Short-Term Electricity Demand Forecasting for the ERCOT zone

Name: Peter Barengo

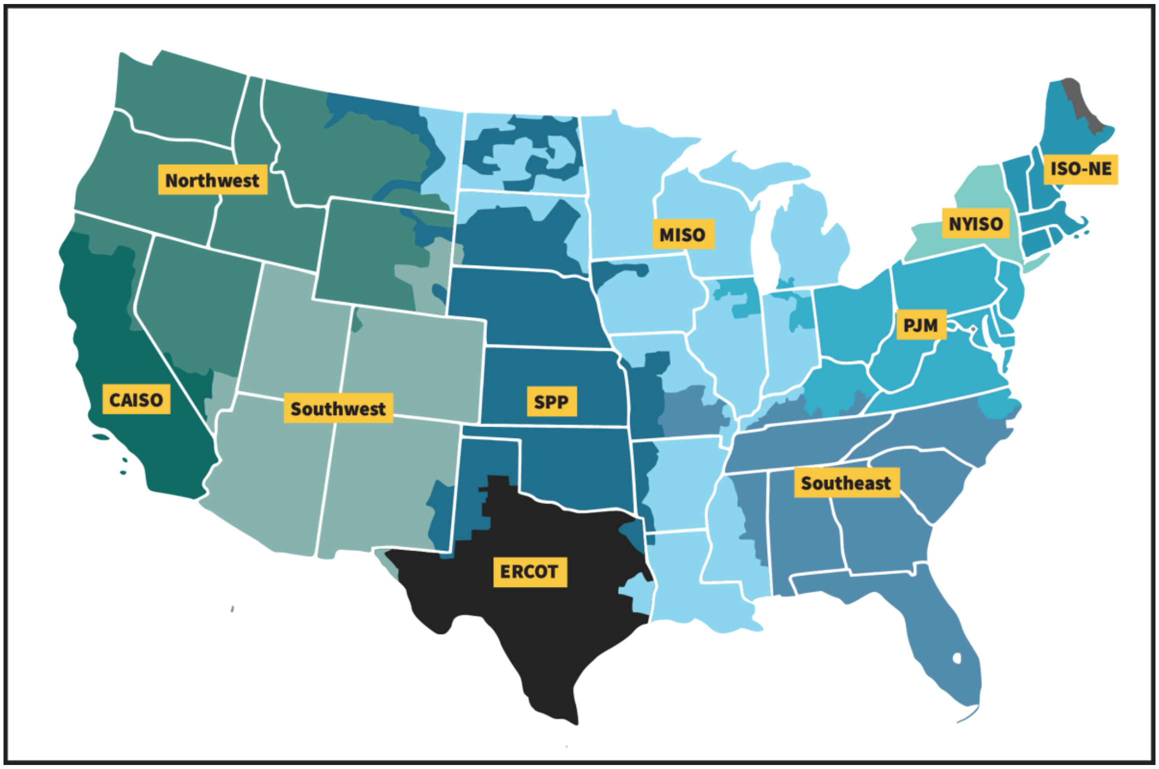
Date: 07/20/2025

Abstract

The objective of this project is to explore the process of building a machine learning model for short-term electricity demand forecasting in the ERCOT region, using XGBoost Regressor as the primary modeling approach. This project was motivated by my interest in power markets and their strong seasonal patterns and relation to the weather, which make them well-suited for machine learning applications. By using lagged demand variables, weather data from across Texas, and various time features, I developed an ERCOT demand forecasting model that achieved an MAPE of 0.93% on the test set for a 1-hour forecast. Accurate short-term demand forecasting is valuable for improving power scheduling, managing grid reliability, and supporting real-time and day-ahead energy pricing.

Introduction

The Electric Reliability Council of Texas [ERCOT] is a major ISO and manages the flow of electric power to over 26 million Texas customers. ERCOT is just one of many large ISOs in the US, others include PJM, MISO, and CASIO.



Accurate electricity demand forecasting is essential for maintaining grid reliability, balancing supply and demand, and reducing operational costs. Inaccurate forecasts can lead to inefficient energy generation, price volatility, or even blackouts.

This project aims to develop a predictive model for ERCOT electricity demand using historical data and relevant features, with the goal of producing accurate short-term forecasts. These forecasts can support better energy management decisions and potentially be used in the future to help inform electricity pricing.

Data

This project uses two main data sources, historical electricity demand data from ERCOT and weather data from Meteostat. The ERCOT dataset covers the period from 2017 to the present and provides hourly system-wide electricity demand [1]. The weather data is sourced from a weather station located in Central Texas, offering hourly measurements of temperature, humidity, wind speed, and other relevant variables.[2]

Feature Engineering

To improve the model's predictive accuracy, I engineered a set of features that captures weather patterns, seasonality, and demand history. The following categories summarize the features:

- Time Features
    - Hour of day[3], day of week[4], month[5], and season were extracted from the datetime index.
    - Weekend and school day features were included to capture structural demand patterns tied to human activity.
    - All time-related variables were hot encoded.
- Lag Features
    - Lagged values of demnad and weather were added at 1, 2, 3, 22, 23, and 24-hour intervals for the following. These lagged intervals were chosen to capture immediate and daily seasonality effects[6].
        - ERCOT demand
        - Temperature
        - Dew point
        - Perceived temperature
- Rolling Statistics
    - Rolling means and standard deviations were computed over 6h, 12h, and 24h windows for the following. These helped capture local trends and volatility in both demand and weather signals.
        - ERCOT demand
        - Temperature
        - Dew point
- Weather Features
    - Raw hourly temperature and dew point were included, converted to Fahrenheit.[7]
    - “Feels like” temperature was estimated as the average of temp and dew point.
    - Heating Degree Days and Cooling Degree Days were calculated using a 65°F base to quantify seasonal heating/cooling demands.
    - Squared terms were added to model weather-related nonlinearity.

[1] Figure A1 in the Appendix
[2] Figure A2 in the Appendix
[3] Figure A3 in the Appendix
[4] Figure A4 in the Appendix
[5] Figure A5 in the Appendix
[6] Figure A6 in the Appendix
[7] Figure A7 in the Appendix

Modeling Approach

For this project, I chose XGBoost Regressor as the model of choice for short-term demand forecasting. XGBoost is a gradient-boosted decision tree and it is particularly effective when working with engineered features, making it well-suited for this forecasting task involving seasonal, categorical, and weather-based variables.

The dataset was split to reflect real-world forecasting, where future data should not leak into the past. I used the most recent 19 months (from 2024 to mid-2025 present day) as the test set and used the remaining historical data (2017 to end of 2023) for training. This train-test split avoids data leakage and ensures that the model generalizes well on unseen data.[8]

To prevent overfitting and monitor model performance during training, I used early stopping with a validation set, as well as using a low n_estimator value. Specifically, the model was trained with these parameters:[9]

- n estimators= 50
- learning rate= 0.1
- early stopping rounds= 5

Although full grid search or k-fold cross-validation was not implemented, the model parameters were manually tuned with the primary goal of decreasing the chance of overfitting, while not sacrificing too much performance.

While a formal pipeline was not used, the workflow followed a clear path:

1. Data loading and cleaning
2. Train-test splitting
3. Feature engineering
4. Model training and evaluation

This setup makes the workflow easy to test, debug, and improve upon in the future with k-fold cross-validation, grid search for hyper parameter tuning, or additional models/preprocessing steps.

Results

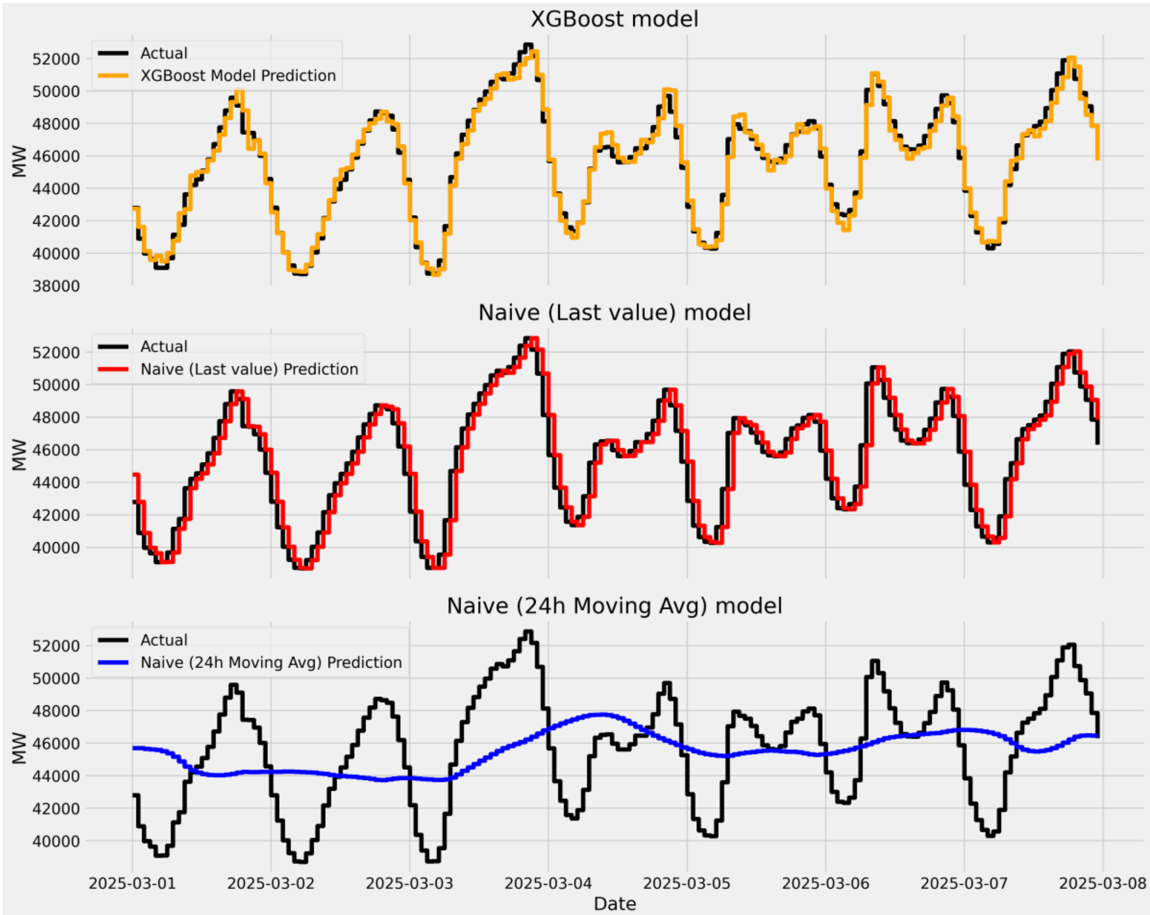
My XGBoost model was able to achieve a MAPE of 0.93% on the out of sample test split. This MAPE equates to an RMSE of 675 MW. To contextualize performance, two naïve baseline models were used. Both naïve models were significantly outperformed by the XGBoost model:

- Last value model: Uses the previous hour's demand as the prediction, this model yielded a MAPE of 2.67%
- 24-hour rolling average model: Uses the average of the past 24 hours as the prediction, this model yielded a MAPE of 9.47%

[8] Figure A8 in the Appendix
[9] Figure A9 in the Appendix

Actual vs Predicted

Here is a weeklong snapshot taken from the test set showing the models predicted demand values in comparison to the actual demand. I will also show plots of the 2 naïve models for reference.



Residuals

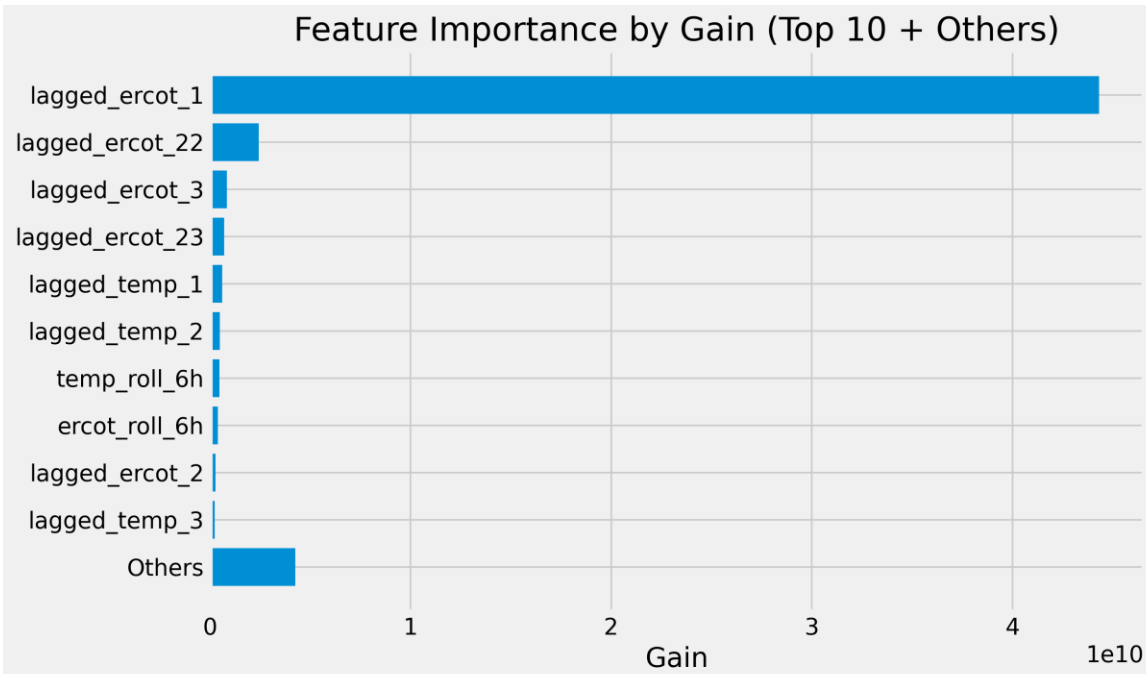
Analysis of the residuals shows that the model errors are centered around zero, with no visible patterns or skews. This suggests minimal bias and good predictive performance.[10]

However, the variance of residuals increases noticeably at higher demand levels (above 60,000 MW)[11]. This pattern aligns with the distribution of ERCOT demand, which is centered around 40,000 MW and exhibits a right skew[12]. As demand enters the upper tail of the distribution, data becomes sparser. This lack of training data in high-demand regions leads to poorer generalization by the model, resulting in increased residual variance and less reliable predictions.

[10] Figure A10 in the Appendix
[11] Figure A11 in the Appendix
[12] Figure A12 in the Appendix

Feature Importance

This gain-based feature importance plot shows that lagged demand values—especially the most recent hour—are the most influential predictors. Non-demand features contribute marginally but still provide slight performance improvements. Given that demand forecasting is highly autocorrelated, a model using only lagged target variables can serve as a baseline, achieving a MAPE of 1.03%. Compared to this baseline, the additional feature engineering contributed to an approximate 10% improvement in MAPE.



Summary of Results:

- XGBoost Model
  - 1 hour forecast 0.93% MAPE
- Naïve (Last value)
  - 1 hour forecast 2.67% MAPE
- Naïve (24h avg)
  - 1 hour forecast 9.47% MAPE

Discussion

Strengths:

The XGBoost model demonstrates high accuracy and good generalization, with minimal overfitting. It effectively captures the key patterns in the data for one-hour demand forecasting.

Weaknesses and Limitations:

Errors tend to increase during upper-tail demand levels (extreme high demand periods), indicating reduced performance in these rare conditions. Additionally, the model is specifically engineered for one-hour predictions and may not generalize well to longer forecasting horizons without further adjustments. For example, a quick test for 24-hour forecasting yielded a MAPE of approximately 4.75%. While this seems reasonable, it is comparable to a naïve model that simply uses the demand value from 24 hours earlier, indicating no real improvement. This suggests that the current modeling and feature engineering approach needs to be reconsidered for effective day-ahead forecasting.

Another limitation is that only data from a single weather station was used. Although this was somewhat useful, incorporating data from multiple weather stations and performing more localized forecasting could potentially improve accuracy. Similarly, having demand data for the different zones within ERCOT could enable more granular modeling and further enhance overall forecast performance.

Overfitting and Data Leakage:

Overfitting was carefully monitored during model parameter tuning to ensure robust performance on unseen data. Data leakage was prevented by maintaining a clean and logical data pipeline: data is loaded, split into training and testing sets, and feature engineering is performed exclusively on training and test data separately. Any features requiring temporal shifting were properly adjusted to avoid incorporating future information, ensuring that the model only uses past and present data at training time.

Patterns Found:

The model's strongest predictors are lagged demand values, especially from the most recent hour, reflecting the inherent autocorrelation in demand data. Non-demand features provide marginal gains but still contribute to improved accuracy.

9. Conclusion

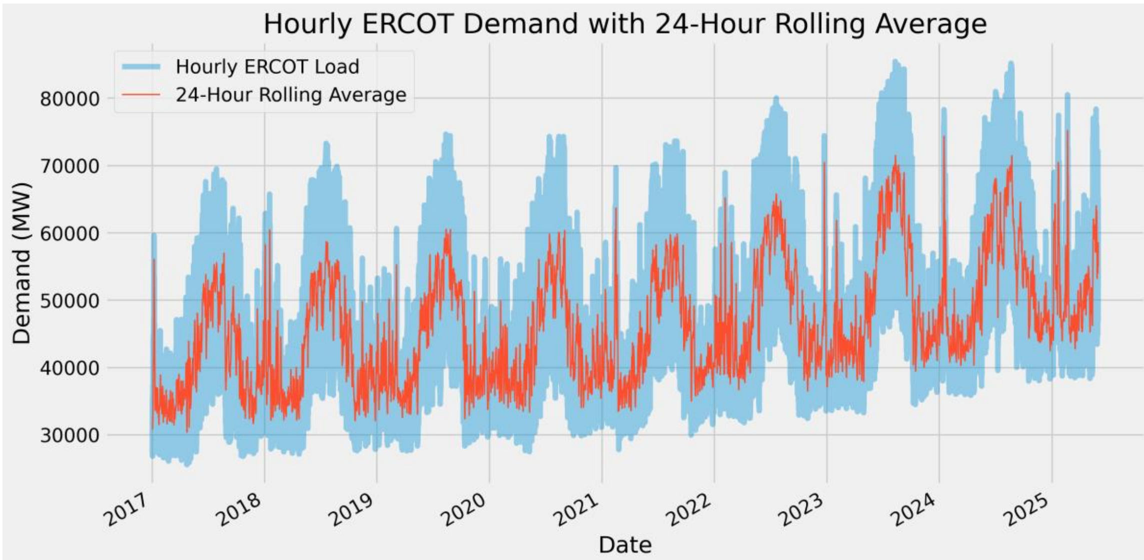
This project demonstrated how machine learning, specifically XGBoost, can be used to accurately forecast short-term electricity demand in ERCOT. By combining lagged demand, weather data, and time features, the model achieved a MAPE of 0.93% for one-hour-ahead predictions.

In the process, I gained practical experience in time series modeling, data pipeline design, and preventing data leakage. The work highlights the value of machine learning for real-world grid operations, where accurate forecasts can improve scheduling, pricing, and reliability. Future enhancements could expand geographic granularity or adapt the model for longer-term forecasts.

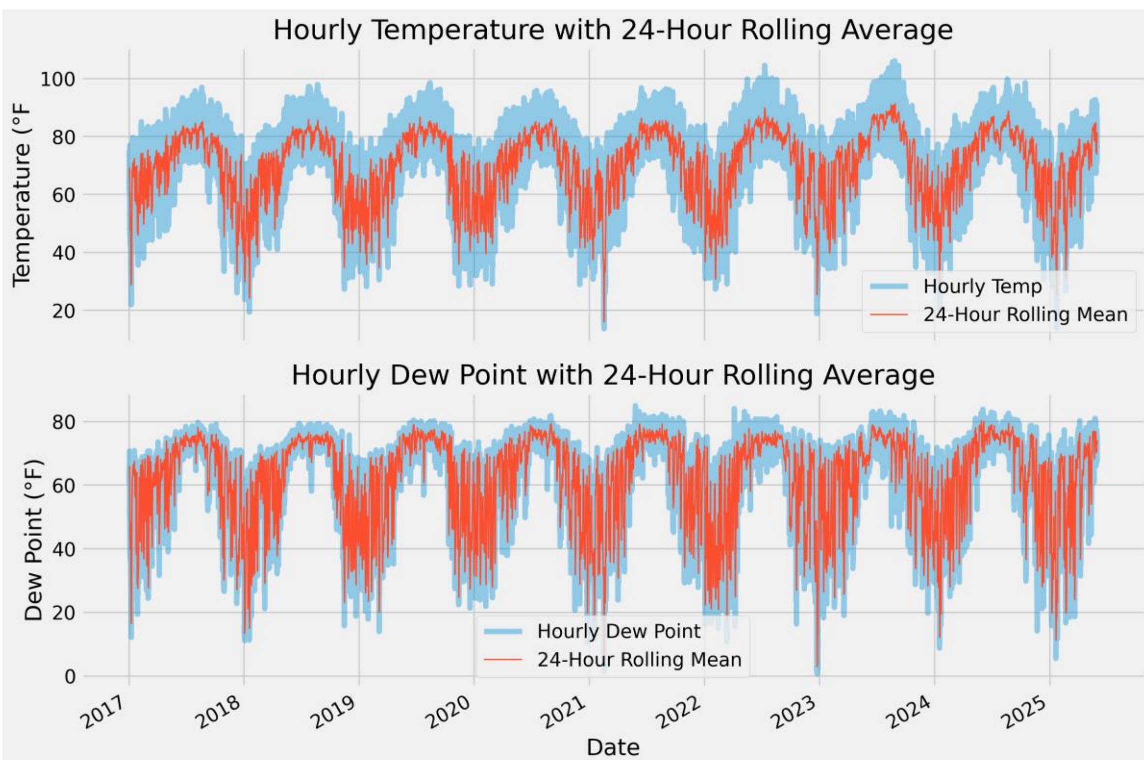
Link to model code

Appendix

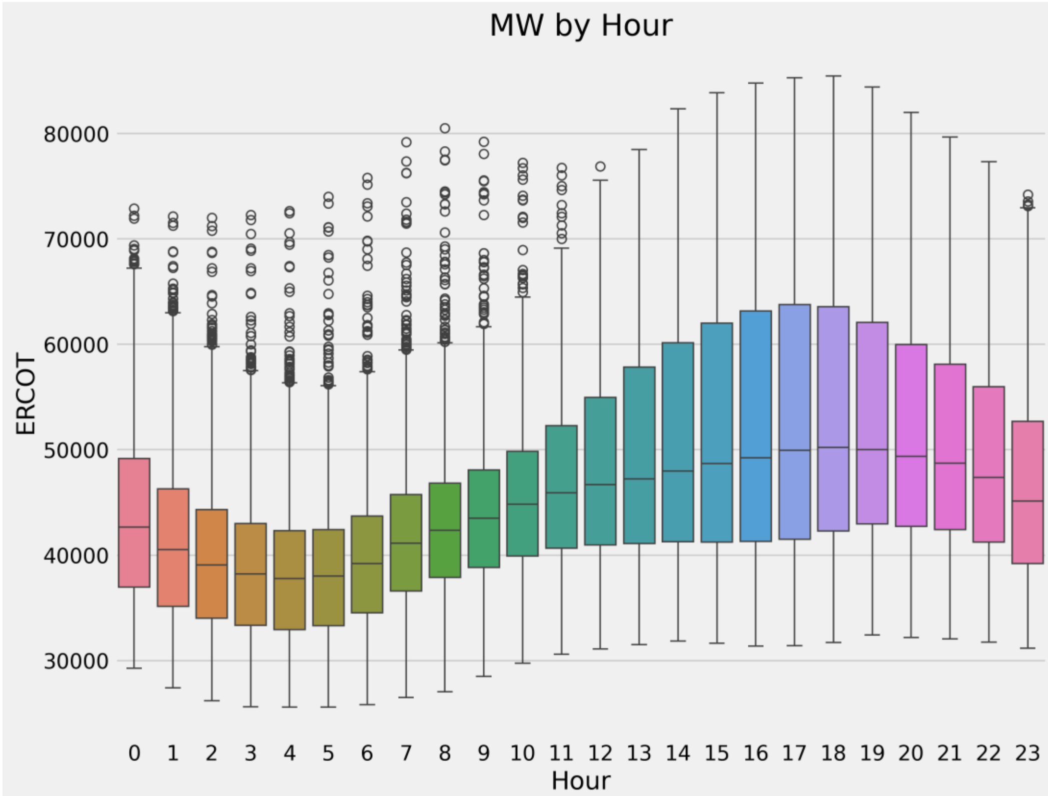
A1. ERCOT Demand



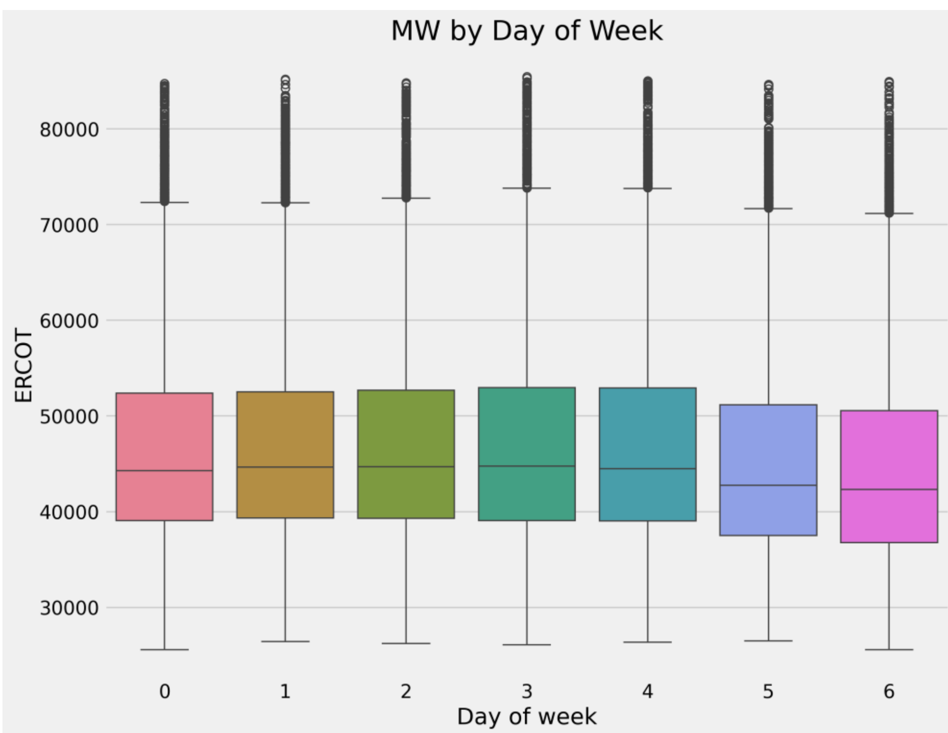
A2. Weather Data



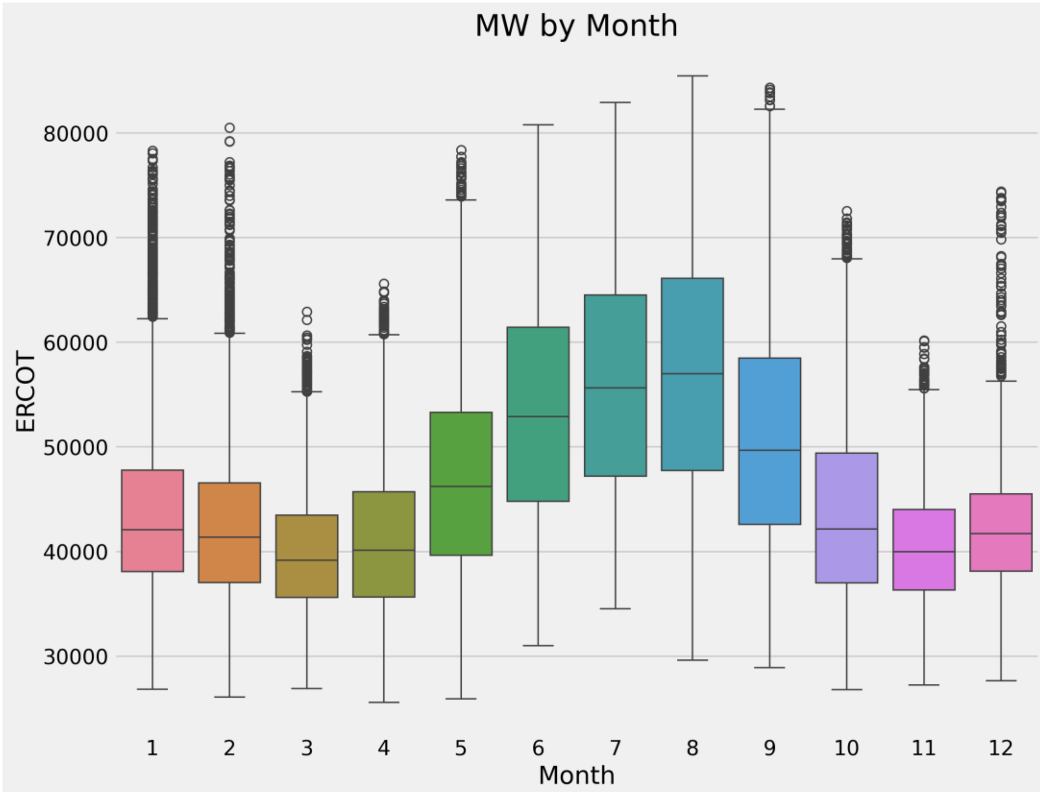
A3. Demand by hour



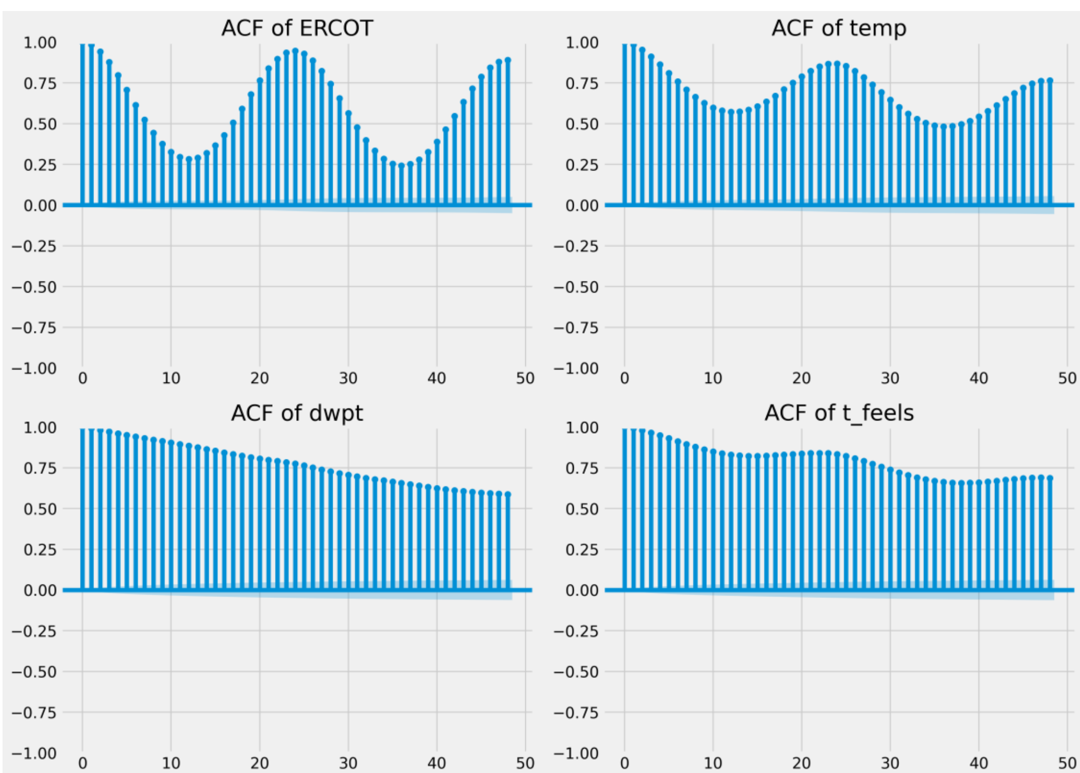
A4. Demand by day of week



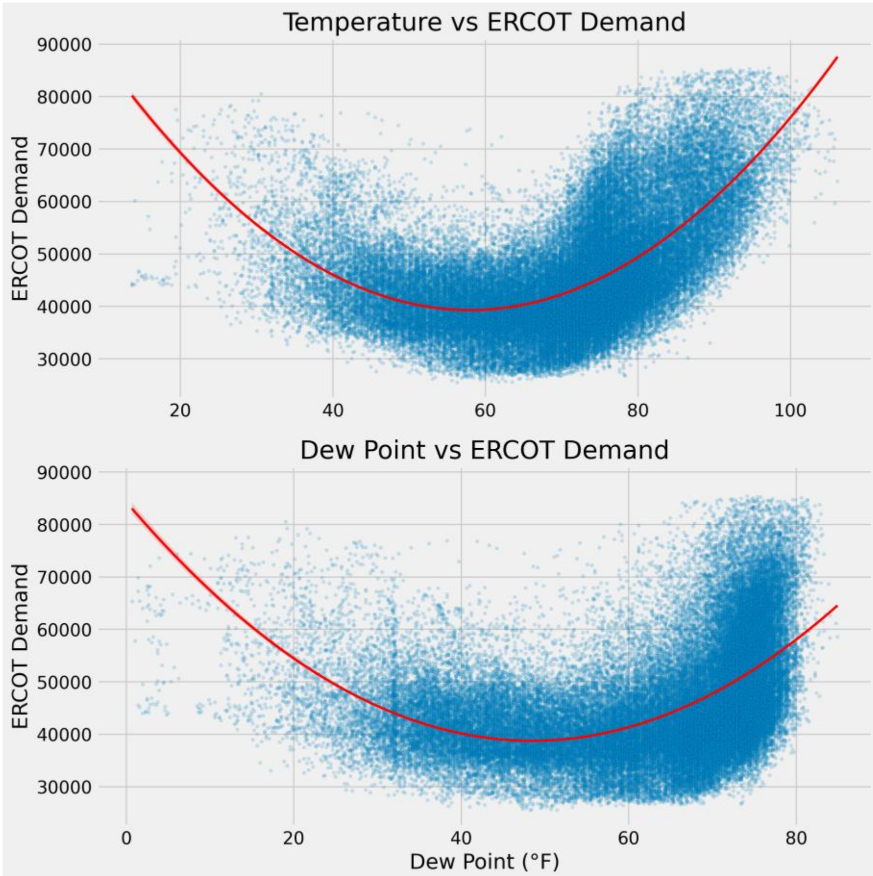
A5. Demand by month



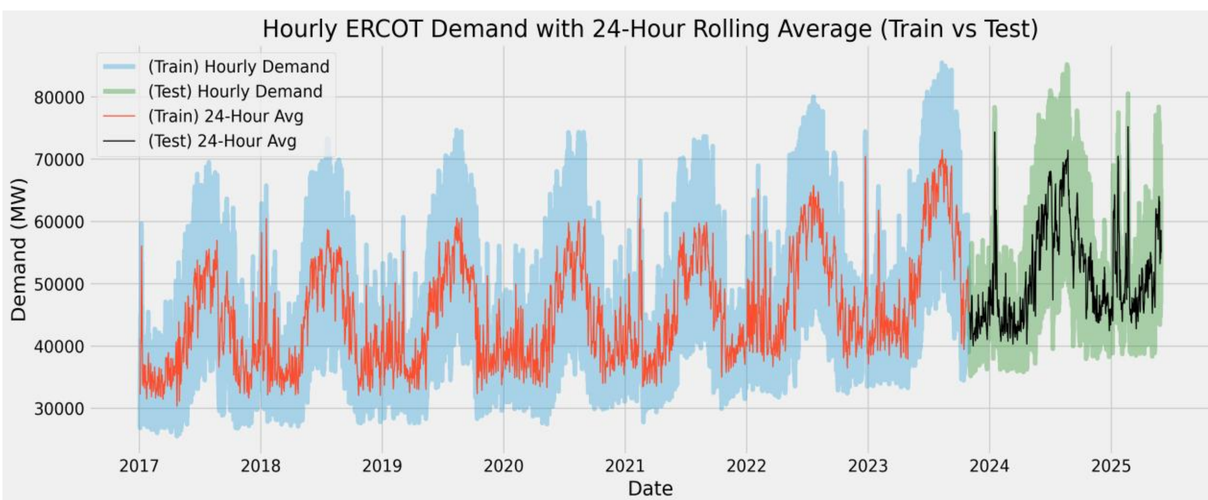
A6. ACF of demand and weather data



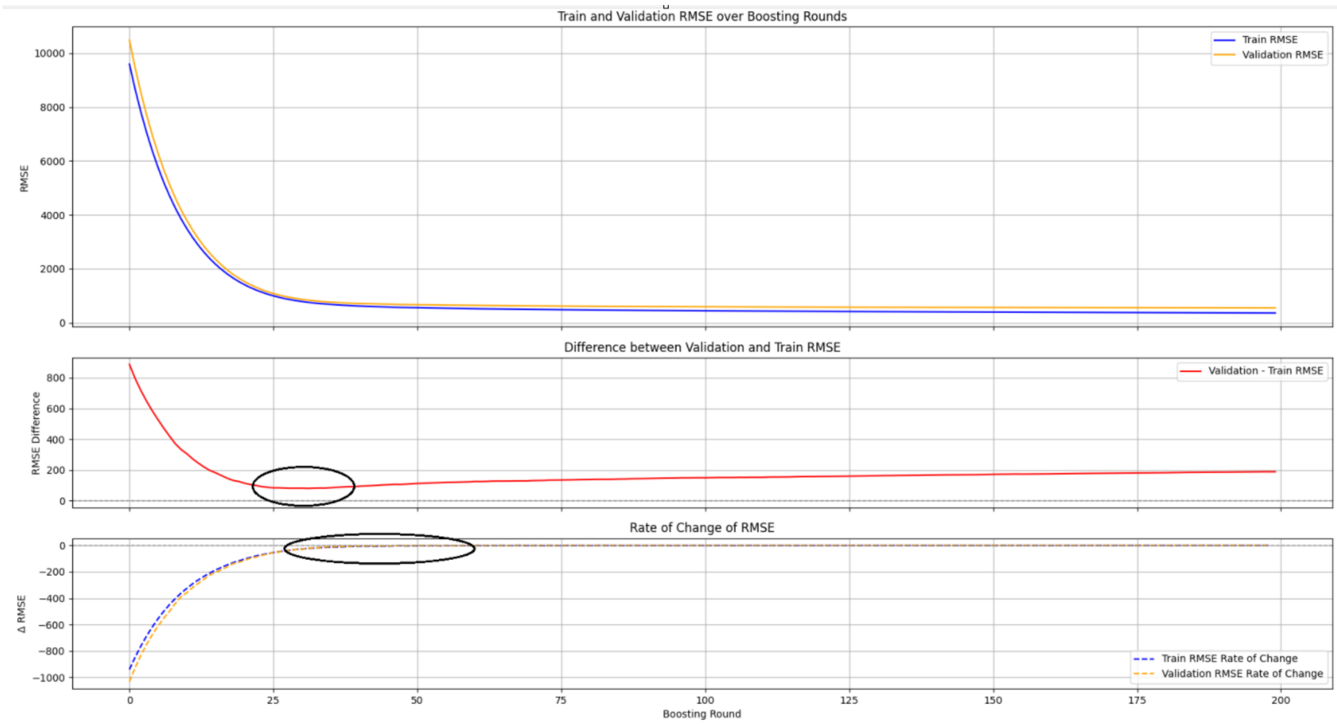
A7. Weather data in relation to demand



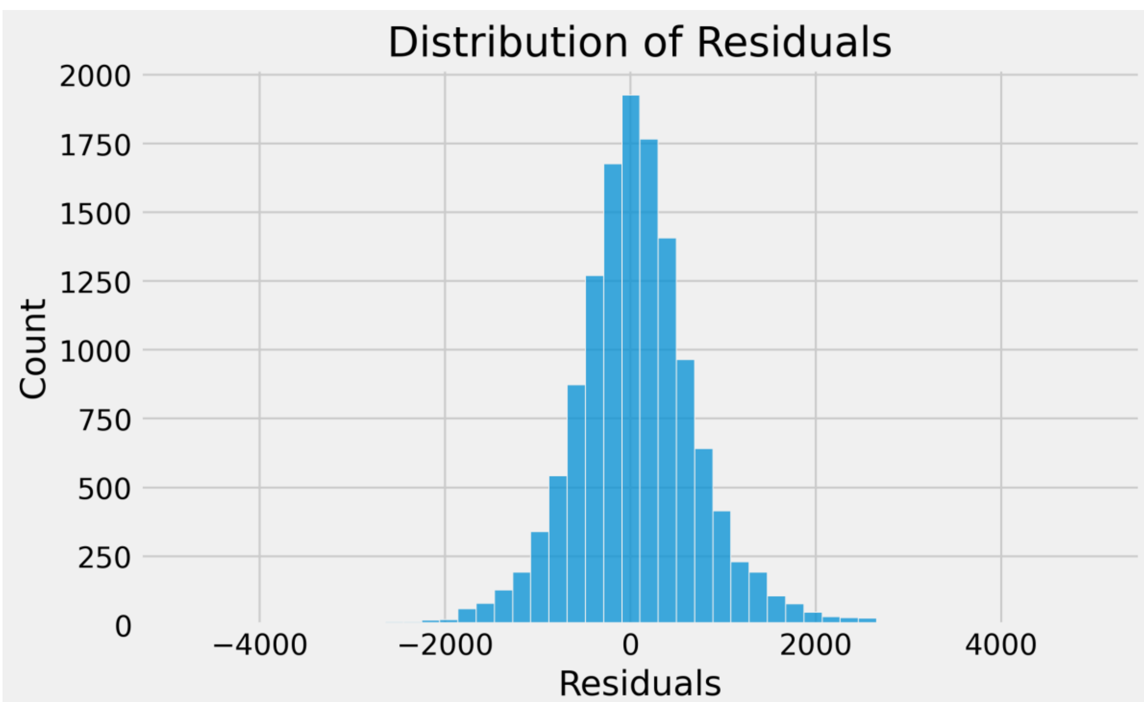
A8. Train and test split for ERCOT demand



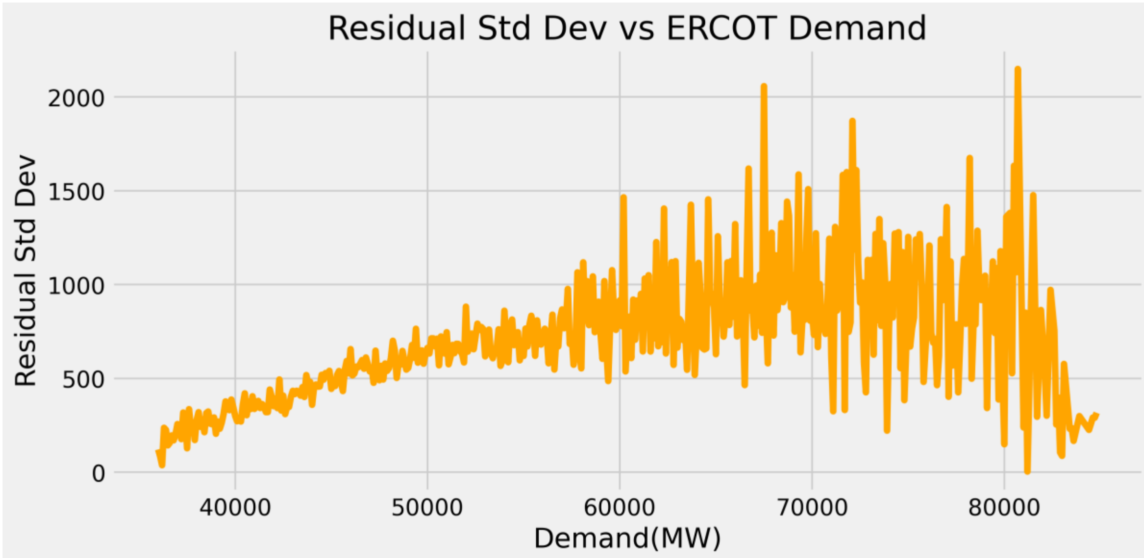
A9. XGBoost parameter tuning. (Recognizing when the model starts to overfit after 50 boosting rounds)



A10. Distribution of XGBoost model residuals



A11. Residuals std dev



A12. Distribution of ERCOT Demand