x] ~~Print the first and last values of the `utc_timestamp` column in both dataframes.~~
  - [ ] Observations:
    - [x] ~~Timestamps are in ISO 8601 format.~~
    - [x] ~~Weather data fully overlaps electricity data in terms of date ranges~~
- [x] ~~Convert `utc_timestamp` columns to datetime format using `pd.to_datetime()`~~
- [x] ~~Verify the data types of the converted columns and check for any missing timestamps (none)~~

3. **Missing Values Analysis**

- [x] ~~Calculate the count and percentage of zeros in `DE_radiation_direct_horizontal` and `DE_radiation_diffuse_horizontal` in `weather_df`.~~
- [ ] Analyze the distribution of zeros by hour.
  - [ ] Observations:
    - [ ] Zeros in radiation columns indicate nighttime.
    - [ ] Create `is_daylight` column based on hour.
- [x] ~~Create `is_daylight` column in `weather_df`~~
- [ ] Remember to apply normalization to radiation data only during daylight hours (`is_daylight` = 1).
- [ ] Weather data:

- [ ] Need to determine if missing values are zeros or actual gaps.
- [ ] Need to analyze target variable behavior during missing data.
- [ ] Decide and implement a strategy for handling missing values in `electricity_df`, considering the cyclical nature of solar generation.

4. **Outlier and Distribution Analysis**

- [x] ~~Display summary statistics for both dataframes using `df.describe()`~~
- [x] ~~Display 1st and 99th percentile values for both dataframes using `df.quantile([0.01, 0.99])`~~
  - [ ] Observations:
    - [x] ~~No outliers in solar data.~~
    - [ ] [ ] Potential outliers in wind features.
    - [x] ~~Weather data is skewed.~~
- [ ] Decide whether to clip outliers in wind features.
- [ ] Consider log transformation for skewed weather data.
- [ ] Visualize data for outlier detection (not implemented in notebook)
- [ ] Ask ChatGPT or other LLM to carry out its own EDA, in case things are missed here

5. **Merging**

- [x] ~~Set `utc_timestamp` as the index for both dataframes.~~
- [x] ~~Sort dataframes by index.~~
- [x] ~~Trim `weather_df` to match the time range of `electricity_df`.~~
- [x] ~~Reindex `weather_df` to match the 15-minute timestamps of `electricity_df`.~~
- [x] ~~Merge the two dataframes using a left join.~~
- [ ] 2 merged versions:
  - [ ] Interpolate missing values in specified columns (`DE_temperature`, `DE_radiation_direct_horizontal`, `DE_radiation_diffuse_horizontal`, `hour`).
  - [ ] Forward-fill remaining missing values. Backward-fill any leading NaNs.
- [x] ~~Display the first few rows of the merged dataframe using `merged_ffill.head()` - matches the needed data~~
- [ ] Decide which to use - interpolated or forward fill!