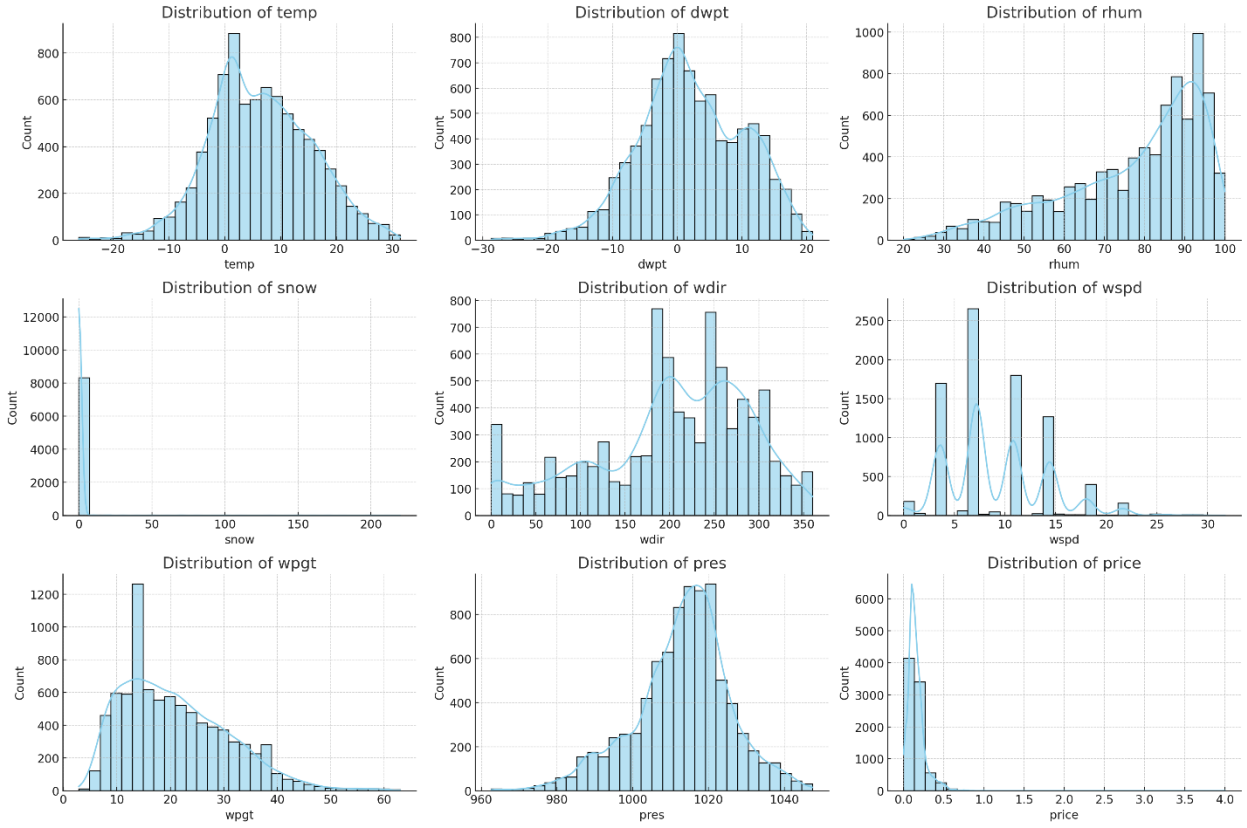
1. Analysis

The dataset consists of time series data for several weather and electricity variables, including temperature, humidity, snow depth, wind speed, and electricity price and demand. The primary goal is to analyze, clean, and model the data to understand electricity demand

Statistical Description:

	temp	dwpt	...	price	demand
count	8424.000000	8424.000000	...	8424.000000	8338.000000
mean	6.435708	2.228359	...	0.156541	1.055267
std	9.064983	8.053260	...	0.112922	1.105235
min	-26.100000	-28.700000	...	0.000070	0.000000
25%	0.300000	-3.000000	...	0.092667	0.367000
50%	6.000000	1.700000	...	0.135120	0.819500
75%	12.800000	8.400000	...	0.198222	1.372000
max	31.400000	20.900000	...	4.000000	10.381000

[8 rows x 10 columns]



temp, dwpt, and *rhum* - typical ranges for weather data.

price - wide range, from nearly zero to 4.0, indicating significant variability.

Distribution plots revealed most variables were normally or skewed distributed. snow was heavily skewed due to the filling strategy.

Missing Data Summary:

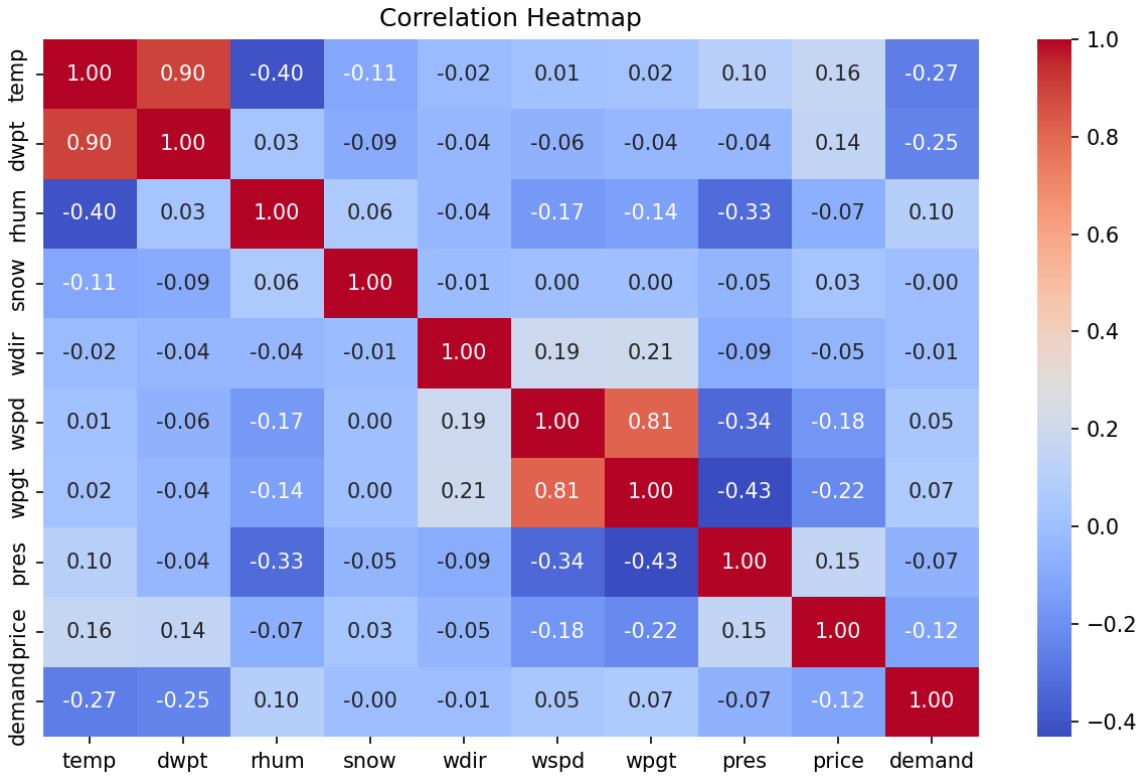
snow *8305*

demand *86*

The *snow* column had significant missing values (8305), which were filled with 0

(assumption: no snowfall when missing).

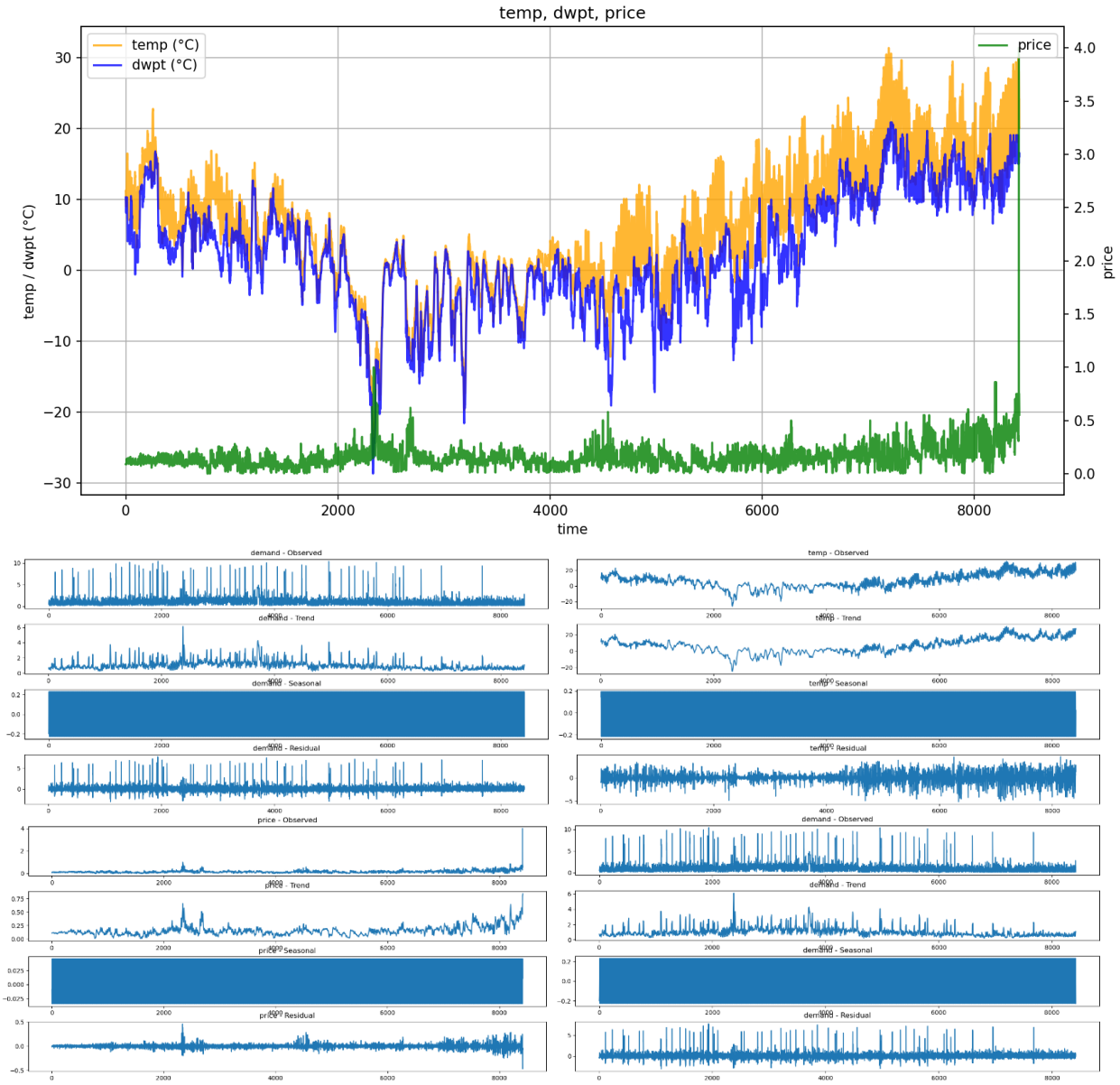
demand column had 86 missing values that might need further imputation column with the median.



temp and *dwpt* had a strong positive correlation.

price and *demand* appeared weakly correlated.

Two visualizations were used: a histogram for each weather variable to understand their distributions and a scatter plot showing the relationship between *temp* and *demand*. These visuals highlighted that *temp* and *dwpt* had a strong positive correlation, indicating their potential impact on energy demand.



Transformation

Standardization scales all numerical variables to a mean of 0 and a standard deviation of 1.

Feature Ranking

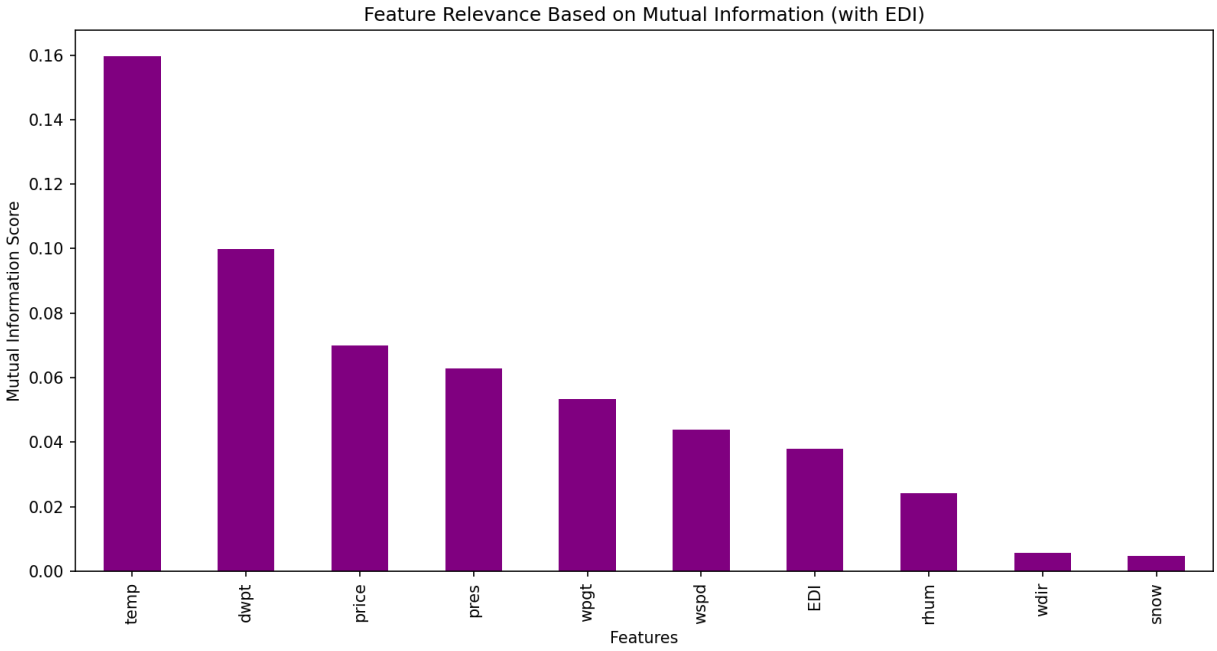
The mutual_info_regression function calculates the relevance of each feature with the target (demand).

New Feature - Energy Demand Interaction (EDI)

Interaction of energy price and weather. Defined as the interaction between energy price and temperature differences

price * |*temp* - *dwpt*|

This measures how extreme weather impacts energy consumption in cost-sensitive conditions.



Feature	Relevance Ranking
temp	0.159690
dwpt	0.099853
price	0.069867
pres	0.062795
wpgt	0.053310
wspd	0.043776
EDI	0.038014
rhum	0.024098
wdir	0.005730
snow	0.004704

Mutual Information Rankings:

temp - The most relevant feature affecting demand, likely due to its strong impact on heating/cooling needs.

dwpt - Second most relevant, indicating the impact of humidity on energy demand.

price - Economic factor influencing demand, ranked third.

EDI - Newly created feature ranked sixth, showing modest relevance, combining price and temperature variations.

Observations:

temp and *dwpt* dominate the relevance ranking, highlighting the strong influence of weather variables on energy demand.

EDI adds new interpretability but shows moderate correlation compared to *temp* and *dwpt*

2. Modeling

Stationarity Test

Used the Augmented Dickey-Fuller (ADF) test to check if the demand data is stationary.

If the p-value is greater than 0.05, it means the data is not stationary (trends or seasonality may exist).

Results

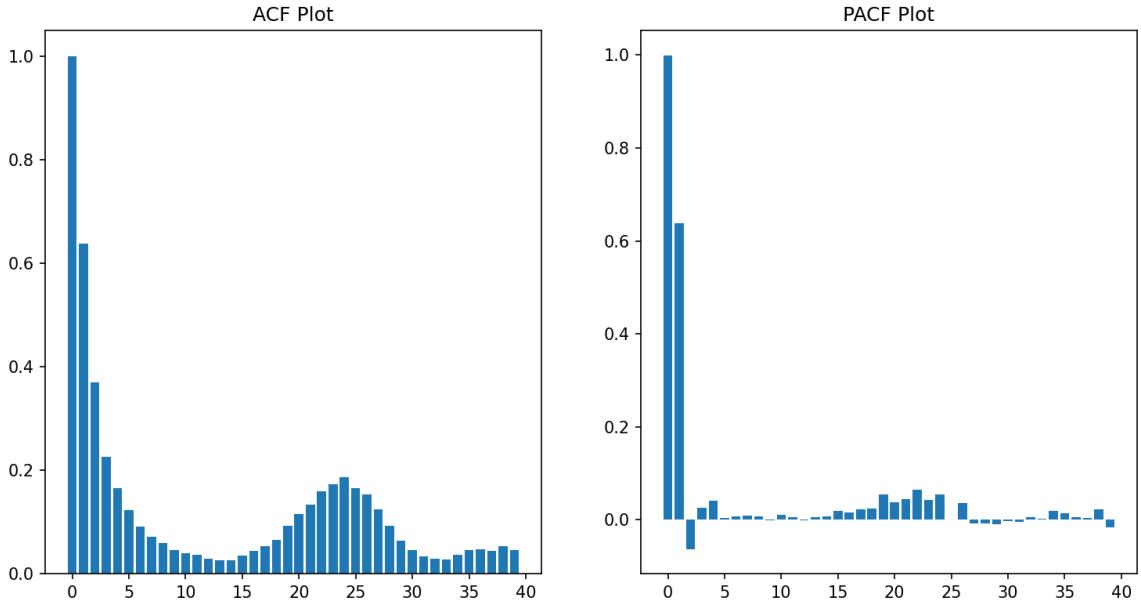
ADF Test Statistic: -11.370 p-value: 9.014e-21

The p-value was very small (9.01e-21), so the data was already stationary.

ACF and PACF Plots

ACF (Auto-Correlation Function): Measures how the data is correlated with its past values.

PACF (Partial Auto-Correlation Function): Measures the correlation of the data with its past values, removing the influence of earlier lags.



Based on visual inspection, the parameters were chosen for the ARIMA model as:

p=1 (one significant spike in PACF). Count how many bars (lags) are significant before they drop to zero.

q=1(one significant spike in ACF). Count how many bars (lags) are significant before they drop to zero.

Tit = total number of iterations

Tnf = total number of function evaluations

Tnint = total number of segments explored during Cauchy searches

Skip = number of BFGS updates skipped

Nact = number of active bounds at final generalized Cauchy point

Projg = norm of the final projected gradient

F = final function value

N	Tit	Tnf	Tnint	Skip	Nact	Projg	F
6	31	36	1	0	0	9.333D-06	1.242D+00

F = 1.24168666533326

Trained a SARIMA model with parameters (1, 1, 1)

(autoregressive term, differencing, moving average term).

The model found the best parameters for the SARIMA process and stopped:

SARIMAX Model MAE with *temp*, *dwpt*, and *price*: 0.4917

SARIMA relies solely on the past behavior of the demand variable.

SARIMAX benefits from additional information about weather conditions (*temp*, *dwpt*) and energy costs (*price*), which affect energy demand.

Performance Comparison

SARIMA MAE: 0.4996

SARIMAX MAE: 0.4917

SARIMAX performs better with external factors (*temp*, *dwpt*, *price)* affecting demand.

3. Forecasting

Load Train and Test Data from train.csv and test.csv.

Preprocess the Data: Handle missing values and scale the exogenous variables.

snow - values are filled with 0.

demand - values are filled with the median value of the column.

Train Models:

SARIMA: Trained only on demand values from train.csv.

SARIMAX: Trained on demand with *temp, dwpt*, and *price* as additional inputs.

Rolling Forecast:

Perform out-of-sample forecasts for each day in the test set:

Use the trained models to predict demand for the next 24 hours.

Incorporate actual data day-by-day to simulate a rolling forecast.

Evaluate Performance:

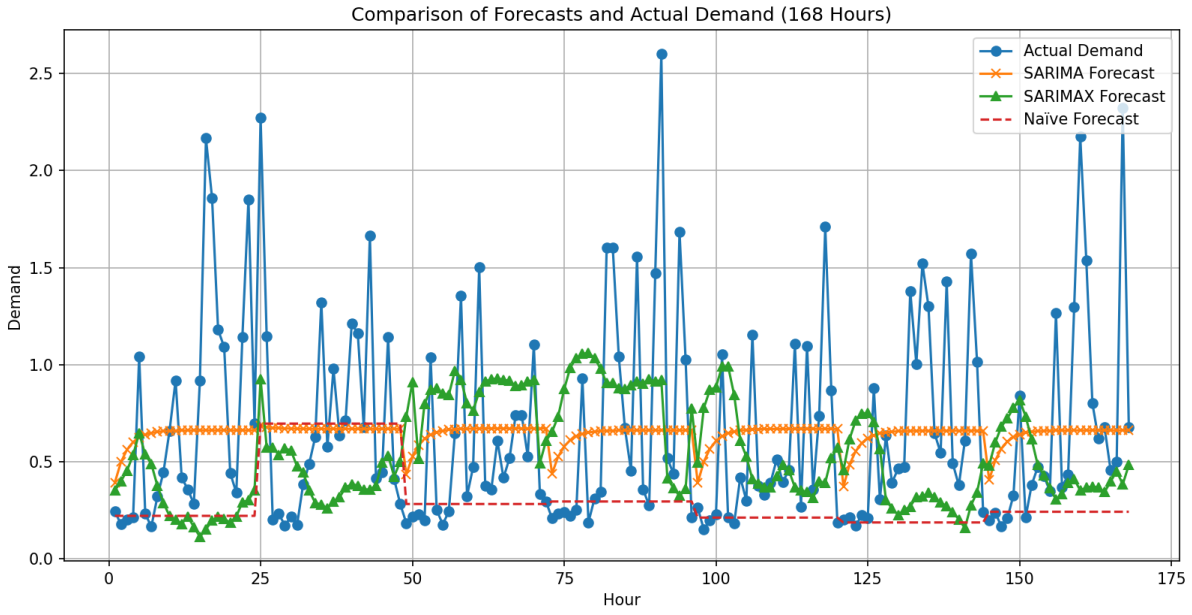
Compare forecasts from both models using metrics like MAE.

Compare with a Simple Method:

Use a Naïve Forecast (e.g., predicting the last known value for the next 24 hours).

N	Tit	Tnf	Tnint	Skip	Nact	Projg	F
6	31	36	1	0	0	4.007D-06	1.236D+00

F = 1.236



Performance Comparison

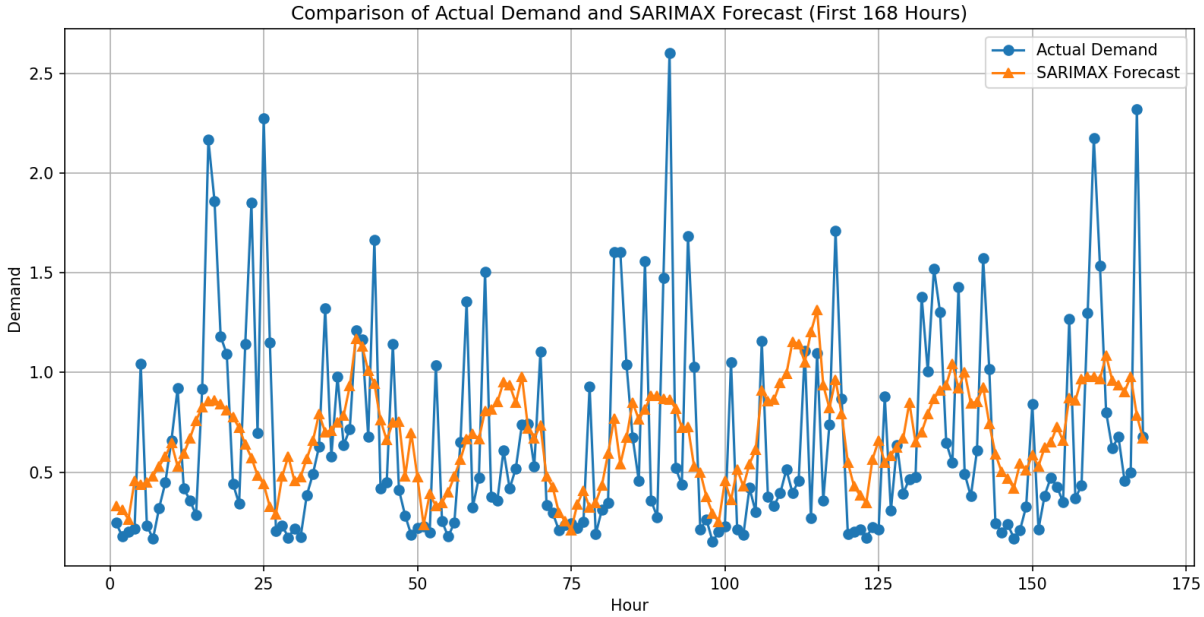
SARIMA Model MAE: 0.395

SARIMAX Model MAE: 0.485

Naïve Forecast MAE: 0.435

2nd Iteration. Feature updates. Time-Based Features.

Original	Time-Based	Cyclical Encoding
temp	*hour*	*hour_sin*
dwpt	*day_of_week*	*hour_cos*
price	*is_night*	*day_of_week_sin*
demand	*prev_day_same_hour_demand*	*day_of_week_cos*



N	Tit	Tnf	Tnint	Skip	Nact	Projg	F
12	50	54	1	0	0	1.344D-03	6.249D-01

F = 0.625

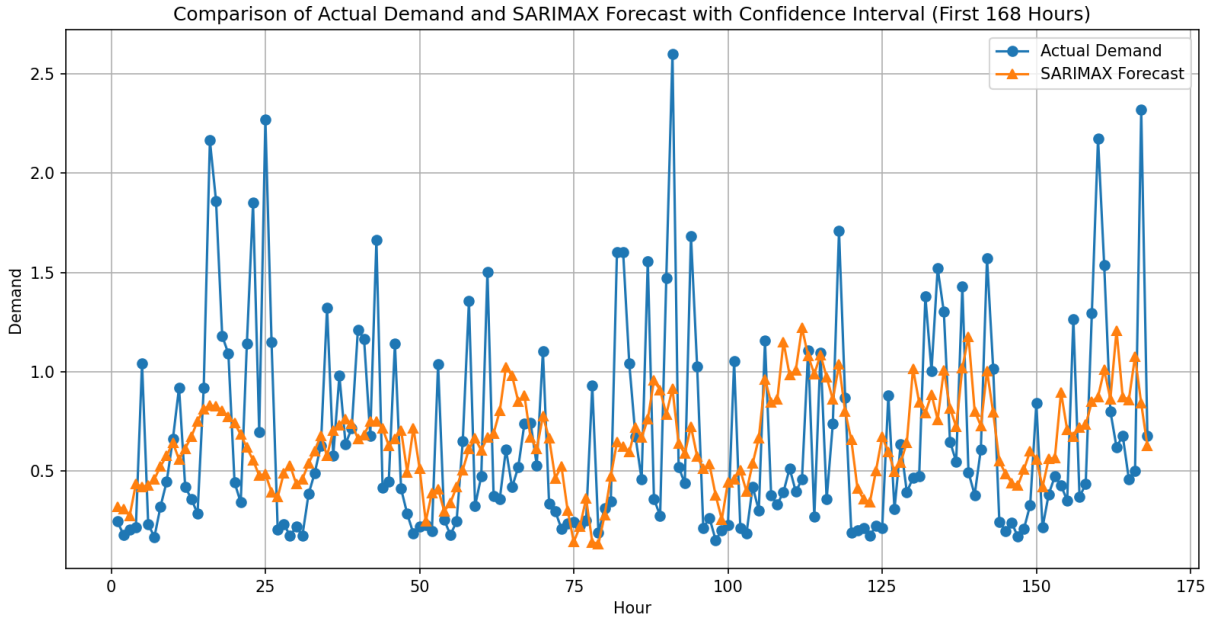
Performance Comparison

SARIMA Model MAE: 0.413

SARIMAX Model MAE: 0.375

3d Iteration. Feature updates. Lag Features.

Original	Time-Based	Lag	Cyclical Encoding	Interaction Features
temp	*hour*	*prev_day_same_hour_demand*	*hour_sin*	*temp_prev_day_same_hour_demand*
dwpt	*day_of_week*	*demand_lag_24*	*hour_cos*	
price	*is_night*	*demand_lag_48*	*day_of_week_sin*	*dwpt_prev_day_same_hour_demand*
		demand_lag_72	*day_of_week_cos*	
		rolling_mean_last_week_same_hour		



N	Tit	Tnf	Tnint	Skip	Nact	Projg	F
12	50	54	1	0	0	1.344D-03	6.249D-01

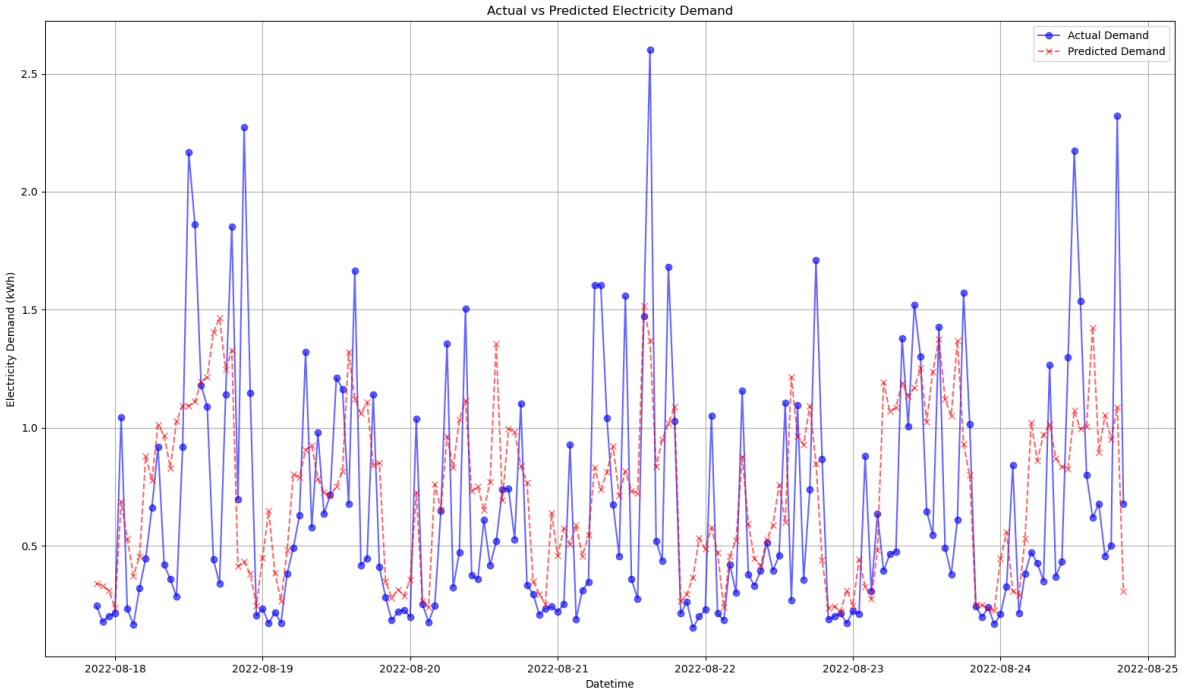
F = 0.625

Performance Comparison

SARIMA Model MAE: 0.413

SARIMAX Model MAE: 0.375

4d Iteration. Random Forest Regressors.



The 'cooling_degree_hours' feature been added. It represents the amount by which the temperature exceeds a defined threshold (15°C). This metric helps quantify the cooling demand, indicating the need for air conditioning when temperatures are higher than comfortable levels.

```
Feature Importance:
                 Feature  Importance
2                  price    0.256883
1                   dwpt    0.174813
0                   temp    0.147468
7  heating_degree_hours    0.127864
4               hour_cos    0.099987
3               hour_sin    0.075183
5       day_of_week_sin    0.064430
6       day_of_week_cos    0.042491
8  cooling_degree_hours    0.010880
```

Performance Comparison

Train MAE: 0.1891, Train R2: 0.9064

Test MAE: 0.3504, Test R2: 0.2262

Summary

The SARIMAX model achieved the best forecasting performance with an MAE of 0.375. This model was able to effectively forecast the average load profile, leveraging the influence of temperature, dew point, and price on energy demand.

There is potential for further improvement by exploring additional features or trying more advanced machine learning models like Random Forest Regressors. Incorporating cyclical features, such as hour-of-day and day-of-week, could also enhance model accuracy.

LLM usage in this work

ChatGPT has been used for Python code generating, comments, and refactoring. Also, for rephrasing some parts of the text and explanation of some specific details in a simple way for learning purposes.<