AI VIET NAM
@aivietnam.edu.vn

Case Study:
IISE PG&E Energy Analytics Challenge 2025
- Load Forecasting using XGBoost

PhD. Cand. Huynh Q. Nguyen Vo
PhD. Student H M Mohaimanul Islam
MSc. Richard Reed
MSc. Yash Patel
STA Nguyen Phuc Thinh

Year 2025 Repo Code & Data

Outline

SECTION 1

Competition Overview

SECTION 2

Data Overview

SECTION 3

Modeling Strategy

SECTION 4

Evaluation

SECTION 5

Future Works

Competition Overview

Pacific Gas and Electric Company (PG&E) is one of the largest combined natural gas and electric utilities in the United States, serving millions of customers in (primarily) **northern and southern California**.

- To advance innovation in energy forecasting, the **Institute of Industrial and Systems Engineers (IISE)** organizes the **PG&E Energy Analytics Challenge.**
- Participants are tasked with **predicting hourly electricity load for an entire calendar year in a California region** strongly influenced by solar generation.
- Competitors are provided with **two years of historical data** and must generate **day-ahead forecasts using only the supplied features**, mimicking real-world operational constraints.

INSTITUTE OF INDUSTRIAL & SYSTEMS ENGINEERS
Solving complex & critical problems of the world.

Outline

SECTION 1

Competition Overview

SECTION 2

Data Overview

SECTION 3

Modeling Strategy

SECTION 4

Evaluation

SECTION 5

Future Works

Data Overview

- **Goal:** Forecast hourly electricity load for a full calendar year (day-ahead).
- **Provided:** Two (2) years of training data - load, temperature, Global Horizontal Irradiance (GHI) - from 5 sites.
- **Constraints:**
  - Use only provided exogenous variables;
  - No external data.
- **Operation Rule:**
  - For day d, we may use exogenous variables only up to and including day d.

Endogenous variable (response) is a variable within a model/system whose value is determined by other variables within that same model/system.

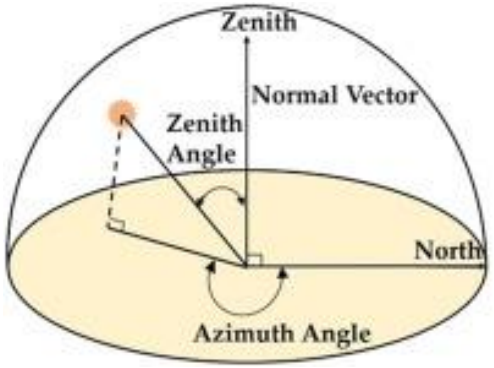
- In our case, **hourly electrical load**.

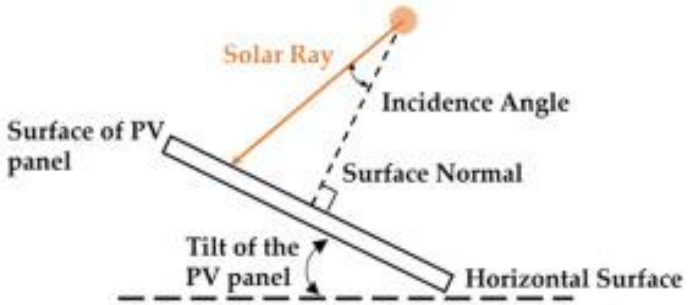
Exogenous variables (factor) are [independent] variables determined by forces or factors entirely outside of a given model/system.

- In our case, **temperature(s)** and **GHI(s).**

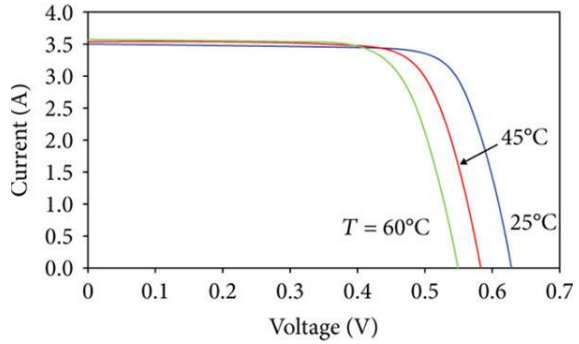
Data Overview

❖Basic Concepts for Domain Knowledge

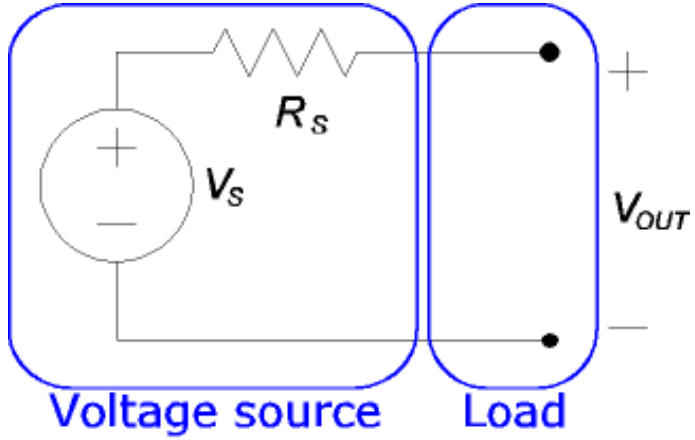




Global horizontal irradiance (GHI) is defined as the total solar radiation received on a horizontal surface, comprising both direct and diffuse horizontal radiation.



$$GHI \propto E_{\text{solar}} - \alpha\Delta_T$$



Electric load is defined as power consumed in the circuitry

$$E_{\text{Load}} \propto \text{Demand}$$

Data Overview (cont.)

❖How the raw data is organized

Timestamp **Endogenous variables** **Exogenous variables**

Year	Month	Day	Hour	Load	Site-1 Temp	Site-2 Temp	Site-3 Temp	Site-4 Temp	Site-5 Temp	Site-1 GHI	Site-2 GHI	Site-3 GHI	Site-4 GHI	Site-5 GHI
1	1	1	1	1,997	8	8.2	5.3	9.4	8.1	0	0	0	0	0
1	1	1	2	1,921	8.3	8.6	5.2	8.6	7.1	0	0	0	0	0
1	1	1	3	1,861	8.1	8.8	5.1	8.7	6.2	0	0	0	0	0
1	1	1	4	1,833	7.6	8.1	4.3	8.5	6	0	0	0	0	0
1	1	1	5	1,847	7.3	7.5	4	8.6	6.9	0	0	0	0	0
1	1	1	6	1,910	6.6	7.3	4	7.8	7.3	0	0	0	0	0
1	1	1	7	1,972	6.9	7.6	5.3	7.7	7.1	0	0	0	0	0
1	1	1	8	1,919	6.8	7.1	5.8	7.3	7.3	0	0	0	0	0
1	1	1	9	1,786	9.3	9.1	8.2	9.3	9.4	73	70	65	77	79
1	1	1	10	1,689	13	13.9	13.7	13.1	12.7	245	238	245	251	252
1	1	1	11	1,603	15.6	16.4	16.5	16.2	15.8	401	393	401	407	409
1	1	1	12	1,552	16.6	16.6	17.7	17.3	17	513	508	512	519	520
1	1	1	13	1,561	16.8	16.3	17.9	17.4	17.2	568	560	567	573	574
1	1	1	14	1,575	16.3	16.2	17.6	17	17	559	555	559	565	565
1	1	1	15	1,745	15.8	15.8	16.8	16.2	15.8	487	477	473	494	494
1	1	1	16	1,987	15.6	15.4	16.1	15.8	15.2	333	308	339	342	343
1	1	1	17	2,277	15.3	14.8	15.1	15.1	14.7	168	170	178	174	174
1	1	1	18	2,515	14.4	13.5	12.9	13.8	12.9	29	39	37	29	30

Data Overview (cont.)

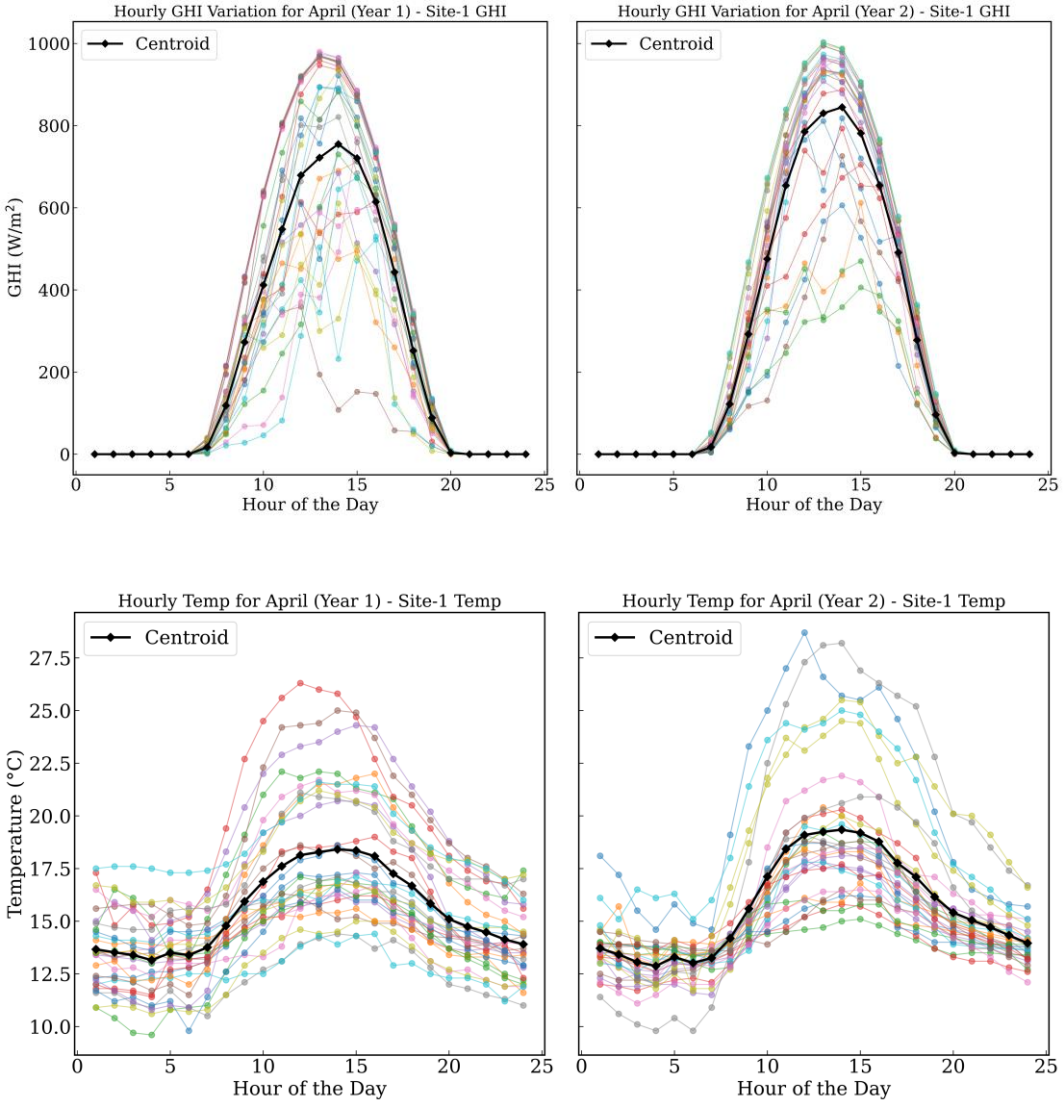
❖Descriptive Statistics

	Time	Load	Temperature (°C)					Global Horizontal Irradiance (W/m^2)				
			Site 1	Site 2	Site 3	Site 4	Site 5	Site 1	Site 2	Site 3	Site 4	Site 5
μ	Year 1	2162.82	17.31	17.17	18.27	17.72	17.60	225.80	222.47	226.91	227.06	228.39
	Year 2	2145.42	16.72	16.47	17.80	17.06	16.84	221.24	218.81	225.35	223.31	225.08
σ	Year 1	465.77	4.86	4.51	7.36	4.75	5.84	305.18	301.51	307.34	306.63	308.05
	Year 2	406.48	4.48	4.18	7.02	4.35	5.37	301.69	298.73	306.39	304.56	306.06
Median	Year 1	2072.00	17.50	17.40	17.30	17.80	17.30	12.00	12.00	12.00	12.00	13.00
	Year 2	2096.00	16.60	16.40	17.20	17.00	16.60	12.00	11.00	11.00	11.00	11.00
Min	Year 1	1101.00	1.90	2.90	-0.50	2.60	0.90	0.00	0.00	0.00	0.00	0.00
	Year 2	1027.00	4.60	4.50	0.20	3.90	2.90	0.00	0.00	0.00	0.00	0.00
Max	Year 1	4397.00	36.60	31.90	43.00	36.60	39.70	1037.00	1028.00	1041.00	1047.00	1049.00
	Year 2	3808.00	30.10	30.20	38.80	31.20	33.60	1031.00	1024.00	1035.00	1042.00	1045.00
γ_1	Year 1	1.18	0.09	-0.00	0.45	0.20	0.42	1.07	1.09	1.09	1.08	1.08
	Year 2	0.67	0.06	-0.02	0.37	0.11	0.27	1.11	1.11	1.09	1.11	1.10
γ_2	Year 1	2.10	-0.14	-0.33	-0.26	-0.02	0.02	-0.25	-0.19	-0.20	-0.23	-0.23
	Year 2	0.83	-0.36	-0.45	-0.45	-0.30	-0.34	-0.17	-0.13	-0.21	-0.17	-0.19

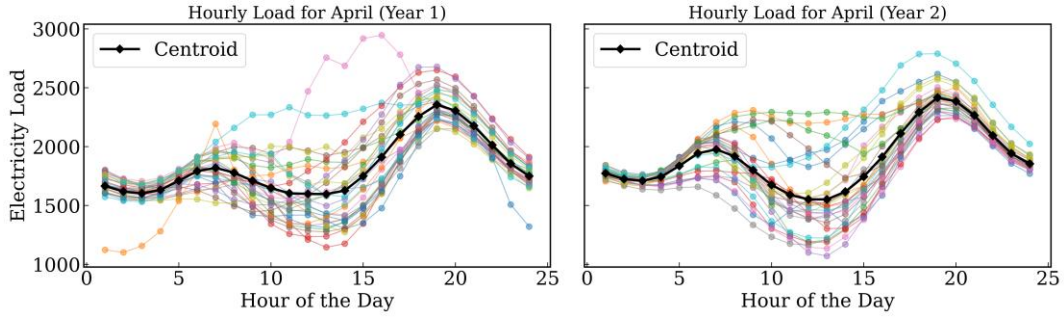
- γ_1: Third statistical moment (asymmetry of the distribution around its mean).
- γ_2: Fourth statistical moment ("tailedness" of a distribution relative to a normal distribution).

Data Overview (cont.)

❖Hourly Variational Analysis:



Hourly Electricity Load Variation for April

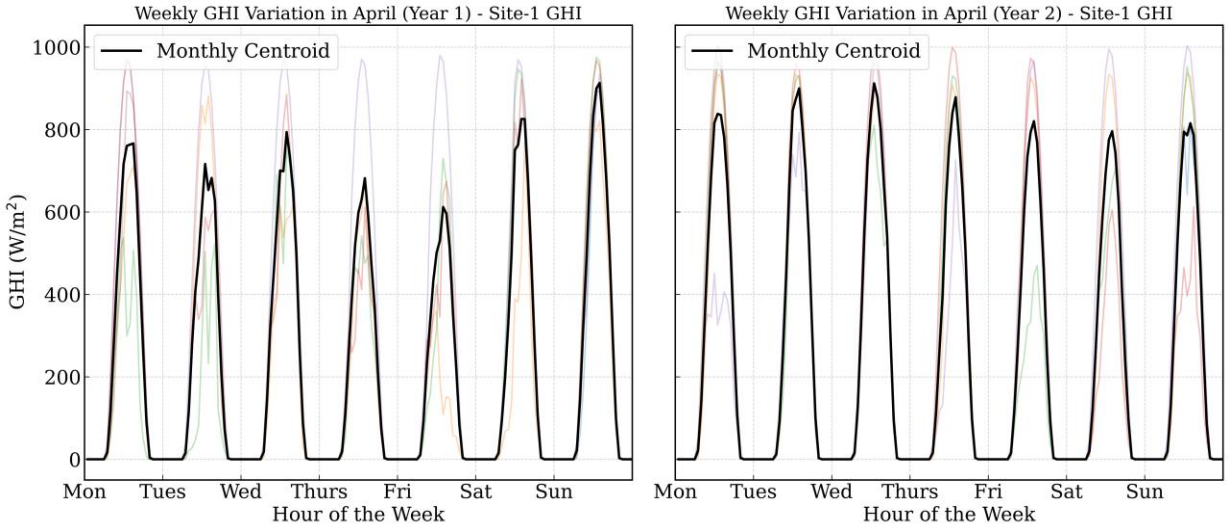


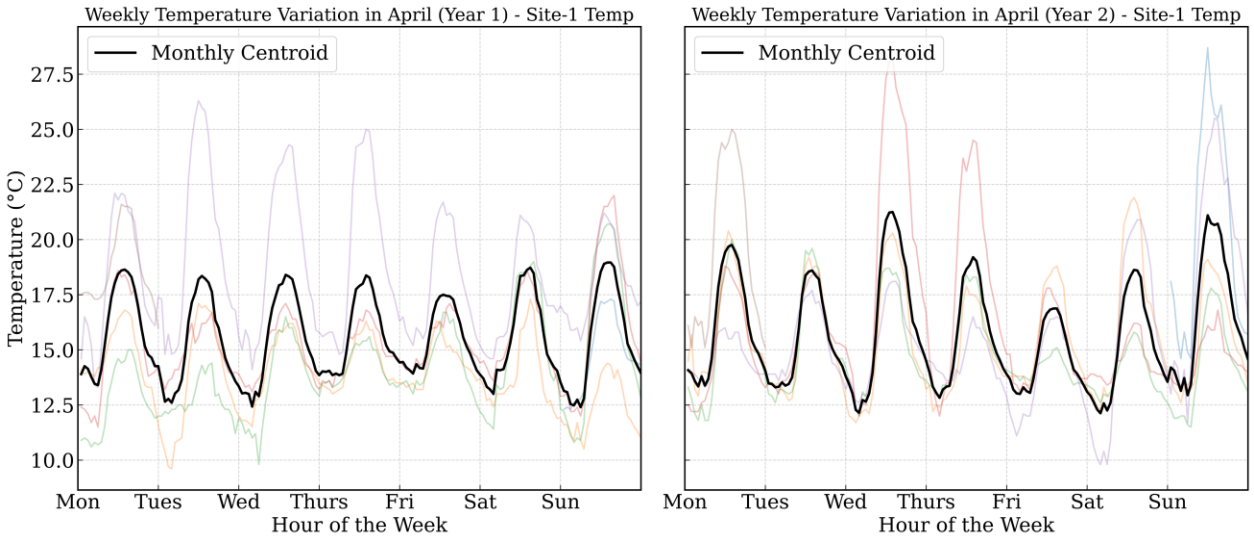
The greatest value of a picture is when it forces us to notice what we never expected to see

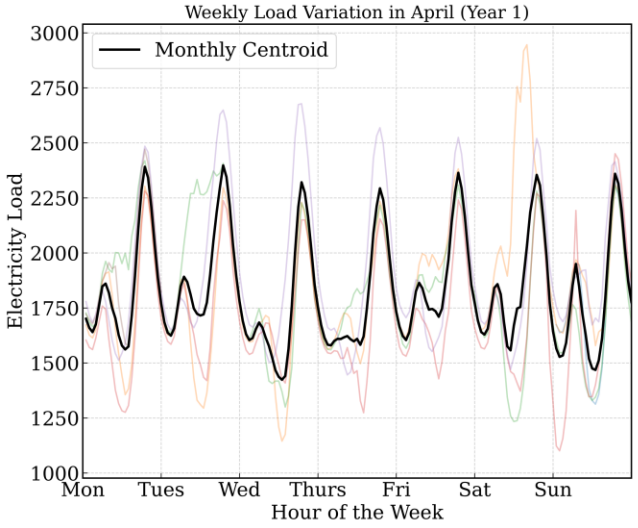
John W. Tukey

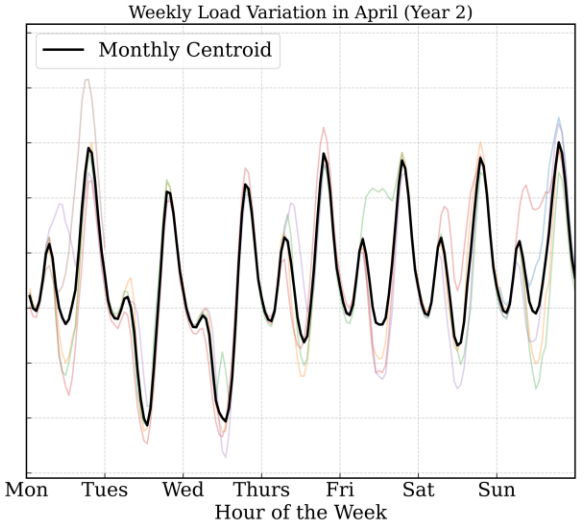
Data Overview (cont.)

❖Weekly Variational Analysis



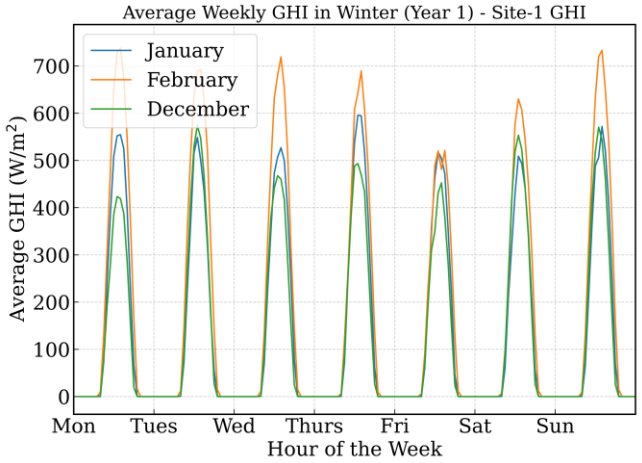


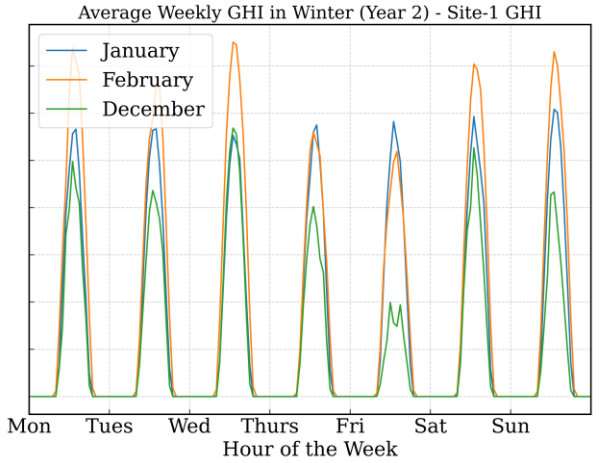


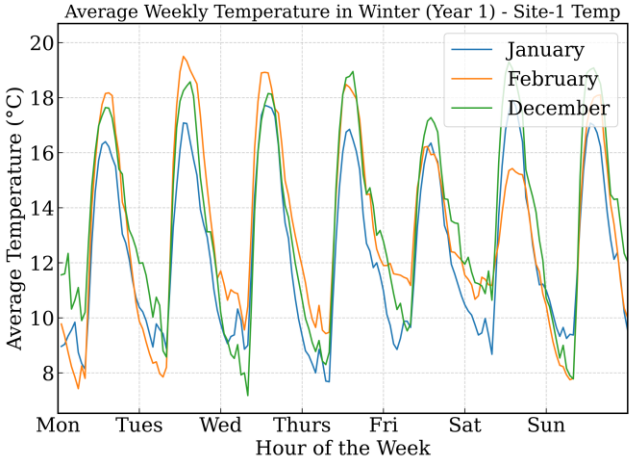


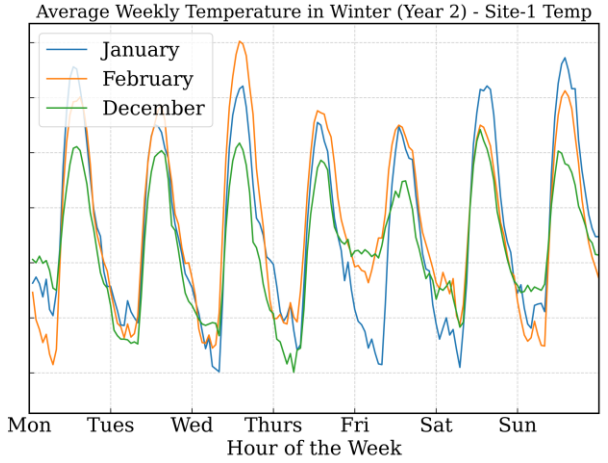
Data Overview (cont.)

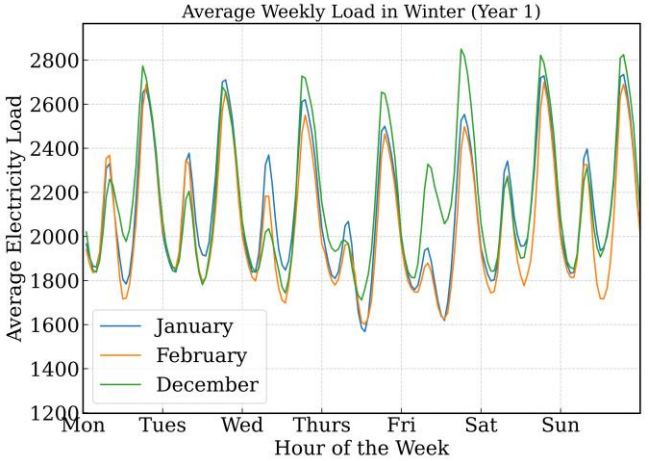
❖Seasonal Variational Analysis

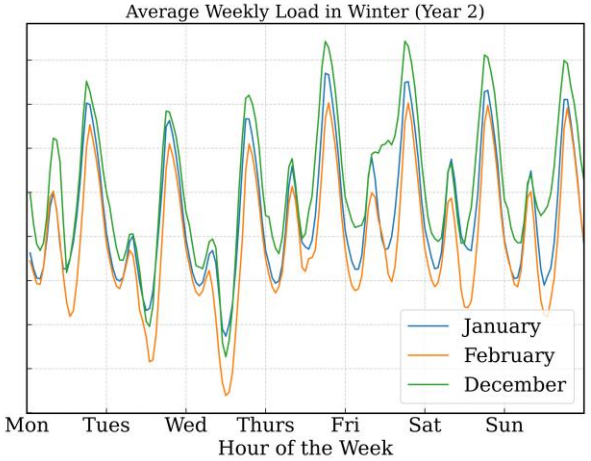






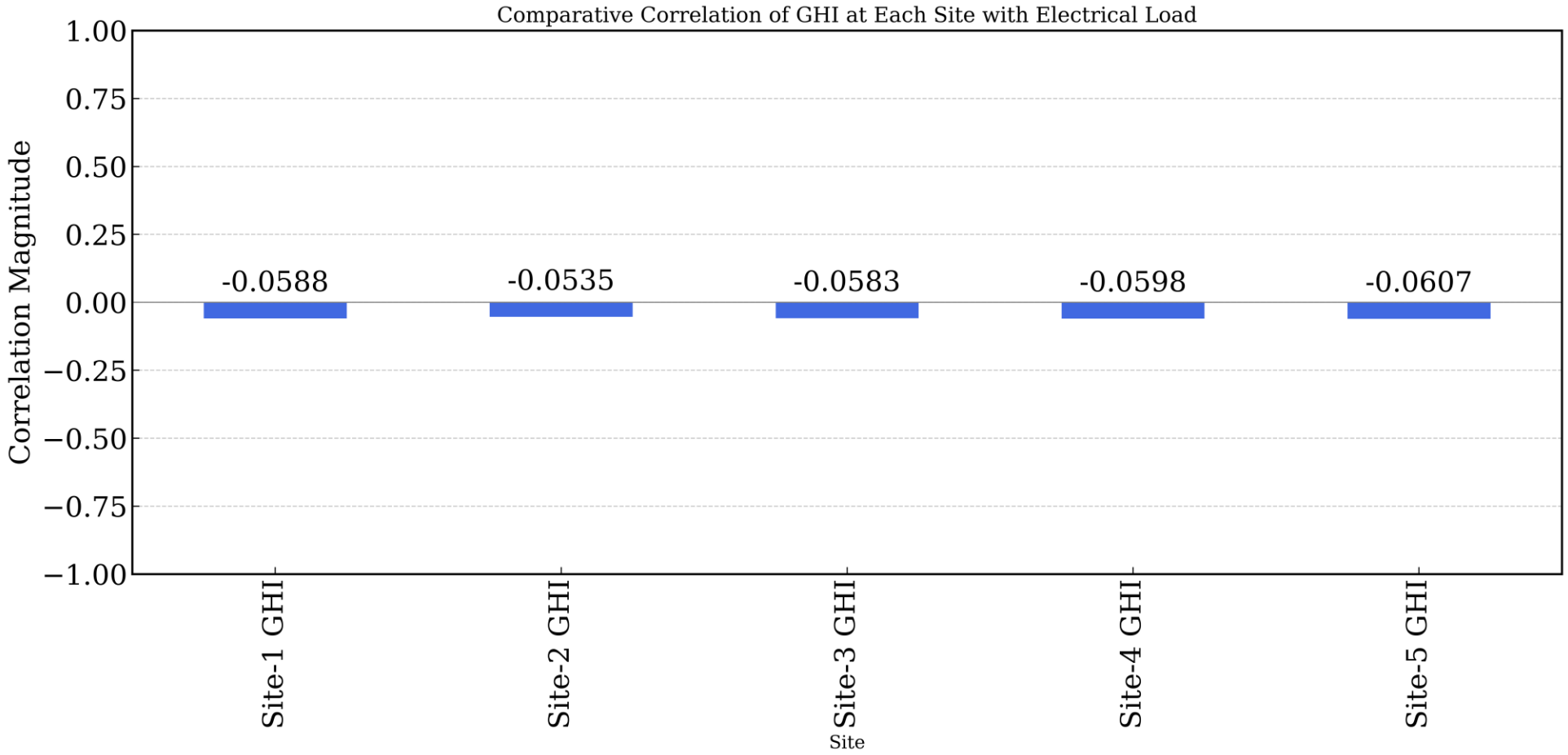






Data Overview (cont.)

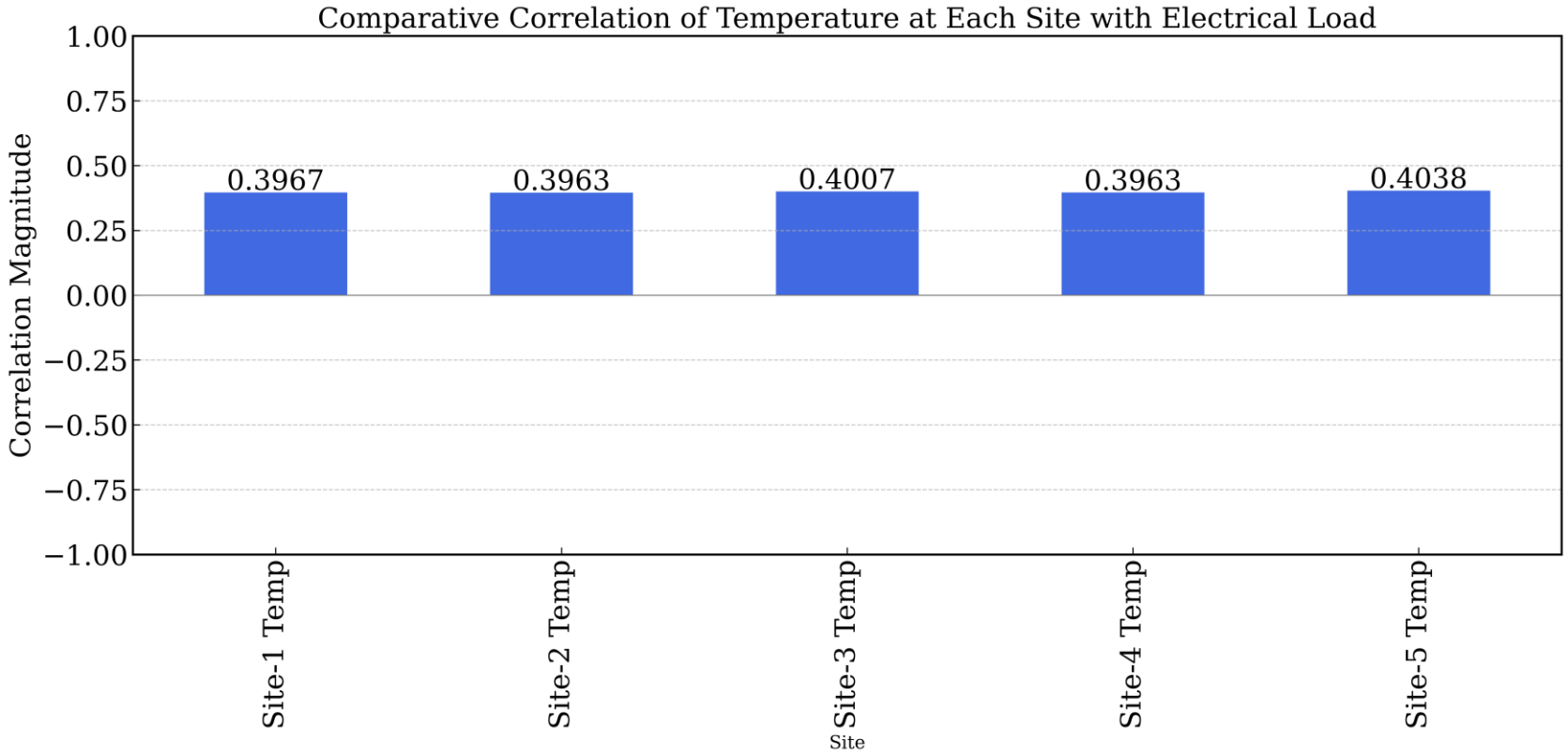
❖Correlation Analysis (Between Load and GHIs)



The **electric load** (equivalent to **power demand**) is negatively correlated to **GHI** across all sites (equivalent to the amount **power generated** from solar energy).

Data Overview (cont.)

❖Correlation Analysis (Between Load and Temperature)



The electric load (equivalent to **power demand**) is positively correlated to **ambient temperature** across all sites (possibly due to increased demand heating and cooling).

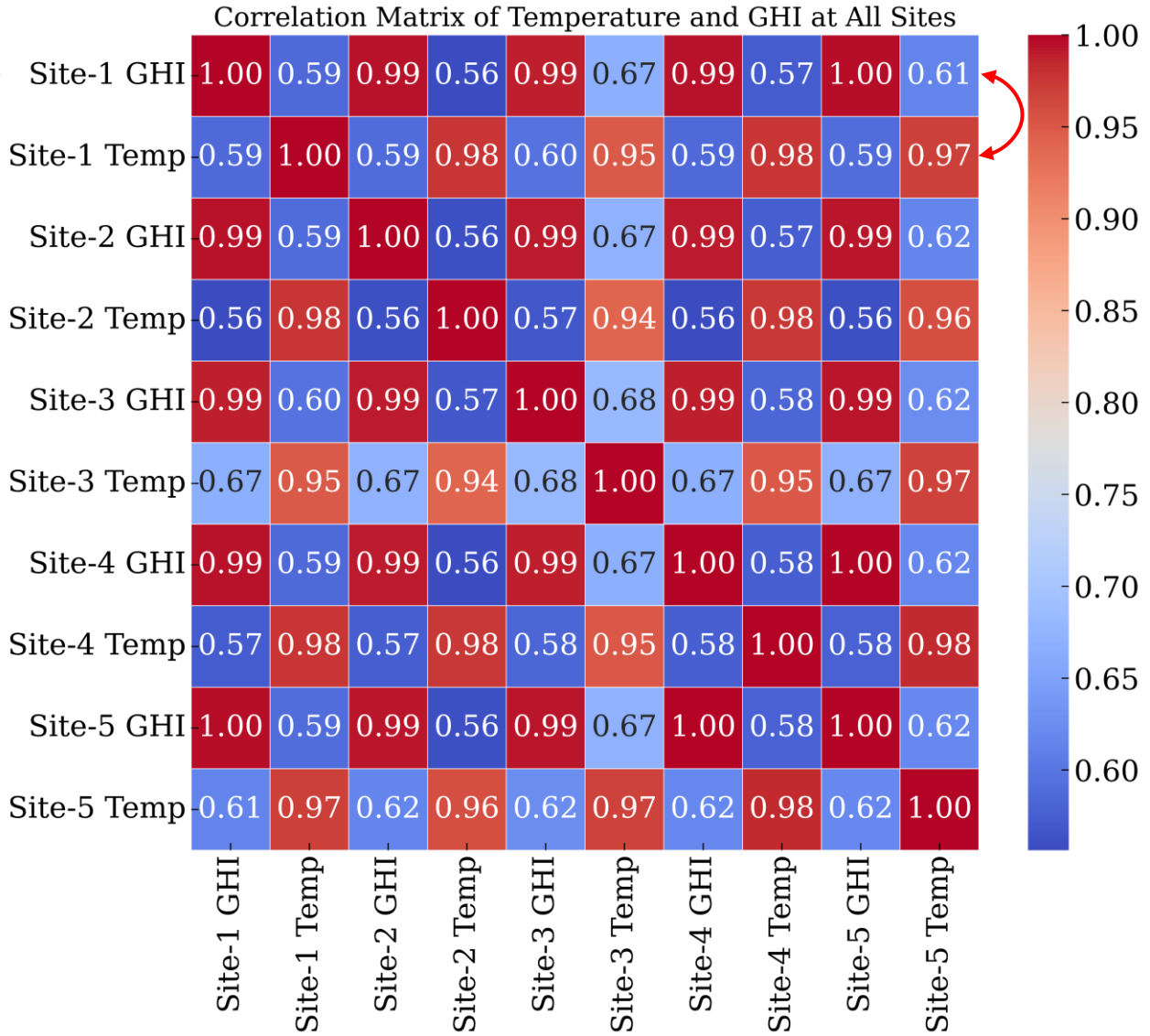
Data Overview (cont.)

❖Correlation Analysis (Between GHIs and Temperatures)

Inter-variable correlations are similar across site-variable pairs.

Share the same climate regime/region.

Can we merge these feature together?



Inter-Variable Correlation

Same for all Site variable pair

How can we handle all these features?

- **Combine Feature:** PCA, PLS
- Aggregate features: Clustering, Embedding.
- **Multicollinearity-robust models:** Ridge, Elastic Net; tree ensembles (RF, XGBoost)

Data Overview (cont.)

❖Collinearity Analysis (among GHIs and Temperatures):

Temp Site 1
GHI Site 1
Temp Site 2
GHI Site 2
…

Probabilistic Model

→ Electrical Load

Feature	VIF
Site-1 Temp	449.441269
Site-2 Temp	458.227488
Site-3 Temp	114.246325
Site-4 Temp	704.495510
Site-5 Temp	463.611923
Load	20.230474

More things should not be used than are necessary

William of Ockham

Should I use data from all sites, or a [linear] combination of them?

$$T = \phi_1 T_1 + \phi_2 T_2 + \phi_3 T_3 + \phi_4 T_4 + \phi_5 T_5$$

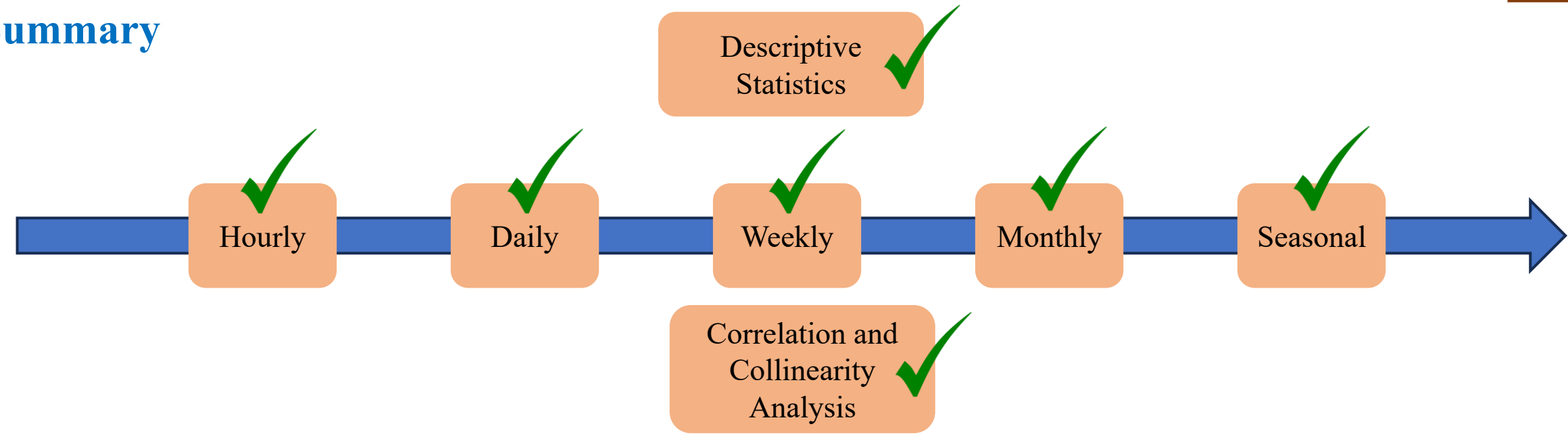
$$GHI = \theta_1 GHI_1 + \theta_2 GHI_2 + \theta_3 GHI_3 + \theta_4 GHI_4 + \theta_5 GHI_5$$

Feature	VIF
Site-1 GHI	279.130327
Site-2 GHI	189.029012
Site-3 GHI	144.154615
Site-4 GHI	1067.949872
Site-5 GHI	1630.840315
Load	1.489604

The **Variance Inflation Factor** (VIF) measures how much the variance of an estimated regression coefficient is increased because of collinearity.

Data Overview (cont.)

❖Summary



- Load fluctuations are **slightly different from one year to another**. This means while we can create validation cases of predicting one year using the training set as another, it should be kept in mind that two years have varying patterns.
- **Strong diurnal pattern** ($s = 24$) and **weekly effects** (weekend dips).
- Load **responds nonlinearly to temperature** (heating/cooling tails) **and GHI**.
- Cross-site weather features are **highly collinear**.

To understand time series data (from the EDA approach), we need to analyze data with **increased level of granularity**!

Mini Quiz

❖ Question 1

In the context of forecasting hourly electricity load, which of the following correctly identifies the variable types?

A. All variables (Load, Temperature, GHI) are exogenous.

B. Load is endogenous; Temperature and GHI are exogenous.

C. Load is exogenous; Temperature and GHI are endogenous.

D. All variables (Load, Temperature, GHI) are endogenous.

Think about which variable is the target to be explained by the model, and which variables are external inputs.

Mini Quiz

❖ Question 2

A data distribution with a flat top and thin tails compared to a normal distribution is described as?

A Positively Skewed

B Platykurtic

C Mesokurtic

D Leptokurtic

Kurtosis is the measure of a distribution's 'tailedness' or peak shape.

Mini Quiz

❖ Question 3

If the distribution of daily electricity load data has a long tail towards higher values, it is:

A Negatively skewed

B Symmetrical

C Positively skewed

D Bimodal

Consider which direction the 'tail' of outliers is pointing.

Outline

SECTION 1

Competition Overview

SECTION 2

Data Overview

SECTION 3

Modeling Strategy

SECTION 4

Evaluation

SECTION 5

Future Works

Modeling Strategy

❖Step-by-step approach

Dimension Reduction
- Use partial least squares (PLS) to reduce dataset's dimensions

Feature Engineering
- Time Features
- Weather Features
- Behavioral Features

Model Selection and Training
- Classical Models
- ML Models
- DL Models

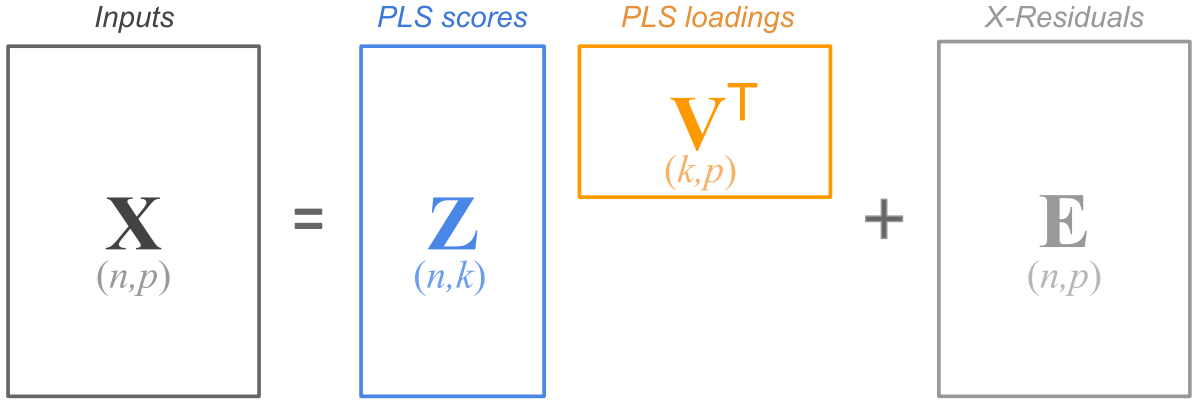
Metric Selection
- MAE
- MSE/RMSE
- MAPE
- ED

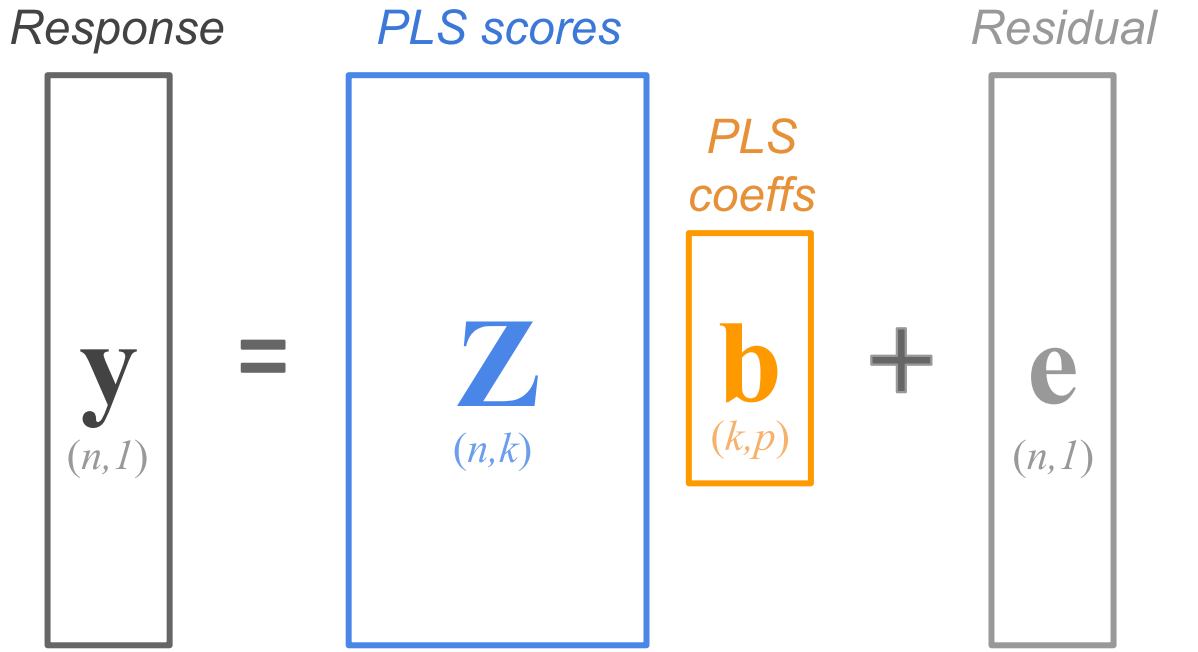
Model Training
- Distributed Computing
- Bayesian Optimization

Modeling Strategy (cont.)

❖Dimensional Reduction

- **Goal**: Find latent factors (linear combinations of predictors) that both:
  - Capture variance in the predictors (e.g., temperatures/GHI across sites).
  - Maximize their covariance with the response (e.g., electricity load).
- **How it works**: Unlike PCA (unsupervised, ignores the target), PLS is *supervised*—it extracts components that are most useful for prediction.
- **Why useful here**: PLS can compress them into one or two latent features that are most predictive of demand patterns.





https://allmodelsarewrong.github.io/pls.html

Modeling Strategy (cont.)

❖What if we do not use dimensionality reduction?

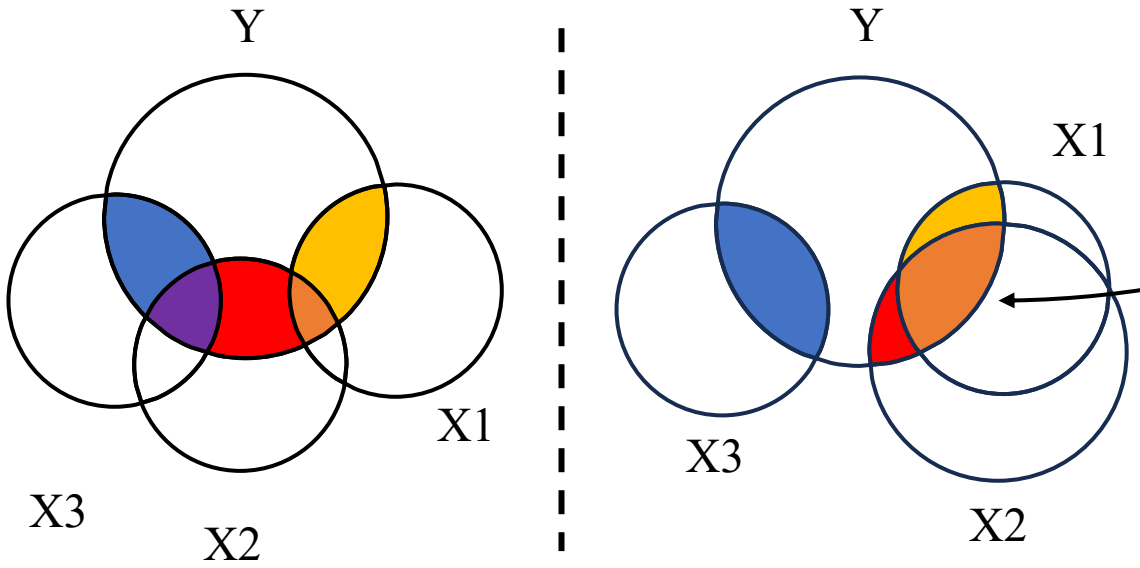
Multi-col-linear-ity

Referring to the multiple independent variables within multiple regression.

Referencing the linear movement in tandem correlation.

Occurring within a linear equation.

Example: If X2 increase 1.0 unit, X1 will increase 0.8 unit (because both is high correlation with together). Leading to an increase of 1.8 of Y.
→ Unstable prediction in real life.



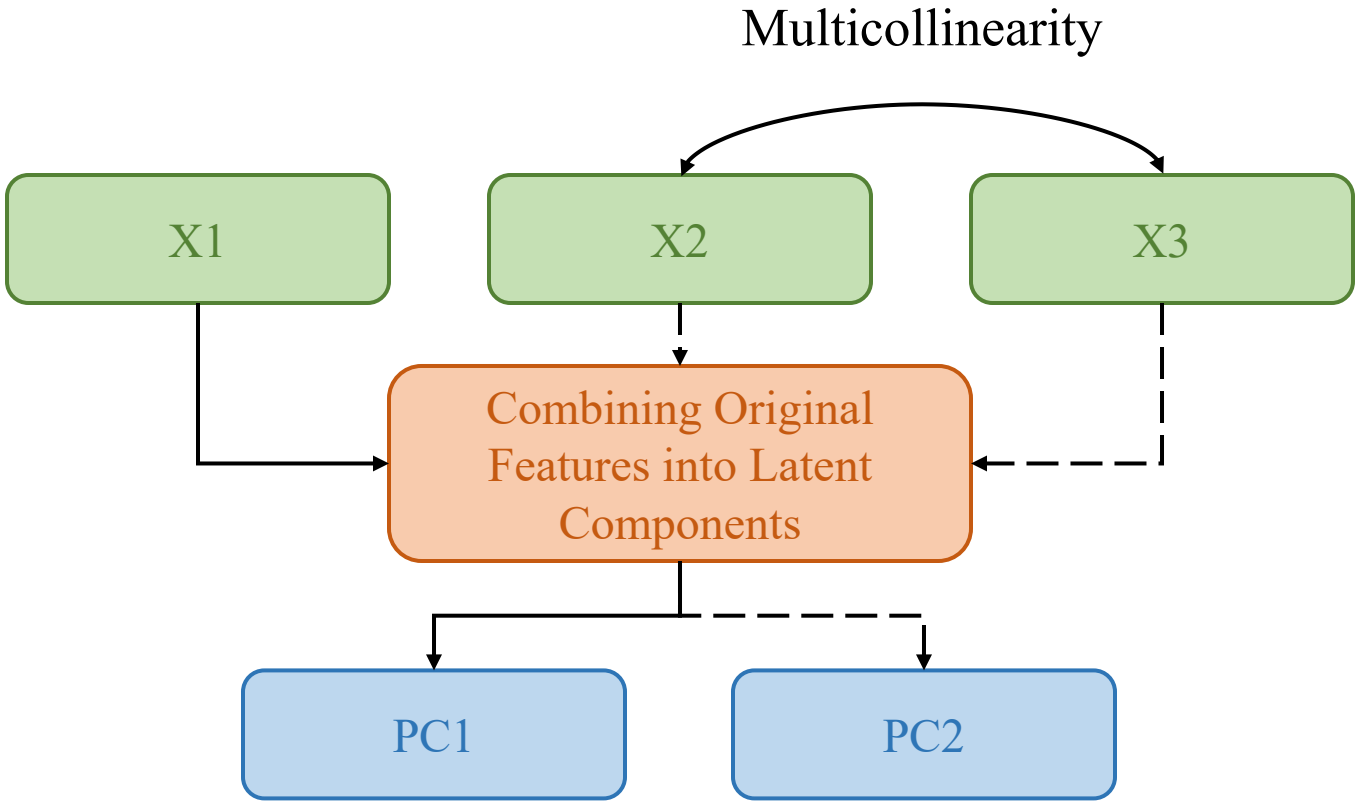
X1 and X2 have multicollinearity

Model don't know that variable to given weight

How to distribute the credit (the coefficient value) between two variables that provide the exact same information?

Modeling Strategy (cont.)

❖Principal Component Analysis (PCA)



PCA compresses data by creating new, uncorrelated components that capture the most variance.

Components are linear combinations of original variables; PCA is *unsupervised* (not used Y).

Flow:

- **Input:** Features X
- **PCA**: Finds new axes of maximum variance
- **Output:** New Features (PC1, PC2) called Principal Components

```
import numpy as np
from sklearn.decomposition import PCA
X = np.array(...)
pca = PCA(n_components=2)
pca.fit(X)
print(pca.explained_variance_ratio_)
print(pca.singular_values_)
```

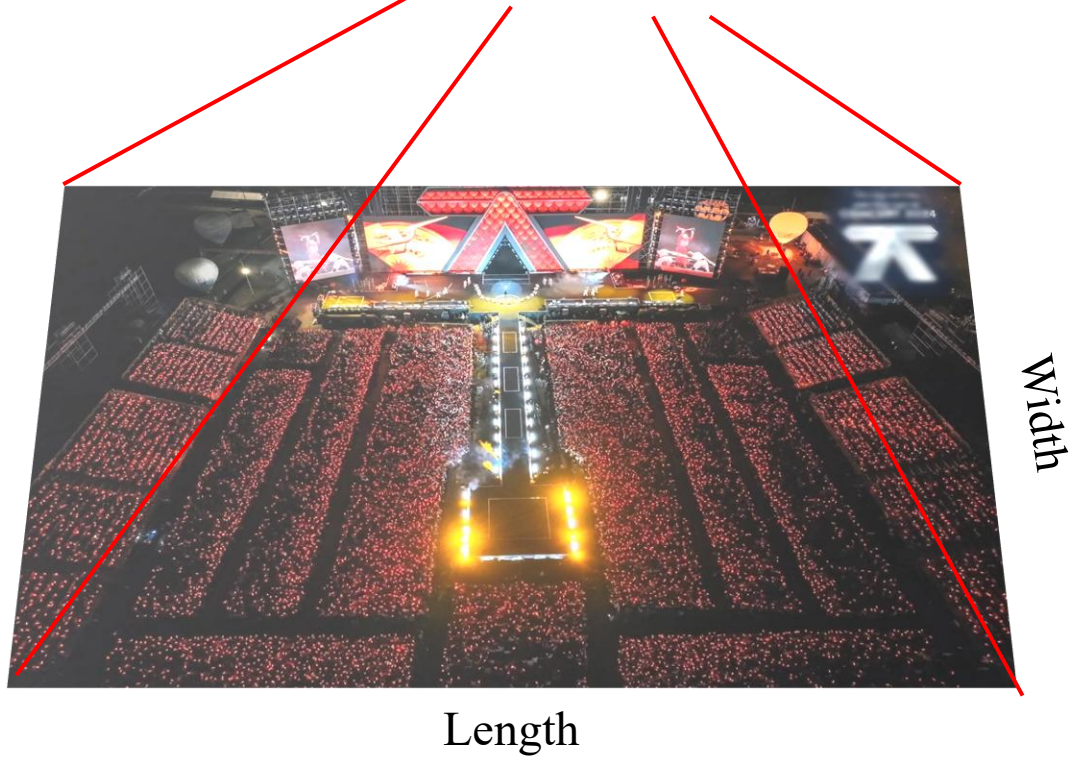
Modeling Strategy (cont.)

❖How PCR Thinks: Noise Level Prediction

PCA

What is best angle to capture the spread of the crowd?

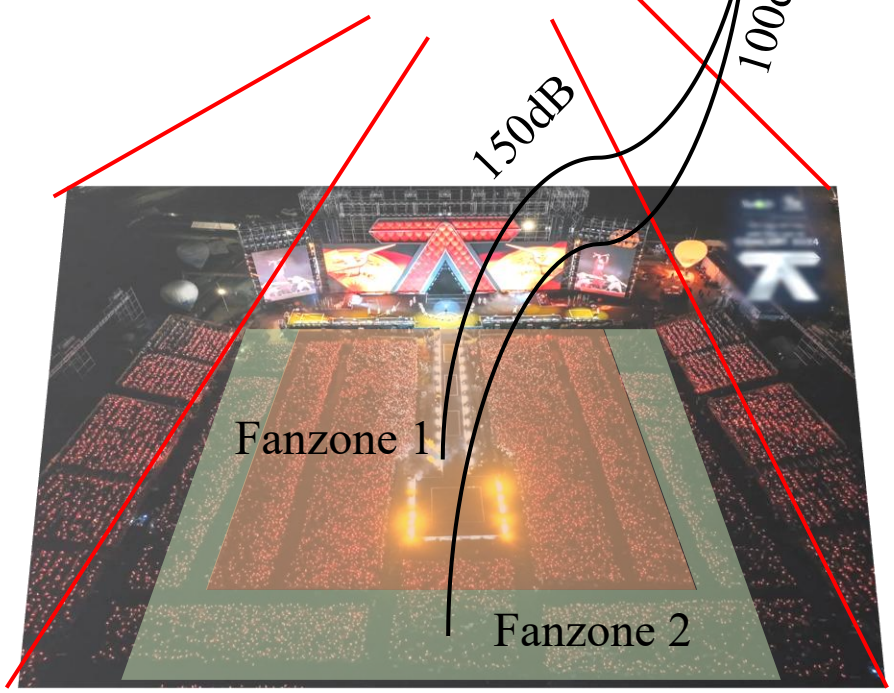
Using these features to predict noise is bad!



PLS

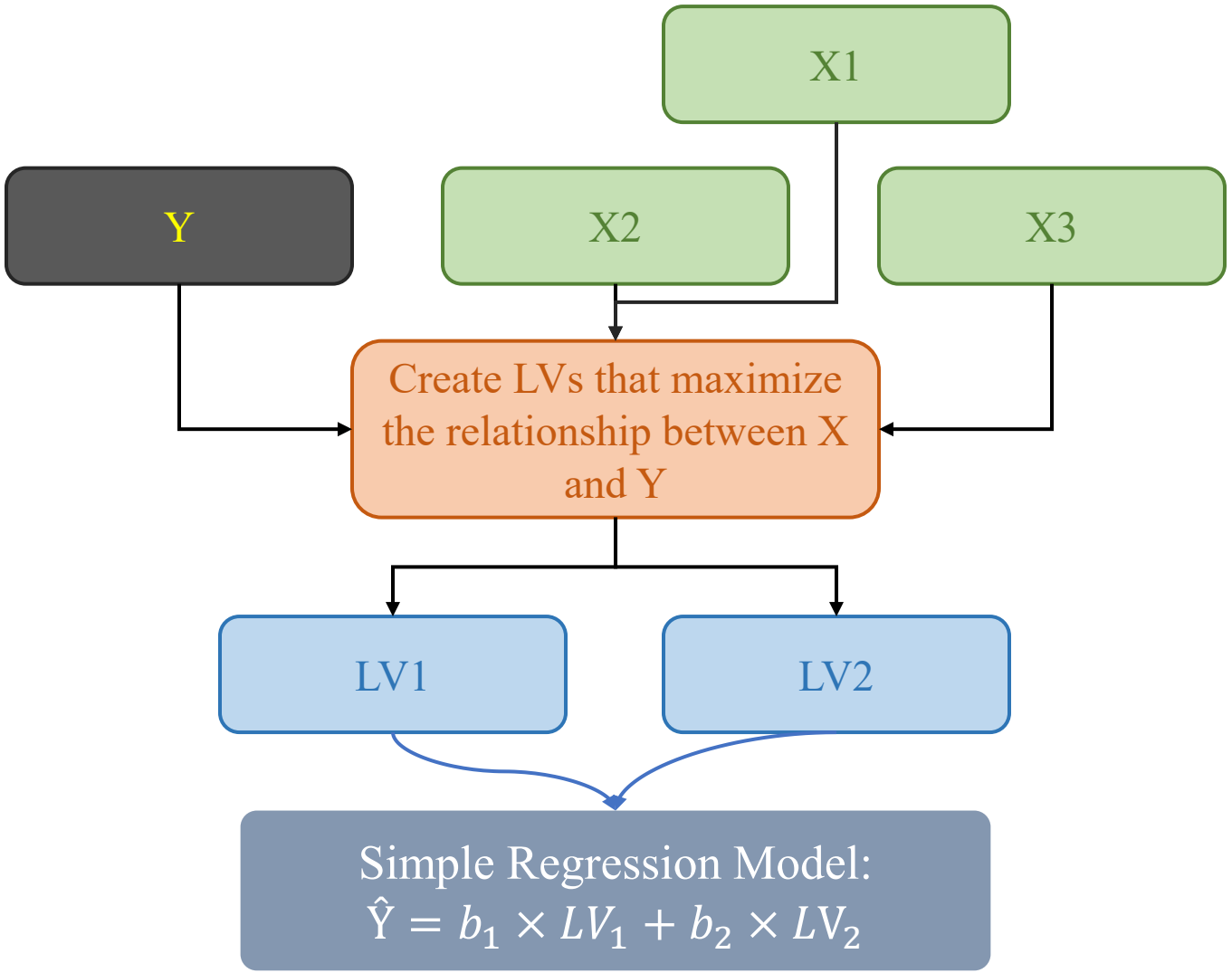
Need to see label for calculated how good feature is to describe target.

What is best angle to predict noise level?



Modeling Strategy (cont.)

❖Partial Least Squares (PLS)



A **supervised** dimensionality reduction technique designed for **prediction**. PLS creates new components (Latent Variables) from X that are designed to find a balance between representing X and predicting Y.

PLS:

- **Inputs:** Features X & Target Y
- **PLS Algorithm:** Finds components by maximizing Cov(X, Y)
- **Output:** Latent Variables

- Components can be hard to interpret.
- **Standardization** of X and Y is usually required.
- PLS is for linear; for nonlinear relationships, consider Kernel PLS.

Modeling Strategy (cont.)

❖Partial Least Squares (PLS)

```
4   from sklearn.cross_decomposition import PLSRegression
5
6   full_df = pd.read_csv("data.csv")
7
8   # Define predictor sets and the target variable ('Load')
9   temp_cols = sorted([col for col in full_df.columns if 'Temp' in col])
10  ghi_cols = sorted([col for col in full_df.columns if 'GHI' in col])
11
12  # Isolate the training data to fit the models
13  X_train_temp = train_df[temp_cols]
14  X_train_ghi = train_df[ghi_cols]
15  y_train = train_df['Load']
16
17  # a) PLS for Temperature
18  pls_temp = PLSRegression(n_components=1)
19  pls_temp.fit(X_train_temp, y_train)
20  full_df['Combined_Temp'] = pls_temp.transform(full_df[temp_cols])
21
22  # b) PLS for GHI
23  pls_ghi = PLSRegression(n_components=1)
24  pls_ghi.fit(X_train_ghi, y_train)
25  full_df['Combined_GHI'] = pls_ghi.transform(full_df[ghi_cols])
26
27  print(full_df['Combined_Temp'])
28  print(full_df['Combined_GHI'])
```

PLS is hybrid function, both a transformer and a regressor. Its have those function below:

- .fit(X, y): learn X and Y relationship.
- .transform(X): transform X into n components.
- .predict(X): Using linear model to both transform X and predict Y internally.

Column that have similar attibute should go together.

When there is a cluster of **highly correlated variables**, the first PLS component (n_components = 1) will almost certainly capture the most important “common signal” (e.g., the overall temperature level).

Mini Quiz

❖ Question 4

What is multicollinearity?

A. A situation where two or more predictor variables are highly correlated with each other.

B. A situation where predictor variables are highly correlated with the target variable.

C. A situation where the relationship between predictors and the target is non-linear.

D. A situation where the model's errors are not normally distributed.

This concept describes a problem within the set of predictor (X) variables, not their relationship with the target (Y).

Mini Quiz

❖ Question 5

What is the key difference between how PCA and PLS create their components?

A. PCA can only create one component, while PLS can create multiple.

B. PCA produces orthogonal components, while PLS components are correlated.

C. PCA creates components to maximize variance in X, while PLS creates components to maximize the covariance between X and Y.

D. PCA is supervised and uses the target Y, while PLS is unsupervised.

Think about what information each algorithm uses to find the 'best' new directions in the data.

Mini Quiz

❖ Question 6

Why can the process of building a PLS model be described as a 'greedy approach'?

A. Because it finds the single best component at each step and then works on the remaining information, without reconsidering previous components.

B. Because it requires the input data to be standardized first.

C. Because it uses a large amount of memory during computation.

D. Because it only works for linear relationships.

A greedy algorithm makes the **best choice** it can at the current moment.

Mini Quiz

❖ Question 7

In Scikit-learn, the class for PLS is named `PLSRegression`. Why?

A. Because PLS is a hybrid algorithm that is a complete regression model, capable of both transforming data and making predictions.

B. Because it is only used for transforming data, not for prediction.

C. Because it is an experimental feature and not part of the main library.

D. Because it is an older version of the algorithm, and the new one is just `PLS`.

Good luck ^^

Modeling Strategy (cont.)

❖Feature Engineering

Time Features

- Sinusoidal Encodings
- Day-of-week Encodings**

Weather Features

- Correlation-weighted average*
- Weather lags and deltas

Behavioral Features

- Holiday effect approximations**
- Heating/cooling degree days indicators**

[*]: Included if we are using all-site temperature and GHI
[**]: These are ideas from Catherine Wang, Ziying Ye, & Yue Jiang (1st place of 2025 PG&E Competition)

Modeling Strategy (cont.)

❖Feature Engineering (cont.)

Time Features

- Sinusoidal Encodings
- Day-of-week Encodings

For a time-index t and a known period P (e.g., 24 hours for daily):

$$x_{k,sin}^{PLS}(t) = \sin\left(\frac{2\pi kt}{P}\right), \qquad x_{k,cos}^{PLS}(t) = \cos\left(\frac{2\pi kt}{P}\right)$$

- P is period length
- 24 hours, 168 hours, 365.25 days).
- k is harmonic number (1 = fundamental frequency, 2 = second harmonic, 3 = third harmonics etc.).
- Each pair of $(\sin, \cos)$ captures a "wave" of frequency $\frac{k}{P}$.

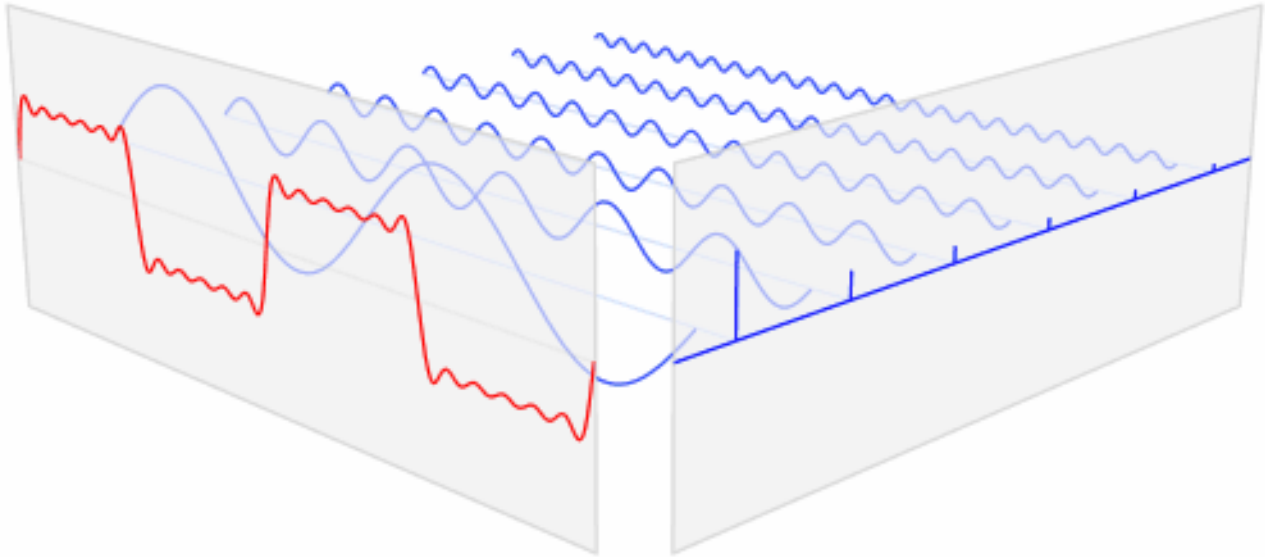
By including several k, the model can approximate more complex seasonal shapes.

- $k = 1$ for $P = 24$: captures "once-per-day" oscillation.
- $k = 2$ for $P = 24$: captures "day" and "night" oscillation.
- $k = 2$ for $P = 168$: captures weekday vs. weekend.
- $k = 3$ for $P = 168$: captures midweek vs. weekend vs. early-week patterns.
- $k = 4$ for $P = 8766$: captures four-seasons pattern.

Modeling Strategy (cont.)

❖Backgrounds behind Sinusoidal Encodings:

- Any complex signals can be broken down into the individual, simpler waves (called **harmonics**) of different frequencies and amplitudes that create it.
- Fourier Analysis transform a signal from the **time domain** (viewing its value over time) into the **frequency domain** (viewing which cyclical patterns are present). This transformation makes it possible to easily identify and analyze the underlying periodic components of complex data.



By breaking down the time index into its daily, weekly, and seasonal cyclical components, we are essentially pre-processing the data in a way that allows the XGBoost model to easily recognize and learn these recurring patterns.

Modeling Strategy (cont.)

❖Feature Engineering (cont.)

Time Features

- Sinusoidal Encodings
- Day-of-week Encodings

- Let $dow(t) \in \{0,1,2,3,4,5,6\}$ be the day of week for timestamp t, where 0 is Monday, 1 is Tuesday,…, 6 is Sunday.
- Let $wd(t) \in \{0,1\}$ be the indicator flag for the day of the week for timestamp t, where 0 means the day is weekday, 1 means it is weekend.
- Let $hd(t) \in \{0,1\}$ be the indicator flag for the day of the week for timestamp t, where 0 means the day is NOT a holiday, 1 means it is a HOLIDAY.

Modeling Strategy (cont.)

❖Feature Engineering (cont.)

Weather Features

- Correlation-weighted average*
- Weather lags and deltas

- **Lag features:**

$$lag_1(x^{PLS})_t = x_{t-1}^{PLS}$$
$$lag_{24}(x^{PLS})_t = x_{t-24}^{PLS}$$

  - $lag_1(x^{PLS})_t$: the immediate previous hour's value.
  - $lag_{24}(x^{PLS})_t = x_{t-24}$: the same hour on the previous day (captures daily persistence).

- **Difference features:**

$$\Delta_1(x^{PLS})_t = x_t^{PLS} - x_{t-1}^{PLS}$$
$$\Delta_{24}(x^{PLS})_t = x_t^{PLS} - x_{t-24}^{PLS}$$

  - $\Delta_1(x^{PLS})_t$: short-term hour-to-hour change (captures ramps/spikes)
  - $\Delta_{24}(x^{PLS})_t = x_{t-24}$: day-over-day change at the same hour (captures seasonal drifts, unusual deviations).

Modeling Strategy (cont.)

❖Feature Engineering

Behavioral Features

- Holiday effect approximations**
- Heating/cooling degree days indicators**

Let:

- T_t^{PLS} be the temperature at time t
- T_{base} be comfort baseline (commonly $20°C$).

Then:

- **Cooling Degree Hours (CDH):** how many °C above baseline

$$CDH_t = \max(0, T_t^{PLS} - T_{base})$$

- **Heating Degree Hours (HDH):** how many °C below baseline

$$HDH_t = \max(0, T_{base} - T_t^{PLS})$$

Modeling Strategy (cont.)

❖Model Selection

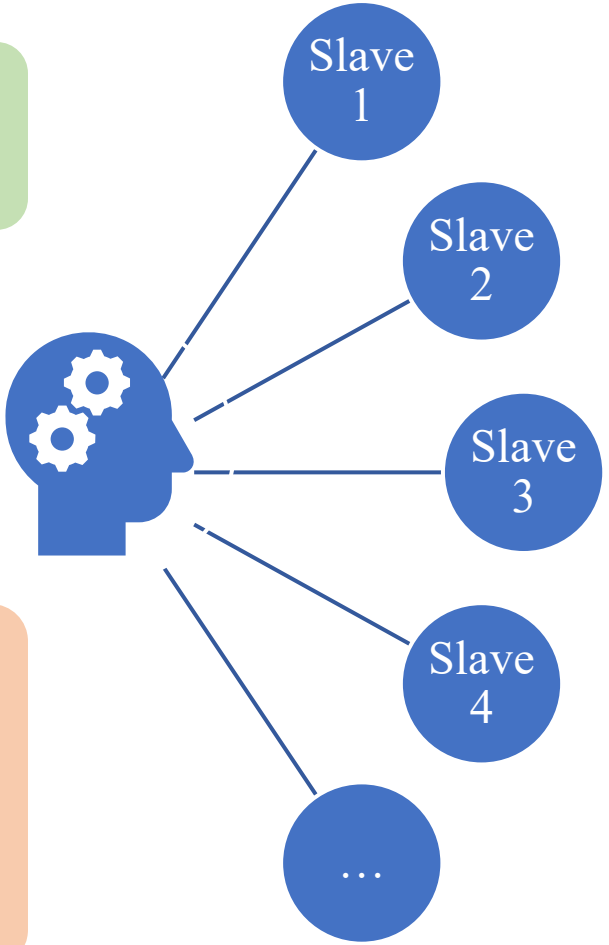
Model	Selected?	Why?
Linear Models	No	Assumes stationarity and normality Performs poorly
Random Forest	Yes	Captures nonlinearity Robust to missing contexts
XGBoost	Yes	Handles complex interaction Effective with limited metadata
LSTM	Yes	Learns long-range temporal dependencies Suited for sequential structure
Transformers	No	Overkill for small data High compute cost Low interpretability

Let us consider **XGBoost** as our primary model.

Modeling Strategy (cont.)

❖Model Training (Greedy approach for Hyperparameter Tuning)

Distributed Computing (MPI) is used to train multiple models at the same time.



Master (rank 0):

- Enumerates all hyperparameter combinations from the grid.
- Dispatches one combination at a time to a worker.
- Collects back results (validation score, best iteration, etc.).
- Assigns new jobs to idle workers until grid is exhausted.

- Number of estimators: {100, 200, 300, … }
- Maximum depths: {3, 5, 7, … }
- Learning rate: {0.01, 0.05, 0.1, … }
- Subsample: {0.8,1.0, … }

Slaves (rank > 0):

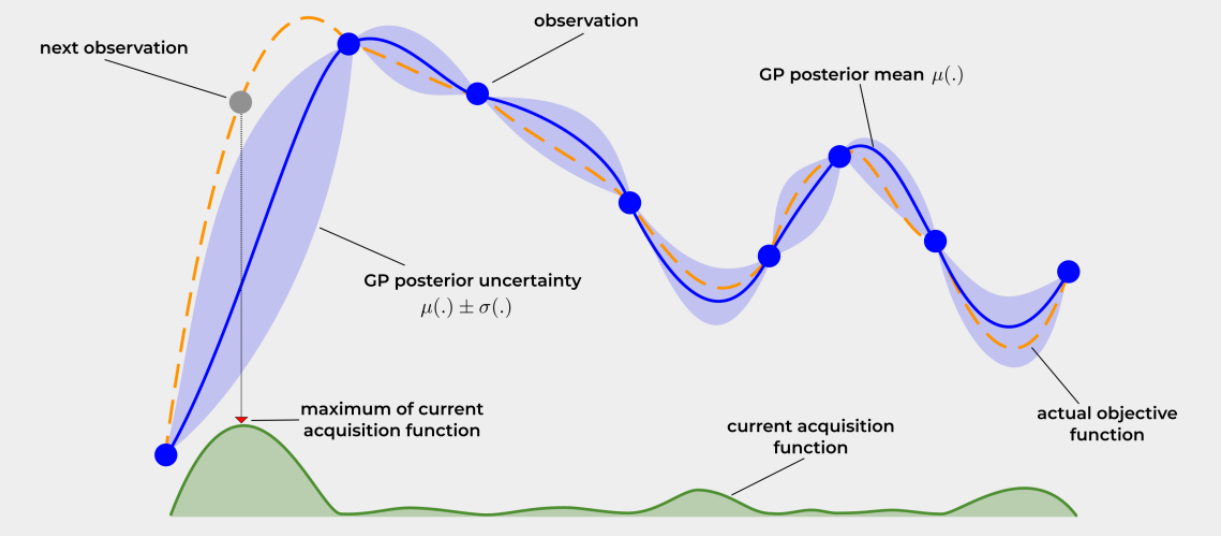
- Receive one hyperparameter dict.
- Train an XGBoost model using those parameters.
- Send evaluation metrics (e.g., MSE, RMSE, MAPE) and parameter set back to master.

Modeling Strategy (cont.)

❖Model Training (Bayesian Optimization approach for Hyperparameter Tuning)

- It treats the tuning process like searching for oil. It starts by "drilling" a few random spots (testing a few random sets of parameters) to build a probabilistic "map" of the performance landscape. This map, called a **surrogate model**, estimates how good the model's performance might be in unexplored areas. This allows the optimizer to make informed, data-driven decisions about where to "drill" next to find the best possible set of hyperparameters.

- The "intelligence" of Bayesian Optimization comes from its **acquisition function**, which guides the search process. At each step, this function carefully balances **exploitation** (testing in areas where the map predicts high performance) with **exploration** (testing in areas of high uncertainty, where an unexpectedly great result might be hiding).

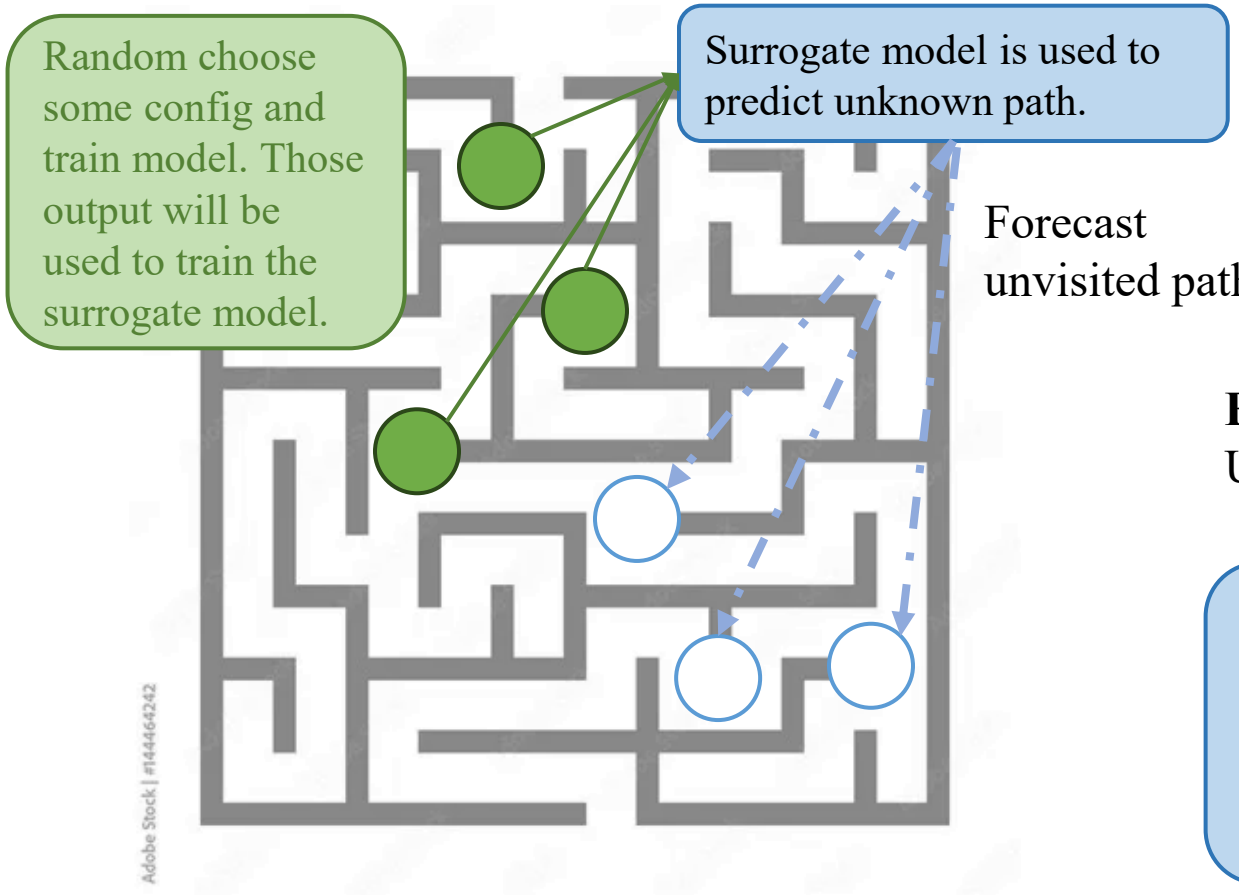


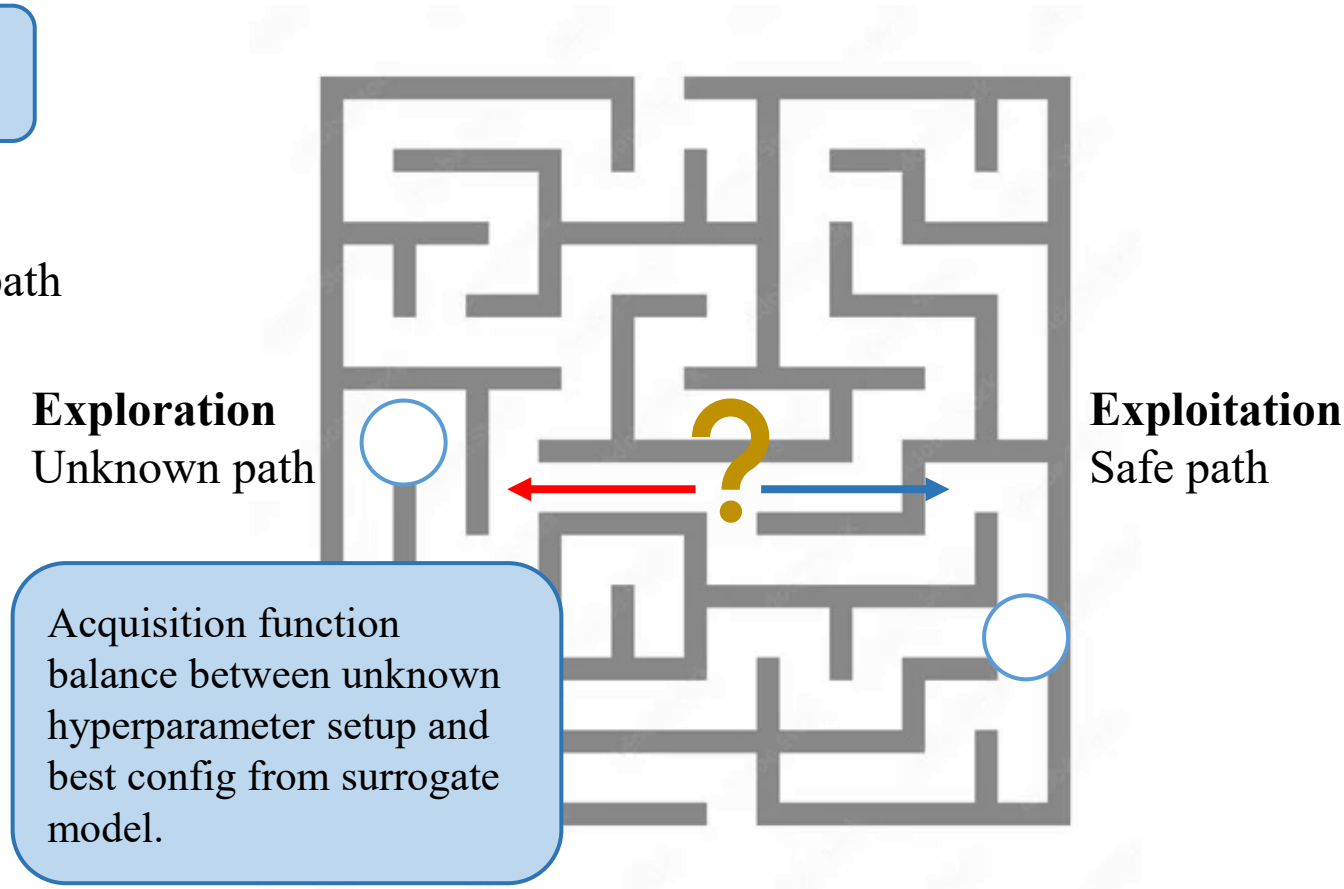
Modeling Strategy (cont.)

❖Model Training (Bayesian Optimization approach for Hyperparameter Tuning)

Step 1: Build surrogate model

Step 2: Acquisition function as the guide

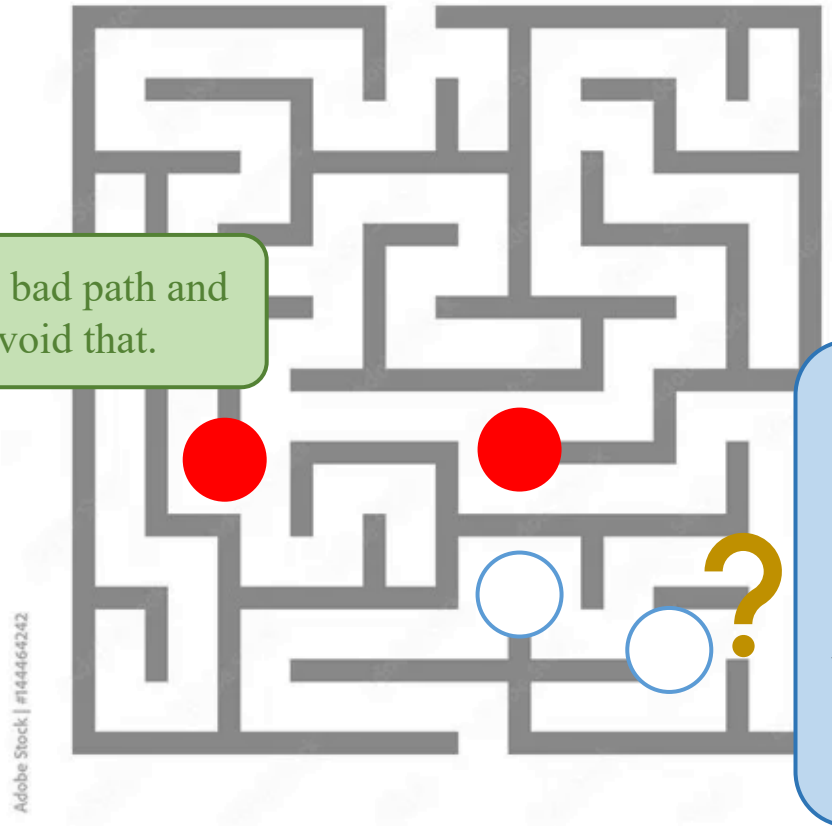




Modeling Strategy (cont.)

❖Model Training (Bayesian Optimization approach for Hyperparameter Tuning)

Step 3: Update the map



Mark bad path and avoid that.

The 'guide' (Acquisition Function) suggests the next best path to try based on the updated map.

```python
# --- Define the Bayesian Optimization Function ---
# This function takes hyperparameters as input, trains a model, and returns a score to maximize.
def xgb_objective_function(max_depth, learning_rate, gamma, colsample_bytree, subsample):
    # The optimizer passes float values, but some parameters need to be integers
    params = {
        'max_depth': int(max_depth),
        'learning_rate': learning_rate,
    }
    model = xgb.XGBRegressor(n_estimators=1000, **params)

    # Use early stopping to find the best number of trees
    model.fit(X_train_opt, y_train_opt,
              eval_set=[(X_val_opt, y_val_opt)],
              verbose=False)

    predictions = model.predict(X_val_opt)
    rmse = np.sqrt(mean_squared_error(y_val_opt, predictions))

    # BayesianOptimization maximizes, so we return negative RMSE (lower is better)
    return -rmse

# --- Define Hyperparameter Search Space and Run Optimization ---
param_bounds = {
    'max_depth': (3, 10),
    'learning_rate': (0.01, 0.3),
}

optimizer = BayesianOptimization(
    f=xgb_objective_function,
    pbounds=param_bounds,
    random_state=42,
    verbose=2 # Set to 2 to see the search progress
)
```

Metrics to calculate our path, RMSE in this case.

Mini Quiz

❖ Question 8

What are the primary reasons for selecting XGBoost as the main model?

A It learns long-range temporal dependencies.

B It assumes stationarity and normality.

C It handles complex interactions and is effective with limited metadata.

D It has low computational cost and high interpretability.

Refer to the 'Model Selection' table in the 'Modeling Strategy' section.

Mini Quiz

❖ Question 9

In the 'Greedy approach' for hyperparameter tuning described, what is the primary role of the 'Master' node?

A. To enumerate all parameter combinations and dispatch them to worker nodes.

B. To receive a single hyperparameter set and train one XGBoost model.

C. To use an acquisition function to balance exploration and exploitation.

D. To build a probabilistic surrogate model of the performance landscape.

Review the 'Model Training (Greedy approach)' slide and differentiate the roles of the Master and Slaves.

Mini Quiz

❖ Question 10

According to the Bayesian Optimization, what is the role of the 'acquisition function'?

A. To guide the search by balancing testing in promising areas (exploitation) with testing in uncertain areas (exploration).

B. To only test in areas of high uncertainty where new results might be hiding.

C. To build the probabilistic 'map' (surrogate model) of the performance landscape.

D. To test every single hyperparameter combination in a predefined grid.

This function is the 'intelligence' that decides where to search for parameters next.

Outline

SECTION 1

Competition Overview

SECTION 2

Data Overview

SECTION 3

Modeling Strategy

SECTION 4

Evaluation

SECTION 5

Future Works

Evaluation

❖Feature Integration and Final Model Summary

Model ID(s)	Features					MSE	RMSE	MAE	MAPE (%)	ED
	Raw Data	Dimensional Reduction	Time Feature	Weather Feature	Behavioral Feature					
[1]	Y	N	N	N	N	25814.90	160.67	124.98	5.70	6.54
[2]	N	Y	N	N	N	19944.42	141.22	103.72	4.66	4.22
[3]	N	Y	Y	N	N	1884.32	137.42	98.28	4.35	3.78
[4]	N	Y	Y	Y	N	17.149.40	130.96	95.63	4.28	3.01
[5]	N	Y	Y	Y	Y	17359.21	131.75	94.14	4.11	2.49

Final Model Hyperparameter

- Number of estimators: {100, **200**, 300, ...}
- Maximum depths: {3, 5, **7**}
- Learning rate: {**0.01**, 0.05, 0.1}
- Subsample: {0.8, **0.8889**,1.0}

Outline

SECTION 1

Competition Overview

SECTION 2

Data Overview

SECTION 3

Modeling Strategy

SECTION 4

Evaluation

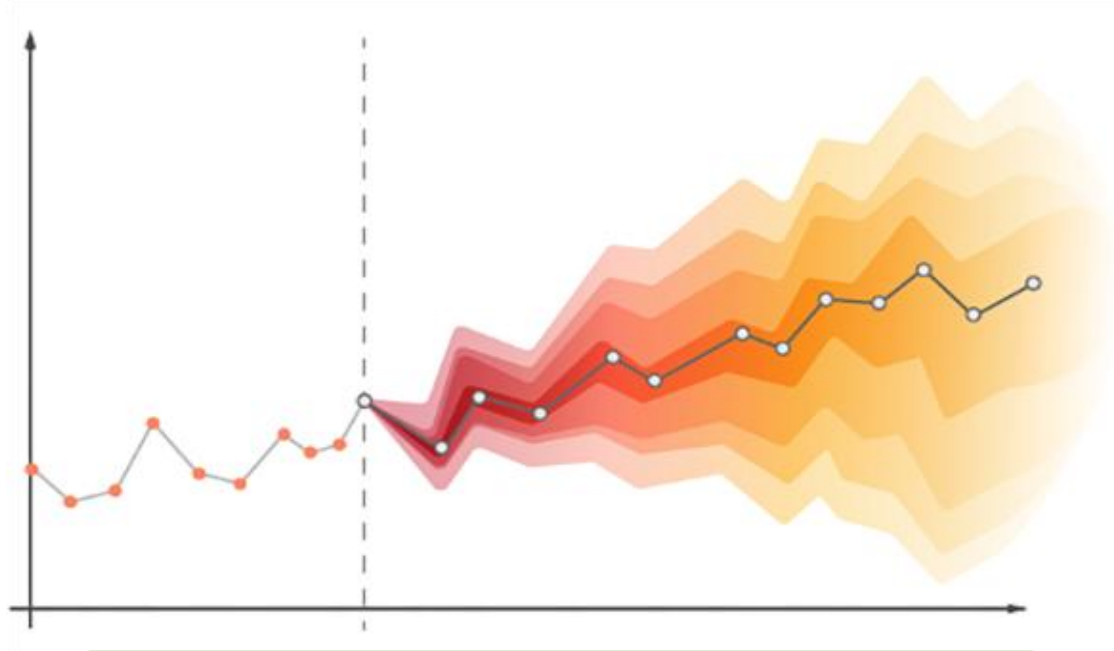
SECTION 5

Future Works

Future Works

❖Probabilistic Forecasting

- **Grid Stability and Risk Management**: They need to know the **uncertainty spread** to plan reserves and prevent blackouts.
- **Renewables integration**
  Solar/wind are volatile. Probabilistic forecasts allow operators to hedge: For instance, *"there's a 90% chance demand will exceed 5 GW at 6 pm, so we should commit extra capacity"*.
- **Market & financial operations**
  Energy markets clear based on not just expected demand but also **risk of under- or over-estimation**. Confidence intervals matter for pricing and bidding.



- **Quantile regression with XGBoost** (P10/P50/P90).
- **Conformal prediction** for calibrated intervals.
- **Bootstraps** for distributional spread.
- **Bayesian & deep probabilistic models** for richer uncertainty.