⚡ Power Grid Demand Forecasting Hackathon

🎯 Task

Predict **electricity demand (MW)** using 10+ power generation, import/export, and renewable energy variables.

- This is a **time-series regression** problem.
- Target variable: **`demand_mw`** (continuous)

🗂️ Dataset Files Provided

You are given the **FULL dataset**:

1. `data.csv` — contains **all timestamps (2015–2025)**
2. `sample_submission.csv` — expected output format
3. (Optional) `starter_notebook.ipynb`

📊 Dataset Description (data.csv)

The dataset contains **~92,000 rows**, one row per hour from **2015 → 2025**.

Below is the **column-by-column explanation**.

📅 Time Column

1. `datetime`

- Hourly timestamp (YYYY-MM-DD HH:MM:SS)

🏭 Grid Generation & Load Variables

These represent the total electricity ecosystem.

2. `generation_mw`

Total electricity generation at that hour (MW).

3. `demand_mw` (Target Variable)

Actual electricity demand in MW for that hour.

4. `load_shedding`

When demand > supply, load is cut (MW).
High load shedding usually indicates unmet demand.

🛢️ Fuel-Based Generation (Conventional Power)

5. `gas`

Gas turbine–based generation (MW)

6. `liquid_fuel`

Diesel/HFO-based thermal generation (MW)

7. `coal`

Coal-fired power plant generation

8. `hydro`

Hydropower generation (MW)

These features affect demand behavior based on availability.

☀️ Renewable Energy Generation

9. `solar`

Solar generation (high missing values during nights & cloudy periods)

10. `wind`

Wind generation (sparse; sometimes 0 when no wind farms operate)

Both introduce daily, seasonal, and weather-driven variability.

🌍 Cross-Border / Regional Power Exchange

11. `india_bheramara_hvdc`

Power exchange with India through HVDC Bheramara link.

12. `india_tripura`

Power import/export via Tripura region.

13. `india_adani`

Adani-based power import (very sparse early years)

14. `nepal`

Power exchange with Nepal.

These capture dependency on regional supply/demand fluctuations.

🟩 Target Variable

`demand_mw`

Your goal is to predict this **continuous** value for the most recent timestamps after your time-based split.

📤 sample_submission.csv

Format:

```
datetime,demand_mw
2025-01-01 00:00:00,1123.3
2025-01-01 01:00:00,1104.8
...
```

- `datetime` must match the timestamps from your test set.
- `demand_mw` is your predicted value.

✂ Split Train/Test by yourself

Since you received the **full dataset**, you must split by yourself.

🏆 Evaluation Metric

`RMSE (Root Mean Squared Error)`

Your model must minimize RMSE on the held-out test split.

□ Important Notes (READ CAREFULLY)

✔ No data leakage