➡️ Context:

With increasing digitalisation and the ever-growing reliance on data servers, the significance of sustainable computing is on the rise. Schneider Electric, a pioneer in digital transformation and energy management, brings you this innovative challenge to play your part in reducing the carbon footprint of the computing industry.

The task is simple, yet the implications are profound. We aim to predict which European country will have the highest surplus of green energy in the next hour. This information will be critical in making important decisions, such as optimizing computing tasks to use green energy effectively and, consequently, reducing CO2 emissions.

🎯 Objective:

Your goal is to create a model capable of predicting the country (from a list of nine) that will have the most surplus of green energy in the next hour. For this task, you need to consider both the energy generation from renewable sources (wind, solar, geothermic, etc.), and the load (energy consumption). The surplus of green energy is considered to be the difference between the generated green energy and the consumed energy.

The countries to focus on are: Spain, UK, Germany, Denmark, Sweden, Hungary, Italy, Poland, and the Netherlands.

The solution must not only align with Schneider Electric's ethos but also go beyond its current offerings, presenting an unprecedented approach.

📉 Dataset:

You will be working with time-series data of hourly granularity extracted from the ENTSO-E Transparency portal using its API: https://transparency.entsoe.eu/content/static_content/Static%20content/web%20api/Guide.html. This data includes:

Electricity consumption (load)

Wind energy generation

Solar energy generation

Other green energy generation

These features are provided for each of the mentioned countries and are aggregated at different intervals (15 min, 30 min, or 1h), depending on the country. All the data should be homogenized to 1-hour intervals for consistency.

You are responsible to use the API to get the data. In order to create your 'train.csv' and test.csv datasets you will need to download the data from 01-01-2022 to 01-01-2023, group it as indicated below and make an 80/20 split, being the 80% the one used for training and the 20% used for testing.

You will need to provide a security token to make the ENTSO-E API calls. You can use the following one:

1d9cd4bd-f8aa-476c-8cc1-3442dc91506d

If the first token reaches its API calls rate limit, you can use the next token:

fb81432a-3853-4c30-a105-117c86a433ca

b5b8c21b-a637-4e17-a8fe-0d39a16aa849

2334f370-0c85-405e-bb90-c022445bd273

⚠ Note: It is possible that a token reaches its API calls rate limit. In this case, try with a different token. If the problem persists, wait a minute and try again.

Repo Structure:

You are provided with a skeleton repository that is mandatory to use for this challenge: https://github.com/nuwe-io/SE-Europe-Data_Challenge_Template. Any deviation from the provided structure might lead to disqualification. Nonetheless, you can include additional files if you wish to. The repository includes:

```
|__README.md

|__requirements.txt

|

|__data

|  |__your_train.csv

|  |__test.csv
```

```
|
|__src
|  |__data_ingestion.py
|  |__data_processing.py
|  |__model_training.py (or model_training.ipynb)
|  |__model_prediction.py
|  |__utils.py
|
|__models
|  |__model.pkl
|
|__scripts
|  |__run_pipeline.sh
|
|__predictions
   |__example_predictions.json
   |__predictions.json
```

The data folder contains a test.csv file that should be used as reference (only contains the header) on how to construct your datasets. You should also include your train.csv file which, together with the test set just mentioned, will be used for training your model and for evaluating its performance, respectively. The src folder contains already started python files that will help you guide through the challenge. For example, the file data_ingestion.py contains the code necessary to extract data from the API. Make sure you understand each file properly so you don't miss anything. You are required to complete the files in the src folder BUT if you prefer to implement it in another language (such as R) it is completely valid. In the scripts folder you'll find the run_pipeline.sh which is a bash script to automate the process from data ingestion to model prediction. This scripts calls, step by step, each python file in the src folder so you can exclusively focus on writing the necessary code. If you used another language we strongly recommend to update this script accordingly. Part of the challenge is demonstrating your capability to perform data ingestion and ETL, hence you should generate these datasets by completing the corresponding scripts under src. To clarify, the file called run_pipeline.sh runs the full pipeline from data ingestion to model prediction -- all except model training. In this file you are tasked to monitor the performance of the pipeline at each stage. Some questions you might want to answer include:

How many data points have we ingested?

Do we loose any data during data processing?

Which data have we lost?

Why did we loose it? However, it is up to you to define the specific measures you are monitoring.

📊 Data Processing:

Missing values in the dataset should be imputed as the mean between the preceding and following values. Data with resolution finer than 1 hour must be resampled to an hourly level. For example, data at a 15-minute resolution should be aggregated into 1-hour intervals by summing every 4 consecutive rows.

You will need to identify what energy types each column represent, and discard the ones that are not green energy sources. You can refer to the ENTSO-E Transparency portal API documentation to understand how the energy source types are represented.

In the end of the data processing workflow, you should end up with a single CSV file which includes columns per country representing the following values: generated green energy per energy type (one column per wind, solar, etc), and load. Make sure that all those values are in the same units (MAW).

As mentioned before, you can check the exact columns that will need to appear in your dataset by looking at the test.csv file provided inside the data folder.

You will also need to add an additional column that will be your label: the ID of the country with the bigger surplus of green energy for the next hour.

The country IDs used to evaluate your model will be the following:

The country IDs used to evaluate your model will be the following:

{

SP: 0, # Spain

UK: 1, # United Kingdom

DE: 2, # Germany

DK: 3, # Denmark

HU: 5, # Hungary

SE: 4, # Sweden

IT: 6, # Italy

PO: 7, # Poland

NL: 8 # Netherlands

}

Again, the surplus of green energy is considered to be the difference between the summation of all generated green energy and the consumed energy (load). The resulting CSV will be the one you use to train and validate your model. Please note that the label column is not included in the provided test.csv set.

⚠️ Please note that the label column is not included in the provided test.csv set.

✴️ Model:

You are free to choose the type of model for this task, whether it be a traditional time series model like ARIMA, or a more complex deep learning model. Your model should predict which of the 9 countries will have the most surplus green energy in the next hour.

⚠️ Your model predictions with the test data should be stored in the same format at the example_predictions.json file provided where for each entry (data point of your time series) you have a country ID predicted for the next hour. The final file should be called predictions.json. This file will be the one used to evaluate your model performance on F1-score macro.

⚠️ To obtain the final predictions.json you need to group all the datasets downloaded by StartTime and by hour (if one day has data in a 4-hour interval, this day only has 4 rows. If a day has no information, no rows must be generated). The predictions.json that you must submit has to be tested with the 20% of an 80/20 split of the 2022 data, as it was indicated before, being the 20% the last values of the year. The total amount of values that you should have is 442, as you can see in the example_predictions.json

✅ Submission:

You must submit the link to your GitHub challenge repository (must be public). This must at least have the writeup with the explanation of the process.

The submission will be available only for the team lead, the other will be related to the team and once the scoring is completed everyone will get the same score.

✍️ Evaluation

The challenge will be evaluated based on the following criteria:

Code quality (400/1200 points): Your code should be well-structured, efficient, and well-documented. The README.md file is also evaluated.

Data ingestion, ETL, and data monitoring (400/1200 points): Your ability to accurately ingest data from different sources and perform the necessary transformations, as well as the depth of insights from your data pipeline monitoring will be evaluated.

Model performance (400/1200 points): The accuracy of your model will be evaluated using the F1 macro score on a test set.

You are required to submit all the content generated during the challenge, including the trained models and the results of the predictions. Best of luck and happy coding!

🌍 Online Virtual World

We are happy to tell you that the Schneider Electric European Hackathon will be held in the Gather virtual world which will allow you to celebrate this European Hacakthon. Here is the link to it: Schneider Electric Gather Online World .

Before you enter, we recommend you to read the next instructions: Schneider Electric Gather Online World Guide .

SCHNEIDER ELECTRIC EUROPEAN HACKATHON 2023

New Deadline: Tuesday 21st November at 10.00 CET

Hello "Data-Science" participants,

Due to some questions aroused between yesterday and today, we have stablished a **new deadline in order to have some more time to work on your solutions and give some clues below for this Data Science challenge.**

You have time till **Tuesday November 21st at 10.00 CET** to upload your solution. Remember, the team leader is the one, who has to deliver the solution in NUWE's platform.

Below, **clues and considerations to take into account:**

☞ The green energies, you can use the ones mentioned above: ["B01", "B09", "B10", "B11", "B12", "B13", "B15", "B16", "B18", "B19"].

☞ The key is in the resampling of 1 hour intervals.

☞ When is it considered that there is no data? The key is that you have to gather all countries and adjust the intervals regularly to 1H, when there are countries that have no data at a certain time or day, that is simply 0 (or the NaN should be set to 0 as the challenge explains, it is part of handling the missing data).

☞ How to interpolate the data: "Missing values in the dataset should be imputed as the mean between the preceding and following values".

Example:

```
#Interpolate any missing data
df.interpolate(method='linear', limit_direction='both', inplace=True)
```

☞ In any case, so that this does not continue to generate confusion and as a sign of good faith, we have decided **to accept any solution, whether or not the predictions of 442 values are reached, and to correct each solution one by one**. We understand that the complexity may have been high. That is to say, even if you try the F1-score and it gives you a low score, it will not be so, as it will be corrected one by one.

Sorry for the inconvenience and I hope this message will help you to finish and deliver!