Overview
The goal of the competition is to create an energy prediction model of prosumers to reduce energy imbalance costs.

This competition aims to tackle the issue of energy imbalance, a situation where the energy expected to be used doesn't line up with the actual energy used or produced. Prosumers, who both consume and generate energy, contribute a large part of the energy imbalance. Despite being only a small part of all consumers, their unpredictable energy use causes logistical and financial problems for the energy companies.

Description

The number of prosumers is rapidly increasing, and solving the problems of energy imbalance and their rising costs is vital. If left unaddressed, this could lead to increased operational costs, potential grid instability, and inefficient use of energy resources. If this problem were effectively solved, it would significantly reduce the imbalance costs, improve the reliability of the grid, and make the integration of prosumers into the energy system more efficient and sustainable. Moreover, it could potentially incentivize more consumers to become prosumers, knowing that their energy behavior can be adequately managed, thus promoting renewable energy production and use.

Enefit is one of the biggest energy companies in Baltic region. As experts in the field of energy, we help customers plan their green journey in a personal and flexible manner as well as implement it by using environmentally friendly energy solutions.

At present, Enefit is attempting to solve the imbalance problem by developing internal predictive models and relying on third-party forecasts. However, these methods have proven to be insufficient due to their low accuracy in forecasting the energy behavior of prosumers. The shortcomings of these current methods lie in their inability to accurately account for the wide range of variables that influence prosumer behavior, leading to high imbalance costs. By opening up the challenge to the world's best data scientists through the Kaggle platform, Enefit aims to leverage a broader pool of expertise and novel approaches to improve the accuracy of these predictions and consequently reduce the imbalance and associated costs.

Dataset Description

Your challenge in this competition is to predict the amount of electricity produced and consumed by Estonian energy customers who have installed solar panels. You'll have access to weather data, the relevant energy prices, and records of the installed photovoltaic capacity.

This is a forecasting competition using the time series API. The private leaderboard will be determined using real data gathered after the submission period closes.

💡 Nota bene:

All datasets follow the same time convention. Time is given in EET/EEST. Most of the variables are a sum or an average over a period of 1 hour. The datetime column (whatever its name) always gives the start of the 1-hour period. However, for the weather datasets, some variables such as temperature or cloud cover, are given for a specific time, which is always the end of the 1-hour period.

Files

train.csv

county - An ID code for the county.

is_business - Boolean for whether or not the prosumer is a business.

product_type - ID code with the following mapping of codes to contract types: {0: "Combined", 1: "Fixed", 2: "General service", 3: "Spot"}.

target - The consumption or production amount for the relevant segment for the hour. The segments are defined by the county, is_business, and product_type.

is_consumption - Boolean for whether or not this row's target is consumption or production.

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.

data_block_id - All rows sharing the same data_block_id will be available at the same forecast time. This is a function of what information is available when forecasts are actually made, at 11 AM each morning. For example, if the forecast weather data_block_id for predictins made on October 31st is 100 then the historic weather data_block_id for October 31st will be 101 as the historic weather data is only actually available the next day.

row_id - A unique identifier for the row.

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.

gas_prices.csv

origin_date - The date when the day-ahead prices became available.

forecast_date - The date when the forecast prices should be relevant.

[lowest/highest]_price_per_mwh - The lowest/highest price of natural gas that on the day ahead market that trading day, in Euros per megawatt hour equivalent.

data_block_id

client.csv

product_type

county - An ID code for the county. See county_id_to_name_map.json for the mapping of ID codes to county names.

eic_count - The aggregated number of consumption points (EICs - European Identifier Code).

installed_capacity - Installed photovoltaic solar panel capacity in kilowatts.

is_business - Boolean for whether or not the prosumer is a business.

date

data_block_id

electricity_prices.csv

origin_date

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

forecast_weather.csv Weather forecasts that would have been available at prediction time. Sourced from the European Centre for Medium-Range Weather Forecasts.

[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.

historical_weather.csv Historic weather data.

datetime - This represents the start of the 1-hour period for which weather data are measured.

temperature - Measured at the end of the 1-hour period.

dewpoint - Measured at the end of the 1-hour period.

rain - Different from the forecast conventions. The rain from large scale weather systems of the hour in millimeters.

snowfall - Different from the forecast conventions. Snowfall over the hour in centimeters.

surface_pressure - The air pressure at surface in hectopascals.

cloudcover_[low/mid/high/total] - Different from the forecast conventions. Cloud cover at 0-3 km, 3-8, 8+, and total.

windspeed_10m - Different from the forecast conventions. The wind speed at 10 meters above ground in meters per second.

winddirection_10m - Different from the forecast conventions. The wind direction at 10 meters above ground in degrees.

shortwave_radiation - Different from the forecast conventions. The global horizontal irradiation in watt-hours per square meter.

direct_solar_radiation

diffuse_radiation - Different from the forecast conventions. The diffuse solar irradiation in watt-hours per square meter.

[latitude/longitude] - The coordinates of the weather station.

data_block_id

public_timeseries_testing_util.py An optional file intended to make it easier to run custom offline API tests. See the script's docstring for details. You will need to edit this file before using it.

example_test_files/ Data intended to illustrate how the API functions. Includes the same files and columns delivered by the API. The first three data_block_ids are repeats of the last three data_block_ids in the train set.

example_test_files/sample_submission.csv A valid sample submission, delivered by the API. See this notebook for a very simple example of how to use the sample submission.

example_test_files/revealed_targets.csv The actual target values from the day before the forecast time. This amounts to two days of lag relative to the prediction times in the test.csv.

enefit/ Files that enable the API. Expect the API to deliver all rows in under 15 minutes and to reserve less than 0.5 GB of memory. The copy of the API that you can download serves the data from example_test_files/. You must make predictions for those dates in order to advance the API but those predictions are not scored. Expect to see roughly three months of data delivered initially and up to ten months of data by the end of the forecasting period.