



## Scope

- Prototype an LLM-based forecaster for short-term utility load data. Keep readable and runnable.

## Data (provided)

- 3 input .csv files, with each corresponding to a different utility company.
- Each input .csv file contains 3 months worth of hourly measured load data.
- CSV format: utility\_name, timestamp, load.

## Task

- For each utility, generate an hourly point forecast for the next 24 hours.
- The forecasting model must use an LLM as the primary modeling component (not merely for preprocessing or formatting).
- Include a basic out-of-sample check (holdout or simple rolling) and report MAE, MAPE and RMSE.
- Avoid data leakage. Briefly note any assumptions in code comments.

## Constraints & Allowed Tools

- Use only tools requiring API keys for LLMs or open source LLM models. Do not rely on any other external APIs/services that require keys.
- Local/open-source Python libraries are acceptable (e.g., pandas, numpy, matplotlib, statsmodels, etc.) but not for making the actual forecasts.
- Keep things reproducible and straightforward to run.

## What to Submit

- A functional Dockerfile which includes the source code. The container must build on arm64.
- Building and running the container should execute the full pipeline end-to-end on the provided dataset and produce forecasts.
- Code should be clearly commented and include README.md which includes your methodology and how you arrived at your final solution.

## Additional Guidance

- No manual steps during execution; runs should be scriptable end-to-end.
- Use environment variables for any LLM API keys; do not mount secrets.
- We prioritize methodology and code quality over raw accuracy.