# 2258-IE-6318-001-DATA MINING & ANALYTICS

## INFORMS Data Challenge – Progress Report

**Team Members :**

**Anuva Negi - 1002238067**

**Shreyas Rajapur Sanjay - 1002221283**

**Kumar Mantha - 1002233682**

## 1. Data Exploration

As part of our data exploration, we developed two key notebooks. The *initial_Exploration.ipynb* focused on loading and inspecting raw NetCDF (.nc) climate datasets using xarray, examining variables, coordinates, and dimensions before converting them into a structured pandas DataFrame for easier analysis. This allowed us to generate preliminary descriptive statistics and gain an initial understanding of the data structure. The *corr.ipynb* notebook extended this work by performing a correlation analysis across features using pandas, seaborn, and matplotlib, producing correlation matrices and heatmaps that highlighted dependencies and relationships among variables. Together, these efforts provided a strong foundation for feature engineering and model development in subsequent stages.

## 2. Data Preparation & Preprocessing

We developed Python files to handle the flattening of raw NetCDF (.nc) data into a structured train.csv, ensuring that weather variables, target outputs, and temporal information were organized into a machine learning–ready format. In addition, we implemented a comprehensive feature normalization pipeline that applies tailored scaling methods—such as z-score standardization, log-transformed scaling, robust scaling, and min–max normalization—across different groups of variables. These steps established a clean, consistent dataset and laid the groundwork for reliable model training and evaluation.

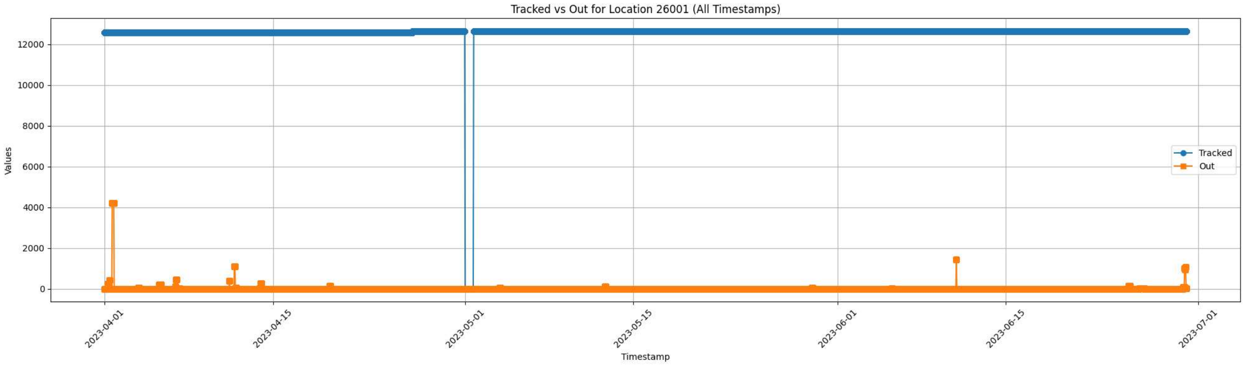
## 3. Baseline & Initial Models

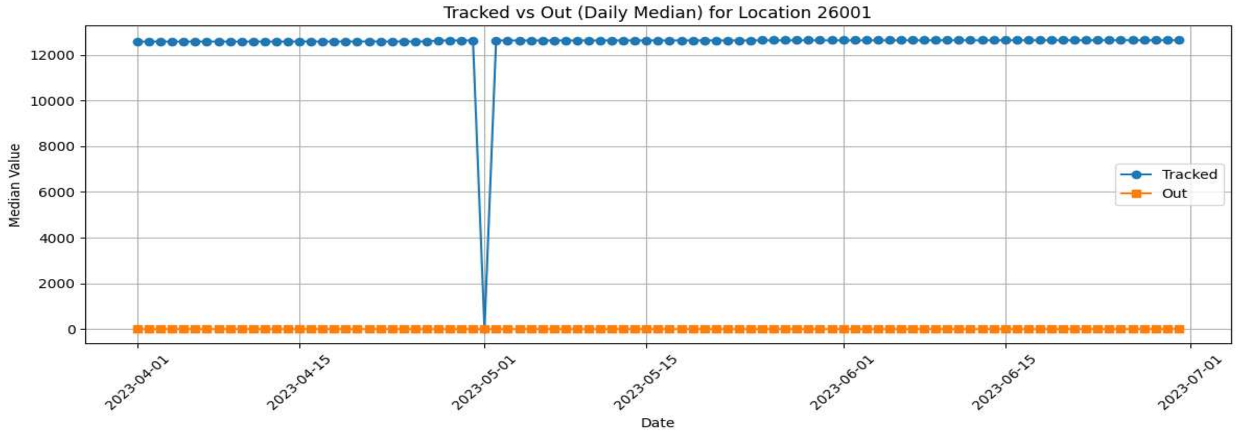
Apart from the demo files of SARIMA and Seq2Seq there were two initial approaches we explored. The first model used tree-based methods (LightGBM and XGBoost) with extensive feature engineering, including lagged variables and rolling statistics over different time windows, to predict the next 24 hours of weather values. These models were trained on historical data split into training and validation sets, with evaluation based on RMSE and predictions saved for analysis. The second approach leveraged deep learning using the N-BEATS architecture from PyTorch Forecasting, designed for time-series forecasting. This model used a sliding-window setup with 72 hours of history to predict the next 24 hours, trained with RMSE loss over multiple epochs. Predictions were then mapped back to timestamps for interpretability. Together, these models provided strong baselines combining classical ML with feature engineering and modern neural forecasting techniques.

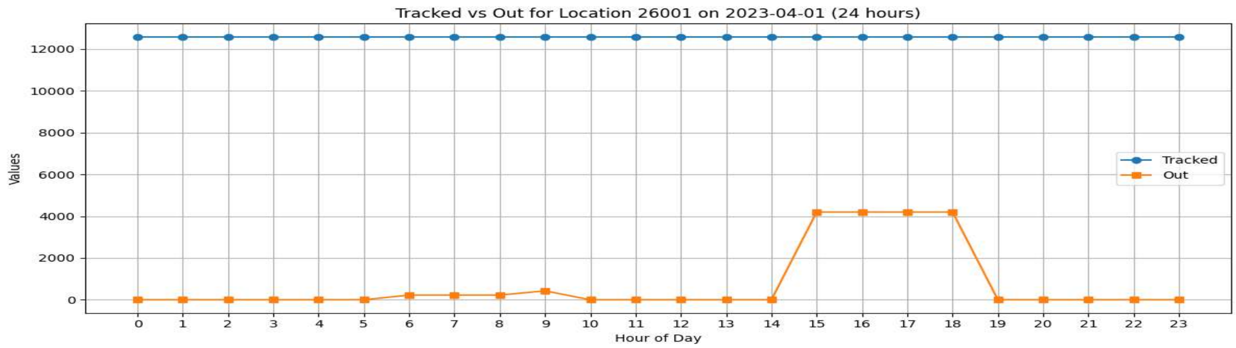
## 4. Advanced Model Demo

We also experimented with a script that implements a forecasting workflow leveraging PyTorch Forecasting's Temporal Fusion Transformer (TFT). It ingests NetCDF datasets, converts them into structured dataframes, and prepares time series inputs for model training. The model is trained on historical weather and output variables to generate forecasts for 24-hour and 48-hour horizons. Predicted results are exported as CSV files, ensuring reproducibility and streamlined evaluation.

## 5. **Data Visualizations for Tracked vs. Out to understand the pattern and dependency of Outage on Tracked data variable**







The visualizations collectively demonstrate that there is no direct correlation between the tracked and out variables. The tracked value represents a stable, high baseline of monitored customers for a given location, showing only minor fluctuations over time. In contrast, the out variable, which represents power outages, is almost always zero and appears as rare, sharp spikes, indicating that outages are sporadic, event-driven occurrences.

The primary relationship is a boundary condition: the number of outages (out) can never exceed the number of customers being monitored (tracked). The plots confirm that a high tracked value does not imply a high out value. Instead, outages are triggered by external factors, not by changes in the tracked customer count. On days with no outage events, both variables remain flat, further highlighting their independence.

## Key Takeaways

- Successfully established a pipeline from raw .nc data to ML-ready datasets.
- Gained insights into the dataset structure through EDA.
- Validated feasibility of both gradient boosting and deep learning approaches for predictive modeling.

## Next Steps (Planned Work)

- Expand model experimentation with additional architectures.
- Perform hyperparameter tuning to optimize model performance.
- Conduct more advanced feature engineering (temporal lags, interaction features, domain-specific transformations).
- Implement systematic model evaluation and benchmarking.
- Explore ensemble approaches to combine strengths of different models.

## Impact So Far

The first week’s work has laid a solid foundation by preparing the data, building reproducible pipelines, and validating initial models. This ensures scalability and sets the stage for deeper experimentation in the coming weeks.