**Introduction**

**1. Day-ahead pricing in energy markets**

Day-ahead pricing is a prevalent mechanism within energy markets, such as those for electricity, natural gas, and various other commodities. This system involves establishing prices 24 hours before they take effect.

For example, on January 11, 2024, the market would announce the prices that are to be in effect for January 12, 2024. It is important to note the distinction in how these resources are priced: natural gas prices are set on a daily basis, whereas electricity prices are calculated for each hour of the day.

**2. Understanding the timestamps in datasets**

The datetime entry in a dataset, regardless of its actual label, consistently marks the beginning of a one-hour interval. In contrast, weather-related datasets capture certain measurements—such as temperature or cloudiness—at a specific point in time, which corresponds to the conclusion of that one-hour window.

To put this into perspective with examples:

In datasets concerning energy:

A timestamp of 14:00 indicates that the listed energy prices are applicable for the duration from 14:00 to 15:00.

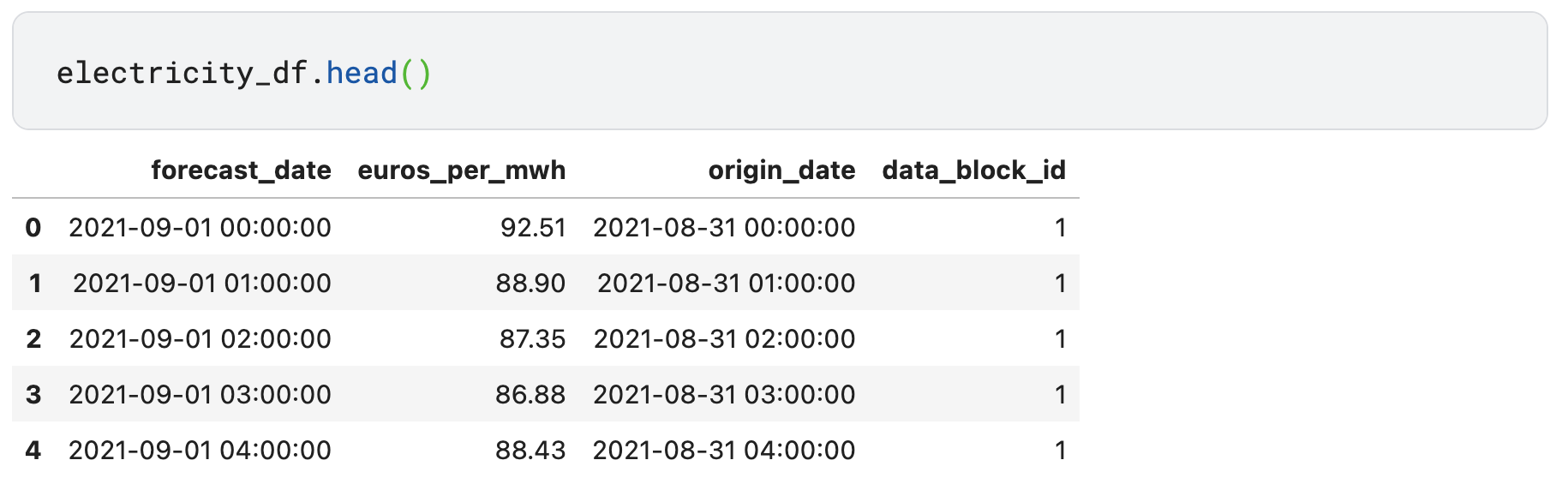
In datasets related to weather:

A timestamp of 15:00 signifies that the weather metrics, like temperature, are observed exactly at 15:00, effectively marking the close of the interval that started at 14:00 and ended at 15:00.

**3. Anatomy the datasets**

electricity\_prices.csv

* origin\_date - The date when forecasting occurs
* forecast\_date - Represents the start of the 1-hour period when the price is valid
* euros\_per\_mwh - The price of electricity on the day ahead markets in euros per megawatt hour.
* data\_block\_id - All rows sharing the same data\_block\_id will be available at the same forecast time.



################## electricity

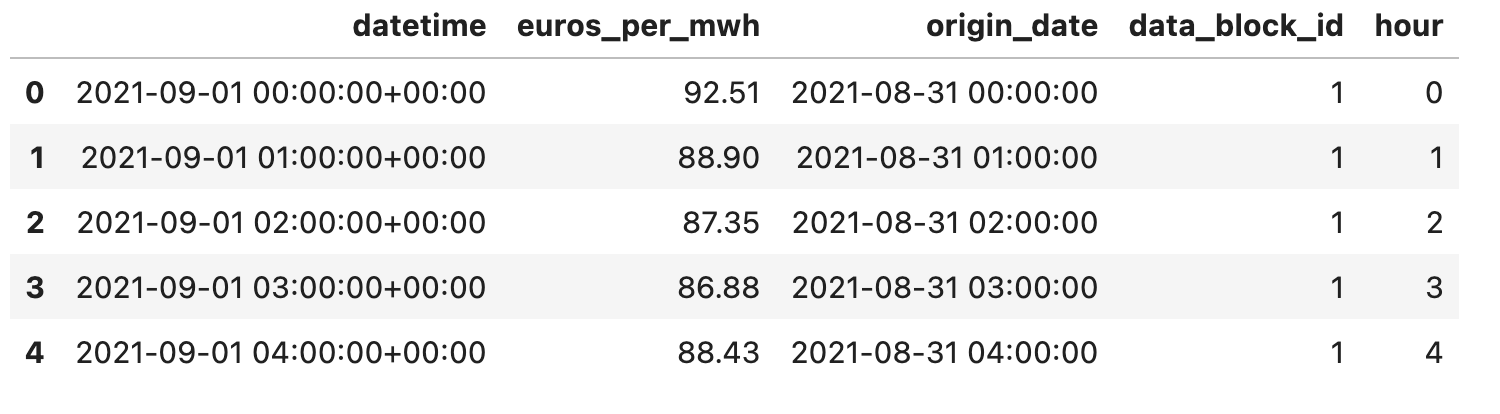
# rename 'forecast\_date' column to 'datetime' for consistency before merging

electricity\_df = electricity\_df.rename(columns={'forecast\_date': 'datetime'})

# convert datetime to UTC

electricity\_df['datetime'] = pd.to\_datetime(electricity\_df['datetime'], utc=True)

# add hour column

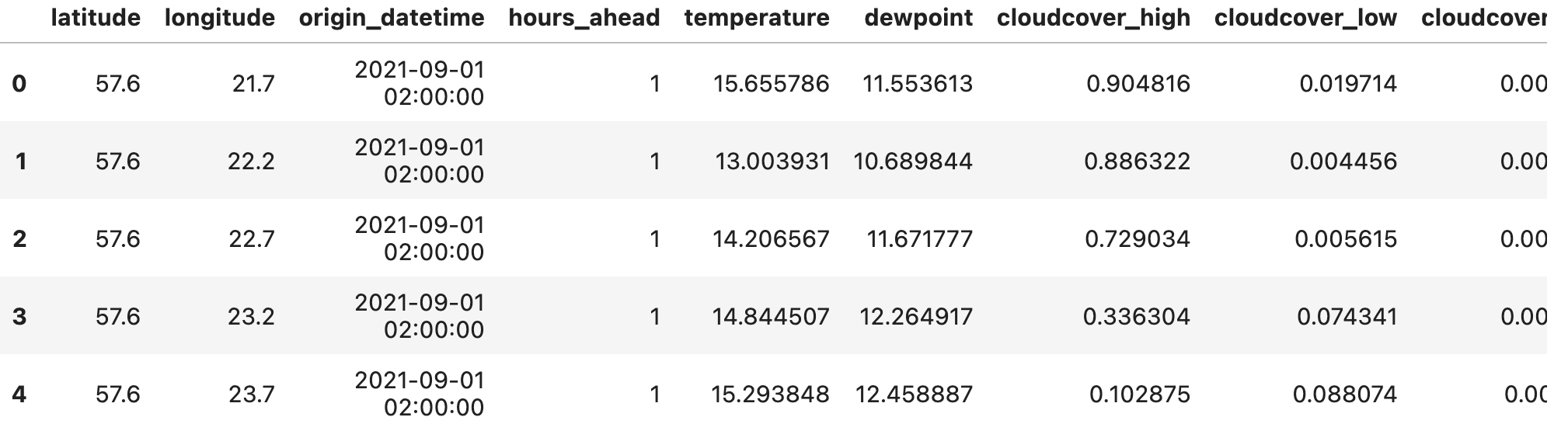
electricity\_df['hour'] = electricity\_df['datetime'].dt.hour

Note: add hour column, the price is assigned to forecast\_date, thus rename it into datetime for consistency.

**Forecast\_weather.csv**

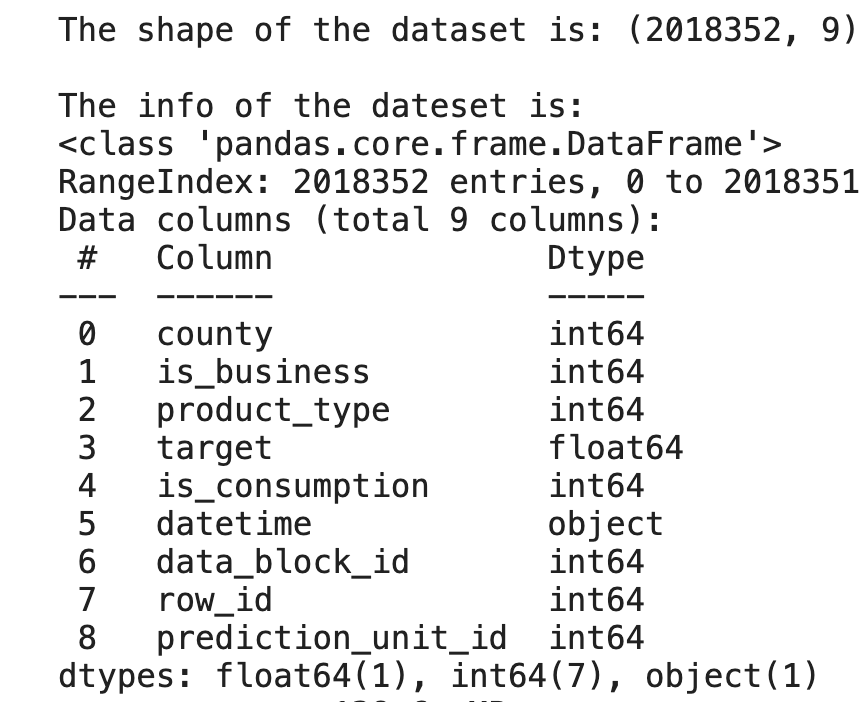
Weather forecasts that would have been available at prediction time.

* [latitude/longitude] - The coordinates of the weather forecast.
* origin\_datetime - The timestamp of when the forecast was generated.
* hours\_ahead - The number of hours between the forecast generation and the forecast weather. Each forecast covers 48 hours in total.
* temperature - The air temperature at 2 meters above ground in degrees Celsius. Estimated for the end of the 1-hour period.
* dewpoint - The dew point temperature at 2 meters above ground in degrees Celsius. Estimated for the end of the 1-hour period.
* cloudcover\_[low/mid/high/total] - The percentage of the sky covered by clouds in the following altitude bands: 0-2 km, 2-6, 6+, and total. Estimated for the end of the 1-hour period.
* 10\_metre\_[u/v]\_wind\_component - The [eastward/northward] component of wind speed measured 10 meters above surface in meters per second. Estimated for the end of the 1-hour period.
* data\_block\_id
* forecast\_datetime - The timestamp of the predicted weather. Generated from origin\_datetime plus hours\_ahead. This represents the start of the 1-hour period for which weather data are forecasted.
* direct\_solar\_radiation - The direct solar radiation reaching the surface on a plane perpendicular to the direction of the Sun accumulated during the hour, in watt-hours per square meter.
* surface\_solar\_radiation\_downwards - The solar radiation, both direct and diffuse, that reaches a horizontal plane at the surface of the Earth, accumulated during the hour, in watt-hours per square meter.
* snowfall - Snowfall over hour in units of meters of water equivalent.
* total\_precipitation - The accumulated liquid, comprising rain and snow that falls on Earth's surface over the described hour, in units of meters.



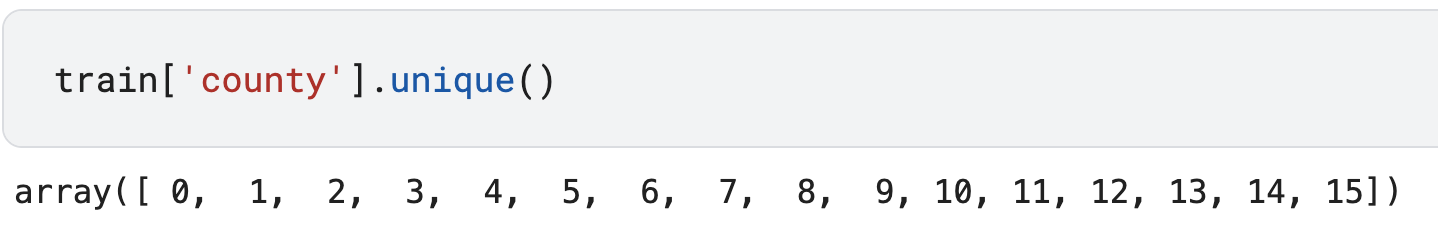
1. We have a lot of dates inside the dataset, what we are going to do is calculate the average hour value through out the whole dates.
2. # fw\_df\_datetime contains the average values for each hour, across all dates
3. # fw\_df\_datetime contains the average values for each county per hour, across all dates

Train.csv



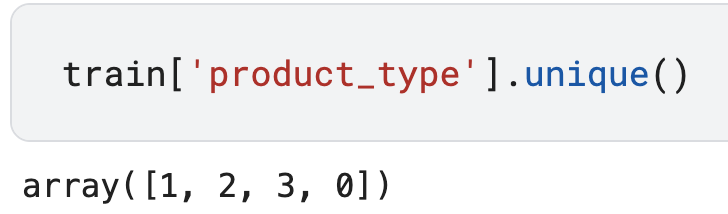
1. County - An ID code for the county.

There’re 15 16 counties inside this dataset.



1. Is\_business - boolean for whether or not the prosumer is a business.
2. product\_type - ID code with the following contract types:

{0: “combined”, 1: “fixed”, 2: “general service”, 3: “spot”}



1. Target - the consumption or production amount for the relevant segment of the hour.

The segments are defined by the county, is\_business, and product\_type.

1. Is\_consumption - boolean for whether or not this row’s target is consumption or production.
2. Datetime - the estonian time in EET (UTC+2) / EEST (UTC+3).

It describes the start of the 1-hour period on which target is given.

1. Data\_block\_id - all rows sharing the same data\_block\_id will be available at the same forecast time. (each number represents a day)

This is a function of what information is available when forecasts are actually made, at 11 AM

1. Row\_id - A unique identifier for the row.
2. prediction\_unit\_id - A unique identifier for the county, is\_business, and product\_type combination. New prediction units can appear or disappear in the test set.

### **Common Columns in Both Datasets:**

| **Common Fields** | **Forecast Weather Description** | **Historical Weather Description** |
| --- | --- | --- |
| latitude | Location's latitude | Location's latitude |
| longitude | Location's longitude | Location's longitude |
| temperature | Forecasted temperature | Observed temperature |
| dewpoint | Forecasted dewpoint | Observed dewpoint |
| cloudcover\_low | Forecasted low cloud cover | Observed low cloud cover |
| cloudcover\_mid | Forecasted mid cloud cover | Observed mid cloud cover |
| cloudcover\_high | Forecasted high cloud cover | Observed high cloud cover |
| cloudcover\_total | Forecasted total cloud cover | Observed total cloud cover |
| snowfall | Forecasted snowfall | Observed snowfall |
| direct\_solar\_radiation | Forecasted direct solar radiation | Observed direct solar radiation |
| data\_block\_id | Data block identifier | Data block identifier |

### **Unique Columns in Each Dataset:**

| **Unique Forecast Weather Fields** | **Description** |
| --- | --- |
| origin\_datetime | Timestamp indicating when the weather forecast was made |
| forecast\_datetime | Timestamp for the predicted weather conditions |
| hours\_ahead | Number of hours between the forecast generation and the forecasted conditions |
| 10\_metre\_u\_wind\_component | Eastward wind speed component at 10m above surface |
| 10\_metre\_v\_wind\_component | Northward wind speed component at 10m above surface |
| surface\_solar\_radiation\_downwards | Downwards surface solar radiation accumulated during the hour |
| total\_precipitation | Accumulated amount of liquid precipitation over the hour |

| **Unique Historical Weather Fields** | **Description** |
| --- | --- |
| datetime | Actual time when the weather conditions were observed |
| rain | Amount of rain measured over the hour |
| surface\_pressure | Air pressure measured at the surface |
| windspeed\_10m | Wind speed measured at 10 meters above ground level |
| winddirection\_10m | Wind direction measured at 10 meters above ground level |
| shortwave\_radiation | Measured global horizontal irradiation |
| diffuse\_radiation | Measured diffuse solar irradiation |