Improved Electricity Technical Documentation

Contents

Introduction & Business Problem

The Electricity Consumption Forecasting Project is designed to utilize multiple time-series analytics methods to improve the operational efficiency and strategic planning of energy suppliers. By analyzing consumption patterns, trends, and seasonal variations in electricity usage data, this project aims to predict future electricity demands accurately. This involves a thorough analysis of historical consumption data to identify patterns and trends that significantly influence energy usage levels.

The project will employ a range of advanced predictive modeling techniques, encompassing both statistical and machine learning models. The objective is to develop robust forecasting models that can reliably predict future energy demands. These models are expected to serve as essential tools for energy managers, enabling them to optimize electricity generation and distribution, thus reducing costs associated with overproduction and minimizing the risk of energy shortages.

Moreover, by integrating detailed time, date, and holiday data, this approach acknowledges the significant impact of different factors on energy consumption. The outcomes of this project are anticipated to have a great impact on the operational strategies of energy suppliers. By providing accurate demand forecasts, this project will empower decision-makers with the insights needed to make informed operational and strategic choices. These choices will span various aspects of the business, from resource allocation and pricing strategies to customer service and grid management, ultimately enhancing the overall efficiency and responsiveness of the energy sector.

Improvement Highlights

We have made several improvements based on the work from the previous group. It is listed below:

New features:

We have significantly enhanced our feature engineering to improve the granularity and accuracy of our predictive models. Our expanded feature set now includes:

- **Temporal Features**: We have introduced detailed time indicators such as year, month, day, and hour, along with the day of the week, which are essential for capturing seasonal and daily patterns in electricity usage. Additionally, a weekend binary indicator has been added to distinguish between weekdays and weekends, recognizing the different consumption patterns on these days.
- **Holiday Feature:** The "is_holiday" binary feature identifies public holidays, which typically have unique electricity usage patterns compared to regular days.
- **Moving Average:** We also added a moving average of the previous hour's electricity consumption. This feature helps in stabilizing the predictions by reducing short-term fluctuations and adding contextual relevance to the observed data.

These new features are designed to provide our models with a comprehensive view of the temporal dynamics affecting electricity consumption, enabling more accurate and reliable forecasting.

More models:

To broaden our analytical scope and leverage the strengths of various advanced algorithms, we have integrated additional predictive models into our evaluation framework. This includes:

- **Random Forest:** An ensemble method that leverages multiple decision trees to improve prediction accuracy and robustness.
- **Elastic-Net Regularization:** A linear regression model that combines L1 and L2 regularization penalties, effective in reducing overfitting and feature selection with highly correlated predictors.
- **XGBoost:** A machine learning model using gradient boosting technique.
- **Facebook Prophet:** A tool designed for forecasting time series data, especially useful in scenarios with seasonal variations.

These models were selected for their strengths in handling complex and large-scale data with seasonal variations, thus providing a comprehensive approach.

Data Extraction, Diagnostics, & Processing

Data Overview

The purpose of this section is to detail the data extraction, data diagnostics performed on data received, and the data processing for modeling.

The Electricity Load Diagrams 2011-2014 dataset contains comprehensive electricity consumption data for 370 clients from 2011 to 2014. The data is characterized as a time-series dataset with real feature types and is commonly used for regression and clustering tasks.

Data Extraction Process

The dataset was extracted from the UCI Machine Learning Repository. It is organized in a text file where data entries are separated by semicolons (;). The first column contains the date and time as strings in the format 'yyyy-mm-dd hh:mm: ss', with subsequent columns representing each client's electricity consumption in kilowatts at 15-minute intervals.

To facilitate hourly consumption analysis, a preprocessing step was designed to aggregate the quarter-hourly measurements into hourly readings. This conversion requires that the kW values be divided by 4 to reflect the quarter-hourly recording intervals, thereby converting them into kilowatt-hours (kWh).

It was claimed on the website that the data has no missing values, but to ensure correctness we also checked for missing values. Additionally, the dataset accounts for clients who were added after the year 2011 by marking their consumption as zero prior to their introduction.

Data Diagnostics & EDA

A series of quality checks are performed on the data extract provided. These checks include:

- Number of Records:

  The dataset contained the number of records as described, with 140,256 features indicating individual time-interval readings across all 370 clients.

- Duplicate Records:

  Duplicate records are allowed to exist as it's possible that the same electronic records exist at different time intervals.

- Missing Values in Relevant Fields:

  The dataset is checked to have no missing values as noted.

- Sum of a Numeric Field:

  For each client, the sums of the electricity consumption were calculated to provide an aggregate view of the data and to verify the data's integrity across the time span.

- Data Period Confirmation:

  The dataset's temporal span was confirmed, ensuring that the records ranged from the beginning of 2011 to the end of 2014. Special emphasis was placed on the Portuguese hour adjustments for daylight saving time changes to affirm that the annual transitions were accurately reflected in the data.

- Daily Trends in a Month/Week:

  Here we extracted the client MT_002 from December 2014 to visualize in a monthly perspective, as well as the December 1, 2014 to December 7, 2014, to visualize the trend from a weekly perspective.

  In the monthly trend plot (Figure 1), we observe pronounced cyclical patterns, indicative of regular daily activities. There is a noticeable increase in consumption during the morning hours, steadily rising to a peak around noon. Following this peak, there is a decline in usage, culminating in the lowest consumption levels between 1 and 3 am. This nocturnal dip corresponds with typical sleep patterns, suggesting lower electricity demand during these hours.

  The weekly trend plot (Figure 2) offers a more granulated view of the daily fluctuations. It mirrors the overarching patterns observed in the monthly trend, further emphasizing the consistency of the client's consumption behavior. The daily peaks and troughs align with the natural circadian rhythms of human activity, where electricity usage escalates during waking hours and diminishes as the night progresses.

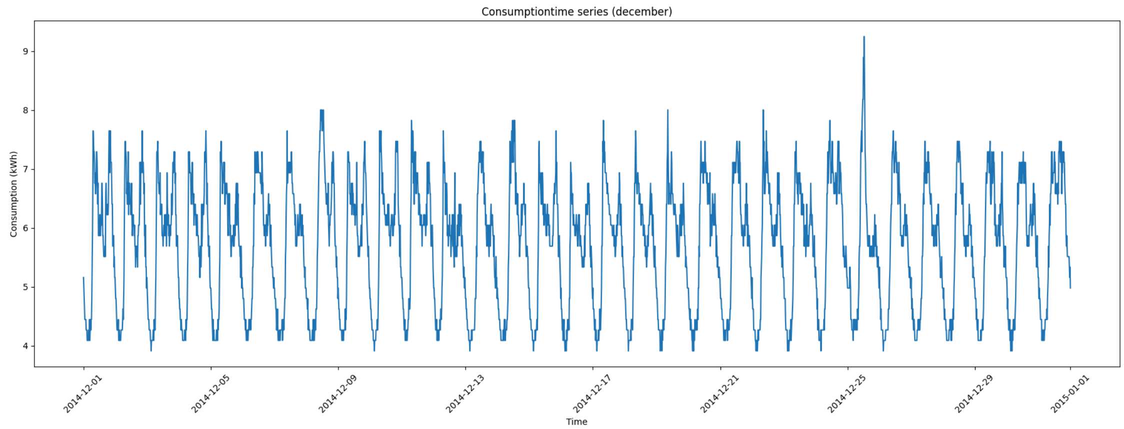

Figure 1: Consumption Trend on December 2014 for MT_002

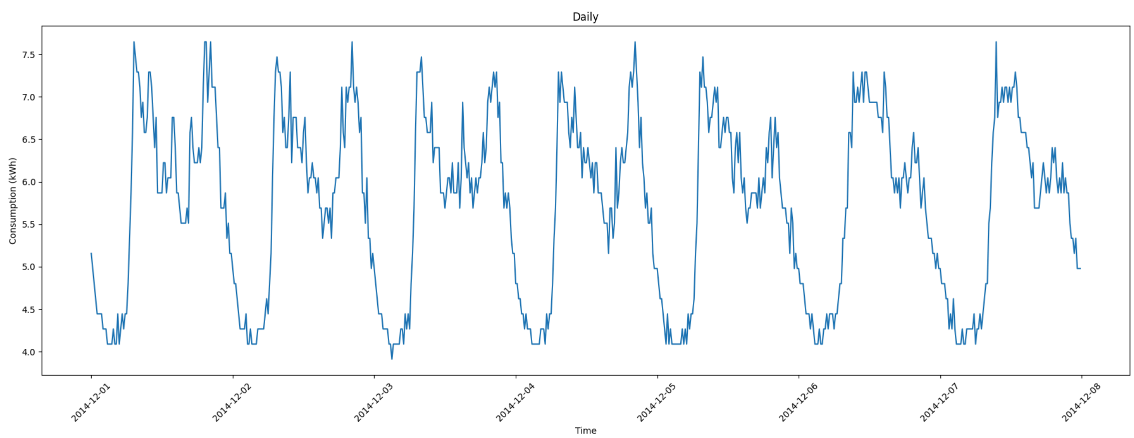

Figure 2: Consumption Trend on 1st Week of December 2014 for MT_002

- Seasonal Decomposition:

  We also conducted a seasonal decomposition on the electricity consumption to see an average monthly changing trend and daily trend (Figure 3).

  A seasonal decomposition analysis highlighted that summer months show the highest electricity consumption, likely due to increased air-conditioning use. Conversely, April presented the lowest consumption, suggesting less energy use due to milder weather and lifestyle factors. This trend is essential for understanding and managing energy demand fluctuations throughout the year.

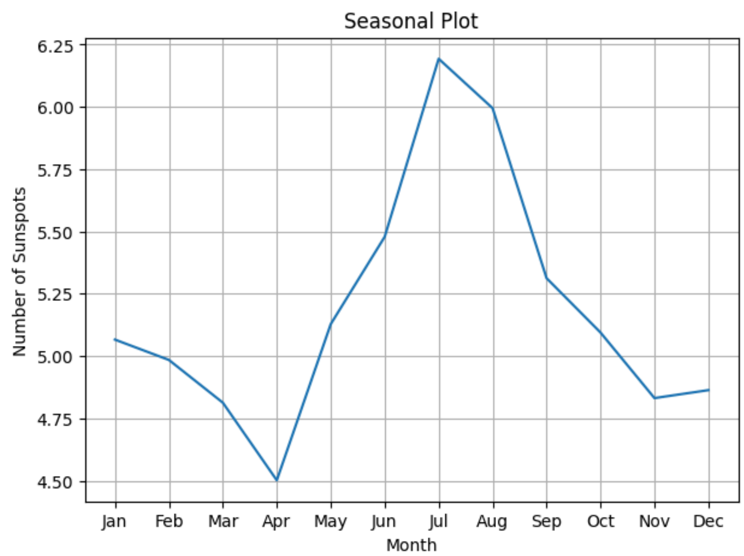

Figure 3: Seasonal Decomposition for MT_002

- Autocorrelation Analysis:

  The 144 lags seasonality (corresponding to one day) and the non-stationarity of the time series are shown in Figure 4. The autocorrelation function (ACF) and partial autocorrelation function (PACF) plots indicate daily seasonality and suggest non-stationarity in the time series data. ACF exhibits a gradual decline in correlation with increased lags, while PACF shows a significant spike at the first lag followed by fluctuations within confidence intervals. These findings indicate the use of a 96-lag autoregression model for subsequent analyses.

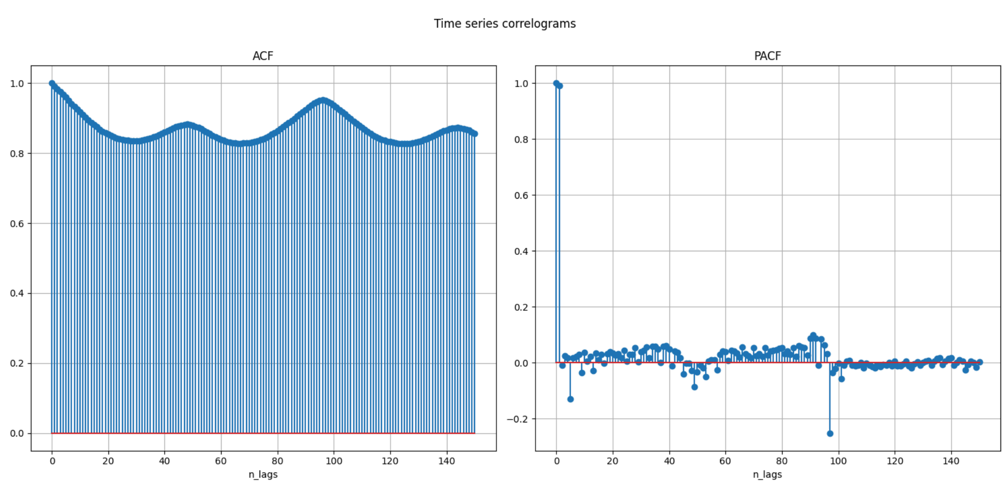

Figure 4: ACF and PACF for MT_002

- Moving Average (MA):

  Based on the ACF and PACF analysis, we apply MA(96) to the first week of December 2014, still for client MT_002. The red line represents the actual data points, indicating the variability in daily consumption. The blue line in the plot represents the moving average and demonstrates how the seasonal effects have been mitigated, offering a clearer view of the true signal within the data. Through applying the moving average filter, we can conclude that after accounting for seasonality and other cyclic components, the data is fairly homogeneously distributed, with no significant outliers or abrupt changes in consumption levels.

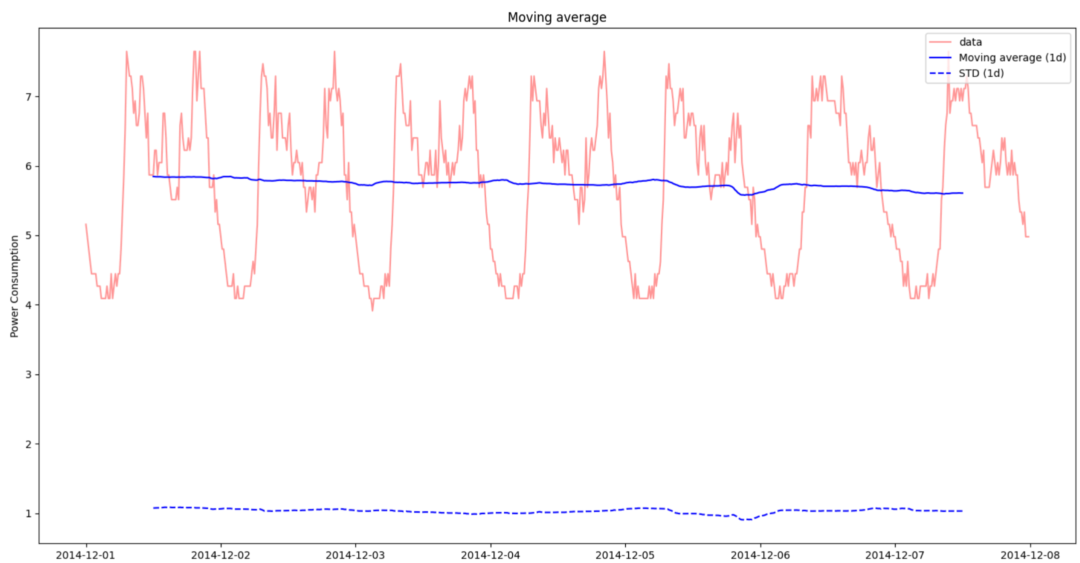

Figure 5: Moving Average(96) for MT_002

Data Processing

For all clients, the modeling data is processed using the following steps.

- kWh Conversion:

  kWh is more general to use in measuring power consumption. Thus, we divided all data points by 4 to convert the kW to kWh.

- Outliers removal:

  We used three methods including IQR, Z-Score, and Hampel package to detect the outliers and check whether they give the more reasonable results. To compare the performance, we calculated the numbers and the percentages of the outliers for each client. We also plotted the corrected data with sampled clients to obtain a more clear comparison among the three methods (shown as follows). Overall, Z-Score achieved the most reasonable performance, and we kept this to remove the outliers.

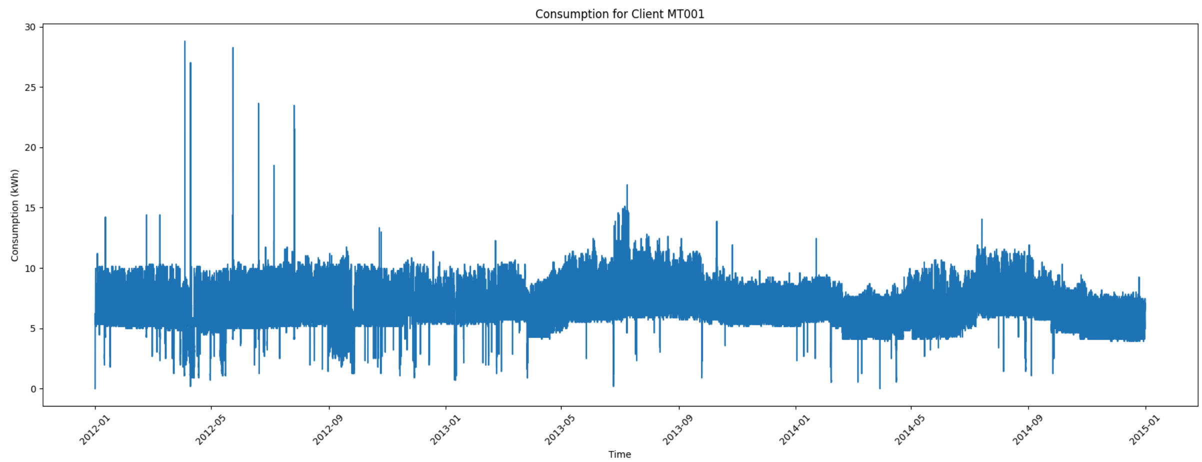

Figure 6: Original data for MT_001

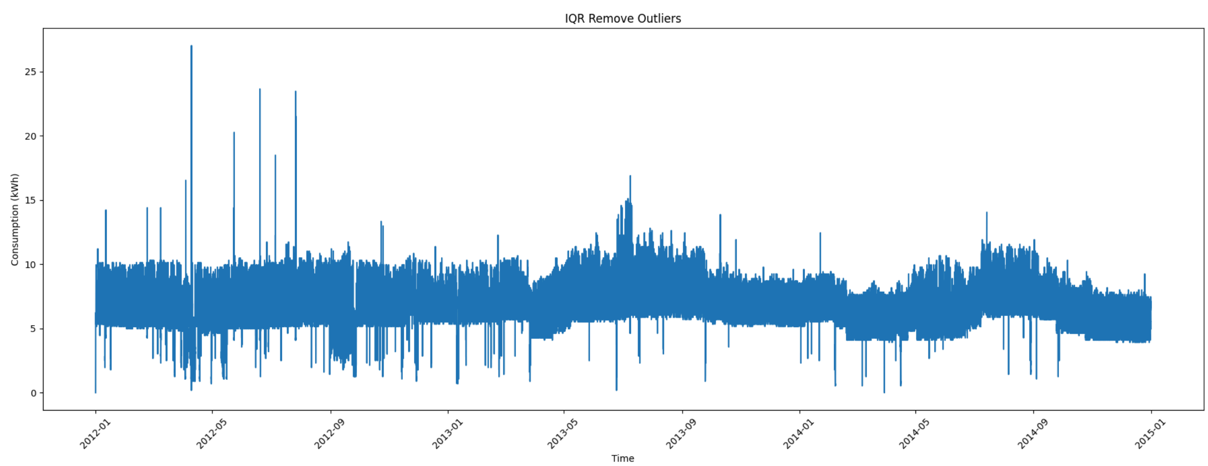

Figure 7: After removing outliers for MT_001 with the IQR method

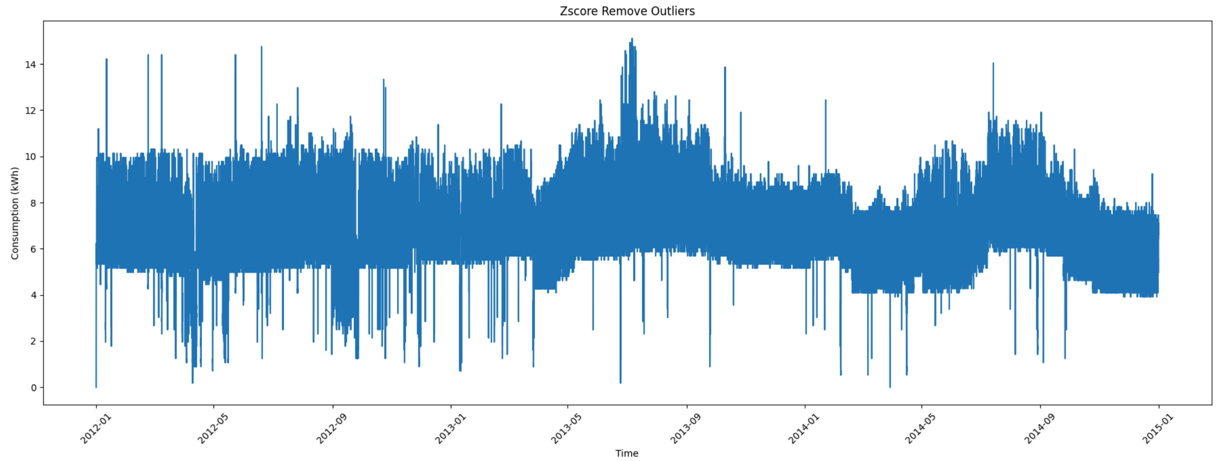

Figure 8: After removing outliers for MT_001 with the Z-Score method

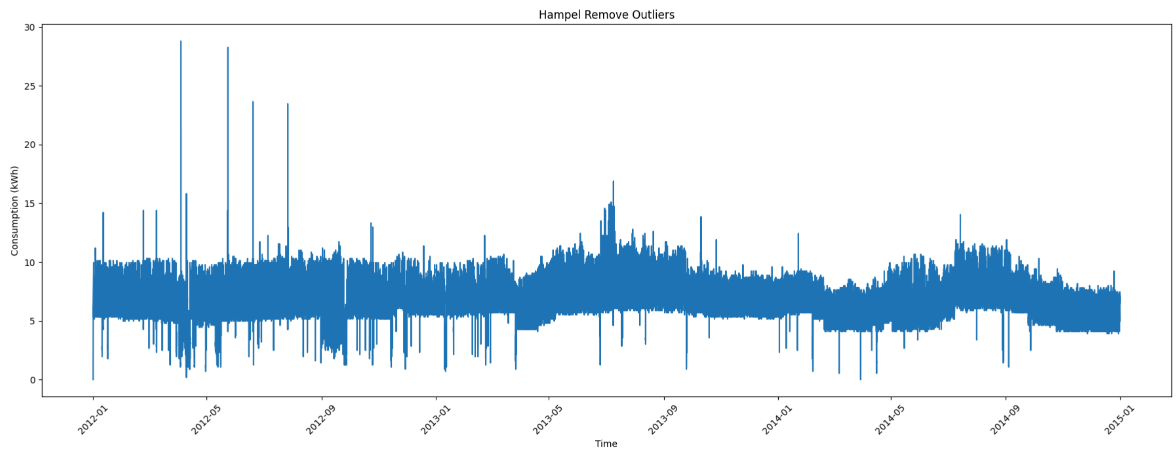

Figure 9: After removing outliers for MT_001 with the Hampel package

- Hour adjustments for time-change date:

  We are focusing on modeling the time series first, and if the model can pick up the pattern of providing 0 values in March/October for a specific date change day, then we'll not adjust anything. Otherwise, if the model is not able to pick up the pattern, then we will include a manual correction of the generated output to take into account the necessary time correction for March/October.

Target Variables

The dataset contains electricity consumption data from 370 clients recorded between 2011 and 2014. Please note that the previous group selected only the second client as the target variable when fitting models. Hence, **the target variable is MT_002.** It seems they were only interested in predicting the electricity consumption behavior of a single individual. However, we believe that a better approach would be to average the electricity consumption data of the 370 clients, utilizing the full potential of the data and providing a clearer understanding of the collective electricity consumption behavior. In this project, we will stick to their initial objective and only predict the electricity consumption behavior of the second client.

Predictive Variables

While the previous group only used past electricity consumption data from the second client to predict future usage, we enhanced the model's accuracy by incorporating additional predictive variables. These include "year," "month," "day," "hour," "day_of_week," and "weekend," all derived from the dataset's timestamps. Including these features allows the model to capture important temporal variations in electricity usage, as electricity consumption often varies significantly depending on the time of year, week, and day.

We also introduced "MT_ma_4," a time-based feature using a moving average of four periods to aggregate 15-minute measurements into hourly data. This is important because electricity consumption patterns can be more accurately predicted over longer periods like an hour.

Furthermore, we added an "is_holiday" feature to account for public holidays. Our exploratory data analysis indicated a significant increase in electricity use on holidays, such as Christmas Day (12/25/2014). Intuitively thinking, the electricity consumption is often high during holiday periods due to celebrations. This underscores the importance of adding this new feature.

Here are all the predictive features we've included:

Features	**Types**	**Descriptions**
year	Numeric	Year (2011-2015)
month	Numeric	Month (1-12)
day	Numeric	Day (1-31)
hour	Numeric	Hour (0-24)
day_of_week	Numeric	Represents the day of the week with 1 being Monday
weekend	Binary	1: Weekend; 0: Weekday
MT_ma_4	Numeric	A moving average feature of the electricity consumption over four 15-minute intervals
is_holiday	Binary	1: Holiday; 0: Non-holiday

Pre-modeling

We enhanced our feature set as outlined in the "Target Variables" section.

The dataset was divided into an 80% training set and a 20% testing set. For each modeling technique, we further split the training set into 60% for training and 20% for validation, as specific methods like Random Forest utilize time series cross-validation (TSCV) via GridSearchCV.

One mistake we would like to point out from the previous group is their misunderstanding of the requirement to divide the testing results into three or four time periods or regions of the same size, and for each region, to report MAPE and create a box plot of errors. The previous group interpreted this as using three different iterations, each iteration splitting the data differently to ensure the final test set has the same amount of data. However, we understand it as needing to split the test data into three equal parts and calculate their MAPE. Hence, we used our approach when fitting the models.

Modeling

Introduction

Upon finalizing the predictive variables through comprehensive feature engineering and preprocessing, our team embarked on the development of advanced forecasting models including Facebook Prophet, Elastic-Net Regularization, Random Forest, and XGBoost to predict electricity consumption accurately. This section delves into the methodologies, evaluation metrics, and performance comparisons between the models employed.

Previous Group's Workflow

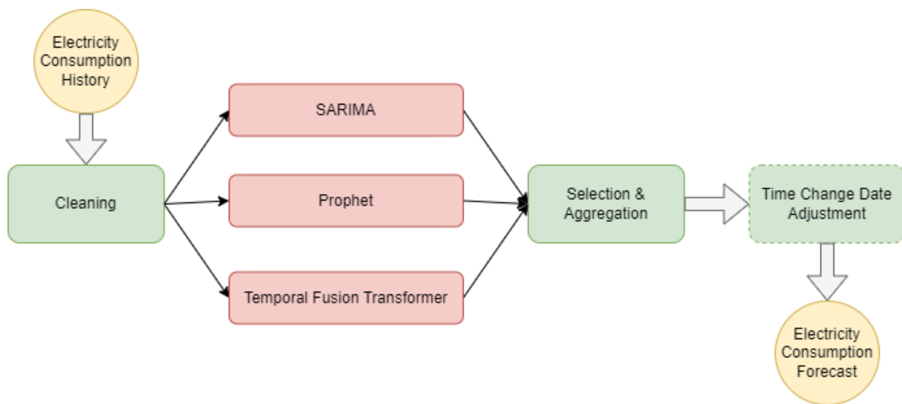

Figure 10: Previous Group's Workflow

Proposed New Workflow

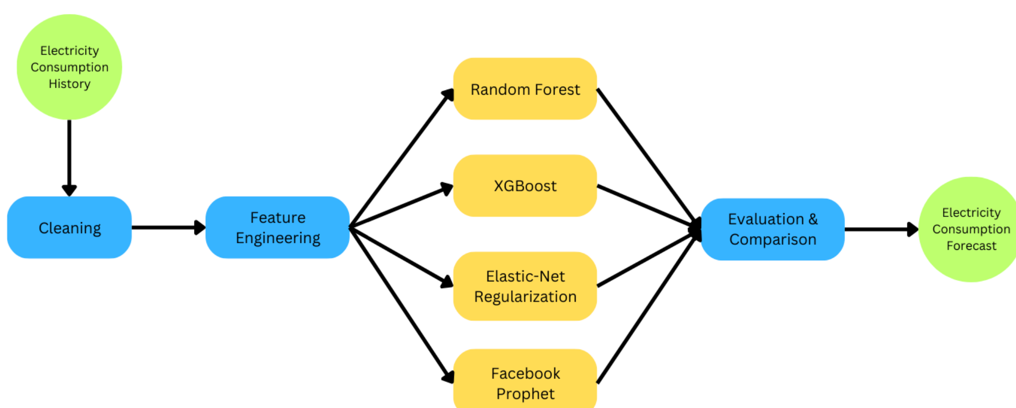

Figure 11: Updated Workflow

1. Facebook Prophet

Facebook Prophet is a powerful forecasting tool designed to handle time series data that exhibits strong seasonal patterns. Prophet is particularly adept at dealing with missing data, large outliers, and historically significant shifts in trends, such as holidays and other recurring events. It uses an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

Model Configuration

- **Growth:** Linear, assuming a steady trend that does not change direction.
- **Seasonality Mode:** Additive, which is suitable for most scenarios where seasonality does not increase with the level of the time series.
- **Yearly Seasonality:** Disabled, focusing the model on capturing daily and weekly patterns rather than annual changes.
- **Weekly Seasonality:** Enabled, allowing the model to adjust for fluctuations within the week.
- **Daily Seasonality:** Enabled, to account for variations within each day.
- **Changepoint Prior Scale:** Set to 0.05 to moderate the model's sensitivity to changes in the trend, balancing flexibility and overfitting.

Results

The Prophet model recorded a MAPE of **23.73%** on the test set, which was the highest among all the models used in this project, indicating less accuracy in forecasting electricity consumption compared to others like Random Forest and Elastic-Net Regularization.

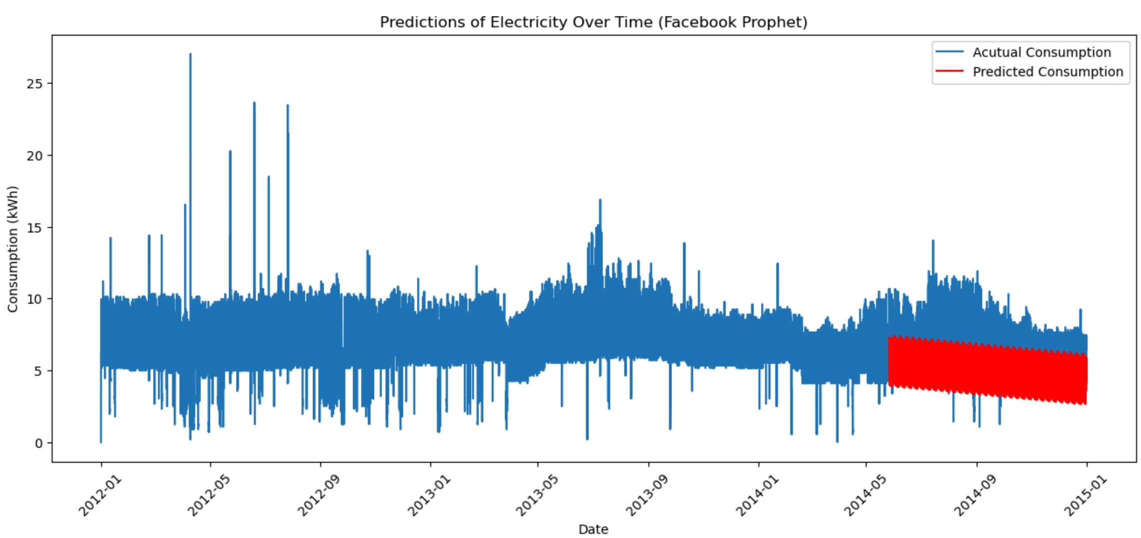

Figure 12: Predictions of Electricity Over Time (Facebook Prophet)

The model's performance (Figure 12) suggests that while Prophet is highly capable of handling seasonality and holiday effects, it may not capture other subtleties in electricity consumption data as effectively as some more finely tuned ensemble methods.

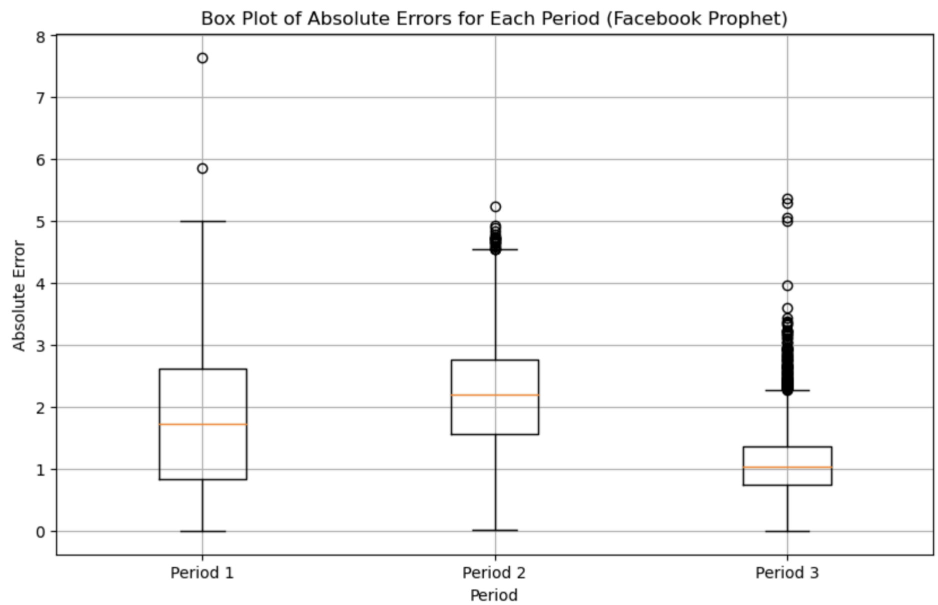

Figure 13: Box Plot of Absolute Errors for Each Period (Facebook Prophet)

The box plot of absolute errors for the three periods (Figure 13) indicates a wide range of errors in the third period, suggesting that the model's forecasting performance may deteriorate over time. This is consistent with the testing MAPE.

2. Elastic-Net Regularization

Elastic-Net is a linear regression model that combines the penalties of both L1 and L2 regularization. This hybrid regularization approach enhances the model's ability to operate under conditions of multicollinearity (where independent variables are highly correlated) and helps in feature selection by shrinking coefficients for less important variables toward zero.

Hyperparameter Tuning

The performance of the Elastic-Net model was optimized through a meticulous grid search process, focusing on two primary hyperparameters:

1. **Alpha (alpha):** This parameter represents the overall strength of the penalty, a blend of L1 and L2 regularization. We experimented with values including 0.1 and 1 to find the optimal balance that minimizes overfitting while maintaining predictive power.
2. **L1 Ratio (l1_ratio):** This parameter controls the mix between L1 and L2 regularization. Testing values of 0.1, 0.5, and 0.9 allowed us to determine the most effective ratio for promoting sparsity in the model coefficients (favoring feature selection) while controlling for multicollinearity.

Training Methodology

Elastic-Net was also trained using a TimeSeriesSplit cross-validation method, which is suitable for evaluating models on time-series data. We utilized five splits to ensure that the model was tested across a diverse set of training and validation scenarios.

Results

Through the grid search, the optimal hyperparameters for the Elastic-Net model were determined to be:

- Alpha: 1
- L1 Ratio: 0.9

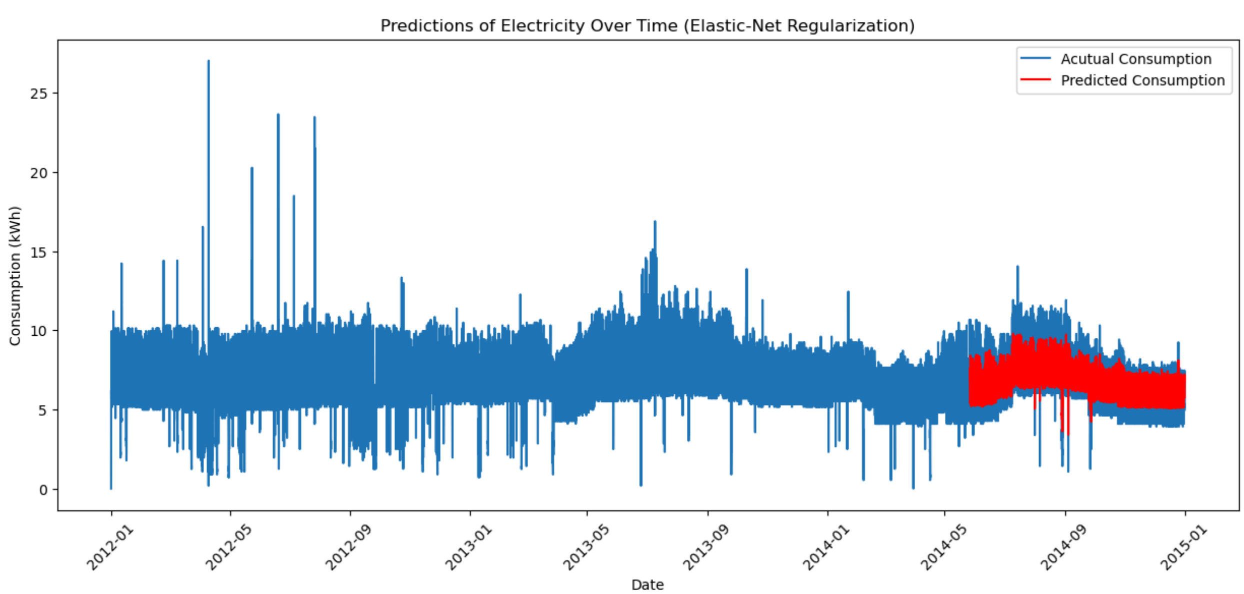

Figure 14: Predictions of Electricity Over Time (Elastic-Net Regularization)

These parameters led to a model that effectively balanced model complexity and accuracy, achieving a MAPE of **8.23%** on the test set, which is much better than the Facebook Prophet model. This performance indicates a high level of precision in forecasting electricity consumption, with the model successfully capturing the essential patterns in the data (Figure 12).

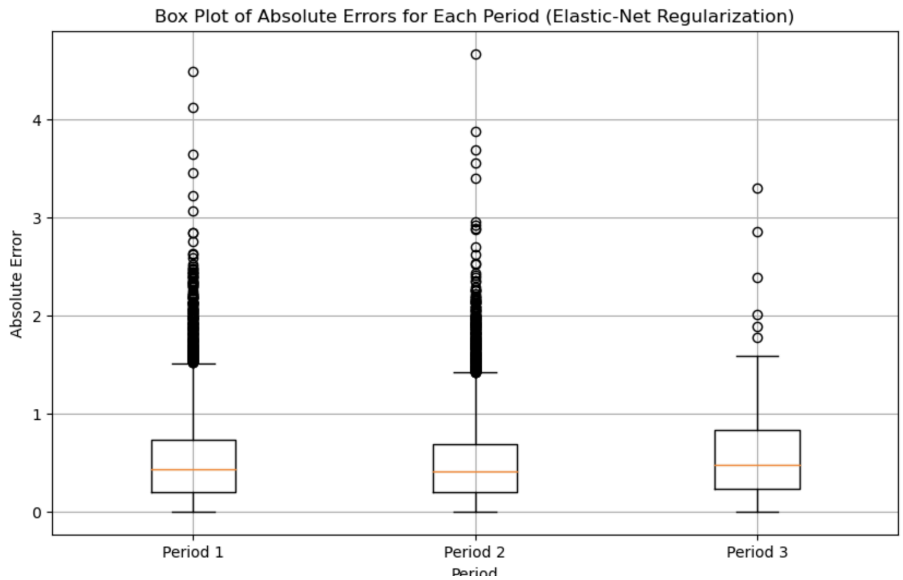

Figure 15: Box Plot of Absolute Errors for Each Period (Elastic-Net Regularization)

The box plot of absolute errors for the three periods (Figure 15) suggests that the Elastic-Net model maintains a fairly consistent level of prediction accuracy across different periods, with most of the errors concentrated at the lower end of the scale. This is consistent with the testing MAPE.

3. Random Forest

Random Forest is a robust ensemble learning method that utilizes multiple decision trees to produce a more accurate and stable prediction by averaging the results of individual trees.

Hyperparameter Tuning

To optimize the performance of the Random Forest model, several key hyperparameters were carefully adjusted through a systematic grid search approach:

1. **Number of Estimators (n_estimators):** This parameter defines the number of trees in the forest. We experimented with values of 10, 50, 100, and 300 to find the optimal balance between model accuracy and computational efficiency.
2. **Maximum Depth of the Trees (max_depth):** We tested various depths for the trees to control overfitting. The depths considered were None (allowing trees to grow until all leaves are pure), 10, 20, and 30 levels deep.

3. **Minimum Samples Split (min_samples_split):** This parameter determines the minimum number of samples required to split an internal node. Values tested included 2, 5, and 10, aiming to prevent the model from becoming too detailed and overfitting the data.

Training Methodology

The Random Forest model was trained using a TimeSeriesSplit cross-validation approach, which is suitable for time-series data. This method involves sequentially splitting the data into training and validation sets, ensuring that the validation data always comes after the training data to mimic real-world forecasting scenarios. We employed five splits to provide a robust estimate of the model's performance.

Results

The grid search identified the optimal hyperparameters for the Random Forest model as follows:

- n_estimators: 50
- max_depth: 20
- min_samples_split: 2

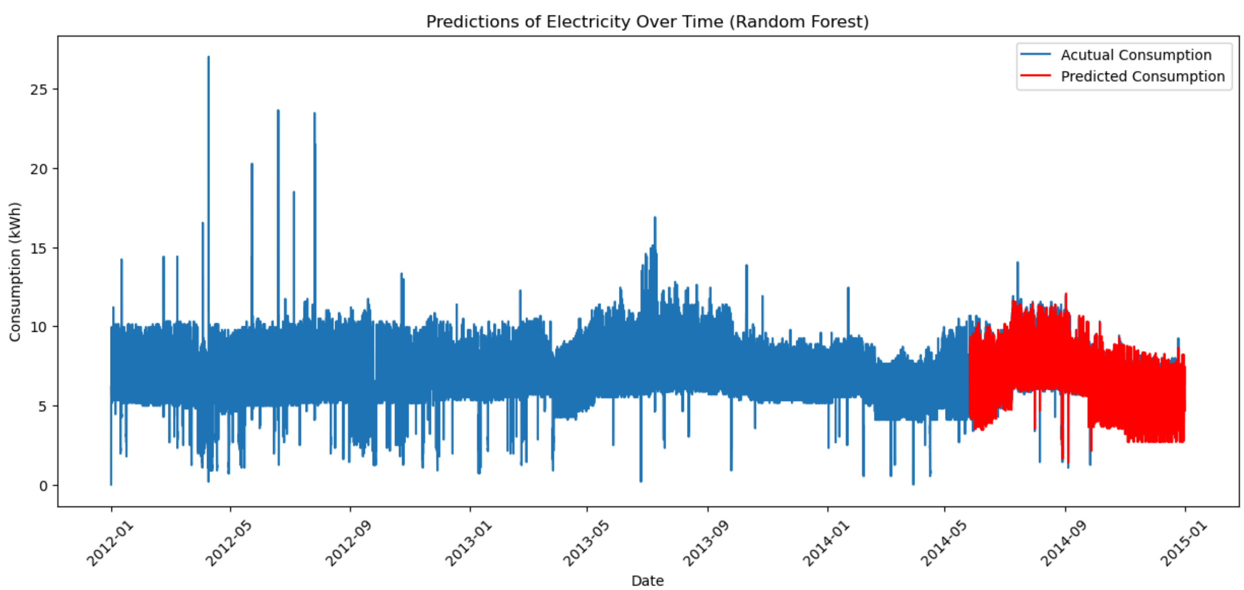

Figure 16: Predictions of Electricity Over Time (Random Forest)

With these settings, the model achieved a MAPE of **5.31%** on the test set, indicating a high level of accuracy in forecasting electricity consumption. The relatively low number of estimators combined with a deeper tree depth proved effective in capturing complex patterns in the data without excessively increasing computational demands. The predictions on the test set as illustrated in Figure 14 clearly map the true values.

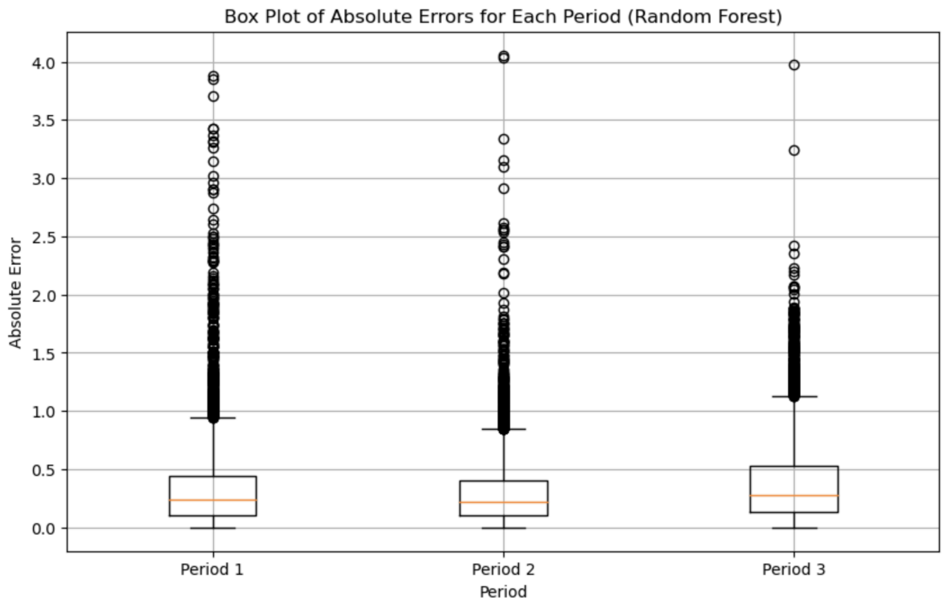

Figure 17: Box Plot of Absolute Errors for Each Period (Random Forest)

The box plot of absolute errors for the three periods (Figure 17) suggests the Random Forest model appears to perform consistently well across different periods, especially in maintaining low prediction errors.

4. XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting that uses decision trees as base learners. This model is celebrated for its scalability, efficiency, and effectiveness in dealing with various types of structured data. XGBoost improves model performance and speed by optimizing computational resources and implementing advanced regularization techniques, which help reduce overfitting.

Hyperparameter Tuning

To enhance the performance of the XGBoost model, we undertook a rigorous hyperparameter optimization process using grid search:

1. **Number of Estimators (n_estimators):** This parameter defines the number of gradient boosted trees to be used in the model. We experimented with 50, 100, 300, 500, and 1000 estimators to identify the most effective number that provides the best trade-off between prediction accuracy and model training time.
2. **Maximum Depth of the Trees (max_depth):** The depth of each tree, which is a critical factor in controlling model complexity and overfitting, was varied among 2, 4, 6, and 8 levels. Smaller values prevent the model from learning overly specific details of the training data.
3. **Learning Rate (learning_rate):** This parameter shrinks the feature weights after each boosting step to prevent overfitting. Values tested included 0.02, 0.05, and 0.1, to manage the speed at which the model learns patterns in data.

Training Methodology

The XGBoost model was trained using a TimeSeriesSplit cross-validation approach, similar to that used for Random Forest. This approach is particularly effective for time-series data as it respects the temporal order of observations. We utilized five splits in our cross-validation to ensure a thorough evaluation across different subsets of the data.

Results

The optimal hyperparameters for the XGBoost model, as identified through grid search, were:

- n_estimators: 50
- max_depth: 2
- learning_rate: 0.02

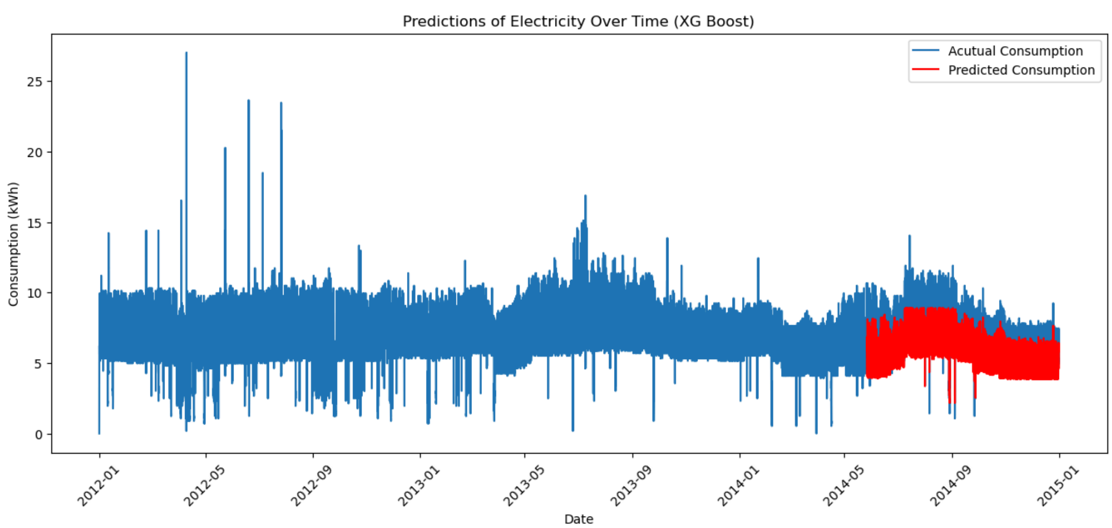

Figure 18: Predictions of Electricity Over Time (XGBoost)

XGBoost model achieved a MAPE of **8.78%**, demonstrating its capability to effectively forecast electricity consumption. However, this performance is higher than the Random Forest model, which achieved a significantly lower MAPE of 5.31%. The lower MAPE indicates that Random Forest was more accurate in predicting electricity consumption for our dataset.

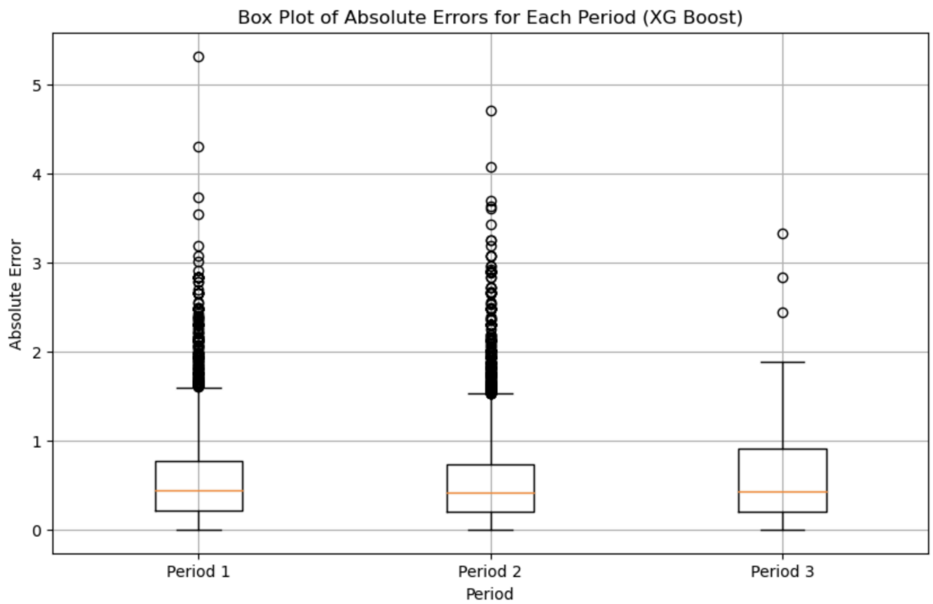

Figure 19: Box Plot of Absolute Errors for Each Period (XGBoost)

According to Figure 19, while the XGBoost model maintains relatively stable median errors in the first two periods, Period 3 exhibits a slight increase in the range of prediction errors. This suggests that the model might be encountering challenges with data patterns or anomalies not as prevalent in earlier periods.

Model Comparison

Model Comparison

Models	Pros	Cons
Random forest	Better controllability.	Training is slow; Requires manual selection of parameters.
Elastic-Net Regularization	Capable of identifying and leveraging complex temporal relationships; Can handle various data types (continuous, categorical) and incorporate external information; Short computation time.	The model's architecture is complex, and the magnitude of hyperparameters is large.
XGBoost	Parameters that are easier to tune; Automatic detection of change points in the trends.	Less control over the model's specific components.
Facebook Prophet	Automatically detects and adjusts for seasonality, which is a significant advantage when dealing with complex patterns that are not straightforward to model manually.	The underlying trend model is linear, which might not adequately capture more complex or non-linear relationships in the data over time.

Results

Comparisons of Model Performance

Previous Group's Results

Model	Best Hyper-parameter	Overall Test MAPE	Period 1 (2013.1-2013.4) Test MAPE	Period 2 (2014.1-2014.4) Test MAPE	Period 3 (2014.9-2015.1) Test MAPE
SARIMA	N/A	N/A	12%	39%	10%
Facebook Prophet	N/A	N/A	80%	10%	12%
TFT	N/A	N/A	Unknown	Unknown	Unknown

P.S.1: We marked the MAPE values for the TFT model as "Unknown" because the provided code resulted in errors and did not output the MAPE values.

P.S.2: **The previous group misunderstood the professor's requirements**. They made three different train test splits, trained the models three times on different training sets, and got the Test MAPE on three discontinuous testing sets. Therefore, they don't have the Overall Test MAPE to compare with our improved results.

Our Improved Results

Model	**Best Hyper-parameter**	**Overall Test MAPE**	**Period 1 Test MAPE**	**Period 2 Test MAPE**	**Period 3 Test MAPE**
Random Forest	{'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 50}	**5.31%**	4.51%	4.31%	7.11%
Elastic Net Regression	{'alpha': 1, 'l1_ratio': 0.9}	**8.23%**	7.13%	7.38%	10.17%

XGBoost	{'learning_rate': 0.02, 'max_depth': 2, 'n_estimators': 50}	**8.78%**	7.53%	7.90%	10.90%
Facebook Prophet	{'changepoint_prior_scale': 0.05}	**23.72%**	22.78%	30.01%	18.37%

The table above presents a detailed comparative analysis of four different forecasting models: Facebook Prophet, Elastic-Net Regression, Random Forest, and XGBoost. Each model's performance is evaluated based on the Overall Test MAPE and the MAPEs of the three periods. As the table shows, Random Forest not only achieves the lowest Test MAPE but also maintains low error rates across all periods, indicating its robustness and effectiveness in capturing and predicting complex patterns in electricity consumption data. Meanwhile, Elastic-Net shows a strong performance with good stability across different periods compared to XGBoost and Prophet, making it a reliable choice for consistent model behavior. It is worth noting that it has a very short computation time compared to the other models. Prophet, while not as accurate overall, provides valuable insights into seasonal and holiday effects, which could be beneficial in combination with other models to enhance forecasting accuracy during specific times.

In conclusion, Random Forest stands out as the most effective model for forecasting electricity consumption due to its accuracy and consistency, with the best test MAPE of 5.31%.

Best Performing Model Analysis (Random Forest)

Feature Important Analysis

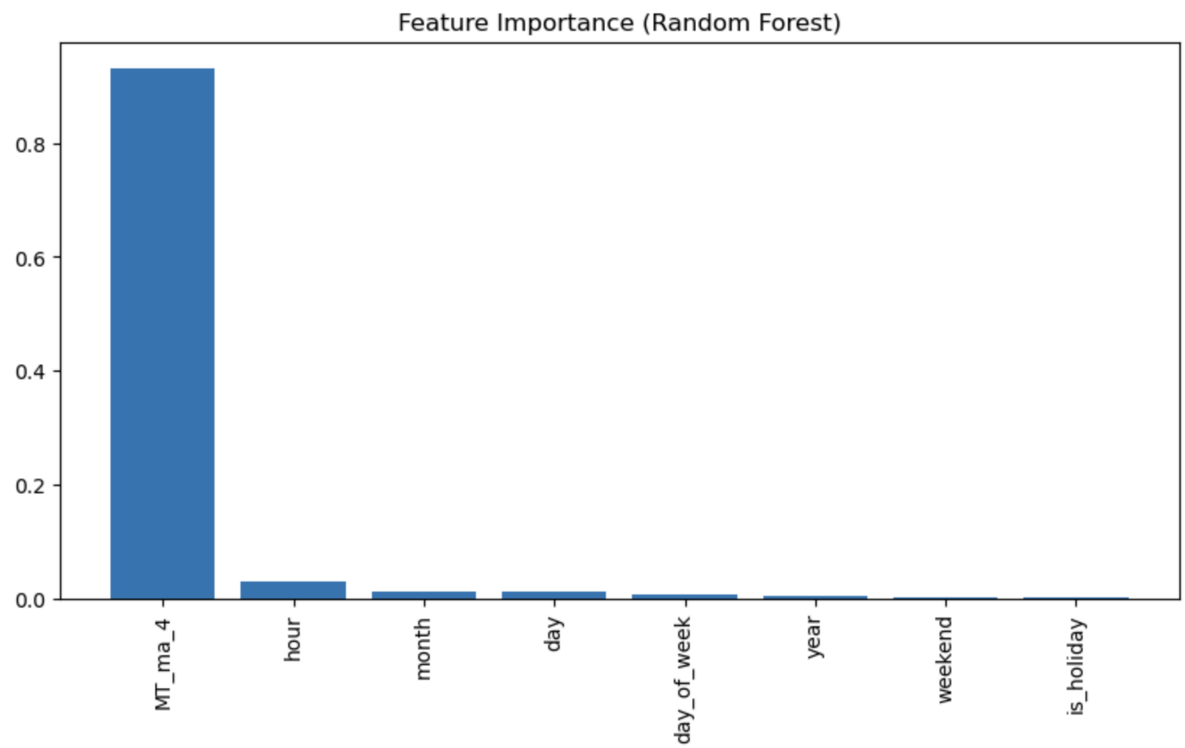

Figure 20: Feature Importance for Random Forest Model

This feature importance plot from the Random Forest model highlights the relative significance of various features used to predict electricity consumption. The most influential feature, by a considerable margin, is MT_ma_4, which represents the moving average of electricity consumption over the past hour. This indicates that short-term historical consumption is a critical predictor, reflecting immediate past conditions as a strong indicator of near-future usage.

Other features such as hour, month, day, day_of_week, year, weekend, and is_holiday show significantly less importance compared to MT_ma_4, suggesting their relatively minor direct impact on the forecasting model.

The analysis suggests that while short-term past electricity usage (MT_ma_4) is a dominant predictor, other time-based features also contribute to the model, albeit to a lesser extent. These features help capture various temporal dynamics but are overshadowed by the immediate past consumption data, which appears to be more predictive of future usage.

Box Plot Analysis

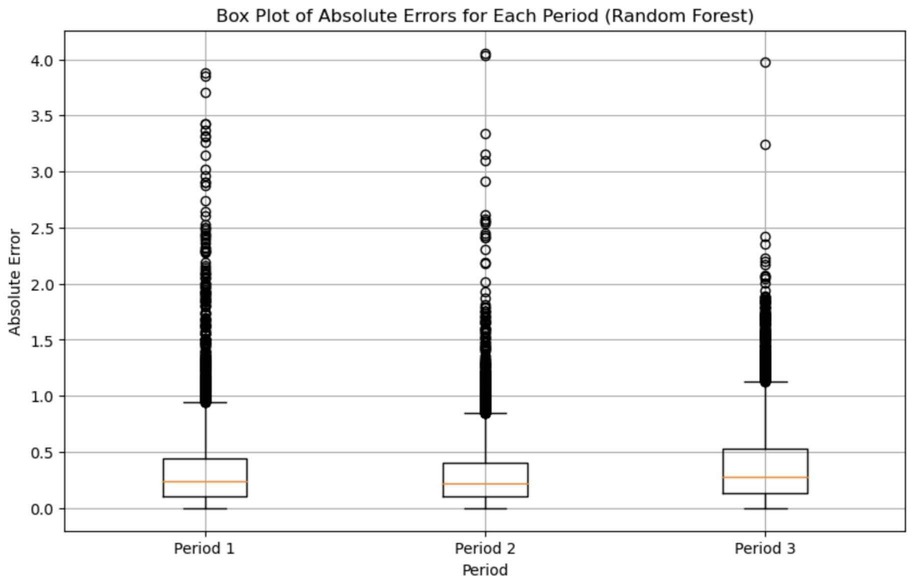

Figure 21: Box Plot of Absolute Errors for Each Period (Random Forest)

This box plot represents the distribution of absolute errors in electricity consumption predictions made by the Random Forest model across three different periods.

Period 1:
The median error in Period 1 is observed to be lower than 0.5, indicating that at least 50% of the prediction errors are below this value.

Period 2:
Maintains a level similar to Period 1, reinforcing the model's consistent predictive performance.

Period 3:
The median error is slightly higher but still quite similar to the previous periods.

Overall, the box plot suggests that the Random Forest model's prediction accuracy remains stable across the three different periods. In the meantime, across all periods, a significant portion of the data points exhibits prediction errors below 0.5, signifying a high level of precision in the model's ability to forecast electricity consumption.

Further Improvements

Incorporation of Additional Data Types:

- **Weather Data Integration:** Beyond basic temperature and precipitation, including solar radiation levels, wind speed, and humidity might enhance the model's ability to predict usage patterns, especially in areas reliant on heating and cooling.

Including Additional Time-Based Features:

- **Expanding Moving Averages:** Given the substantial influence of the moving average of the previous hour (MA_4) on model performance, we propose adding longer-term moving averages to capture broader consumption trends. Specifically, incorporating a moving average for the previous 24 hours (MA_24) could help the model account for daily consumption cycles, enhancing its ability to forecast based on daily peaks and troughs in electricity usage.

Adoption of Advanced Modeling Techniques:

- **Deep Learning Models:** Implementing deep learning approaches such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer models to capture the complex non-linear relationships typical in electricity data.
- **Hybrid Models:** Combining machine learning models with traditional statistical methods (such as SARIMA) might yield better results by leveraging the strengths of both approaches.