Forecasting of Electricity Consumption

**Group 1:**

Wenxin Zhang (wz2585)

Yajie Zhang (yz3876)

Ran Liao (rl3303)

**Group 2:**

Hugo Ginoux (hg2632)

Olivia He (qh2266)

Xinyu Wang (xw2814)

Zhening Zhang (zz3040)

**Github for Code**: <https://github.com/OliviaaaHe/forecasting_proj2>

# Contents

[Introduction](about:blank)

[Data Source](about:blank)

[Data Processing](about:blank)

[Target Variables](about:blank)

[Predictive Variables](about:blank)

[Pre-modeling](about:blank)

[Modeling](about:blank)

[Algorithmic Solution Results](about:blank)

# Introduction

Electricity, as one of the most important energy sources in our daily life, gains more and more attention from the public. How much electricity needs to be produced and how to distribute the electricity over the country has become a very important issue facing the government. A reliable electricity forecasting system can help the government estimate the electricity consumption of an area over a certain period, and then achieve efficient distribution of electricity.

In this project, we are going to analyze the time series of electricity load using different forecasting techniques. The purpose of this project is to learn the pattern from past electricity consumption data to forecast the quantity of electricity consumption in the future under the assumption that the pattern will resemble itself in the future. To achieve this goal, several ML models will be used, and the accuracy of the prediction is based on the RMSE of the prediction output.

The purpose of this document is to provide a detailed technical overview of

* Overview of the dataset
* Feature engineering
* Modeling
* Evaluation of Models

**Based on the previous group’s work, we mainly make the following improvements:**

1. **Changing the spectral clustering parameter to ‘nearest\_neighbors’ to get a more balanced division of clusters.**
2. **Selecting the representative clients by computing the degree of freedom using the affinity matrix generated by spectral clustering.**
3. **Changing the evaluation of predicting other clients for each group from selecting 6 clients and computing their mean MAPE to computing the MAPEs of every client for each group and getting their medians.**
4. **Adding a method that trains a model for every client and uses the median of MAPEs as the overall error which significantly improves the MAPE.**
5. **When predicting overall electricity consumption, add a new feature “number of people” which significantly improves the model performances.**

# Data Source

The dataset used in this project was collected from UCI Machine Learning Repository, which is called [ElectricityLoadDiagrams20112014 Data Set](https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014). It was stored as a csv file and was downloaded in a zipped folder. The data are stored locally and are processed and integrated using Python scripts.

# Data Processing

## Data Overview

The Electricity Load Diagrams dataset is a time series dataset that is saved as txt in CSV format. The data are separated using semicolons. By using the function “read\_csv” in Python, we can read the dataset in the data frame format.

## Data Diagnostics

This dataset describes the electricity consumption of different clients in the period from January first, 2011 to January first, 2015. It contains 140256 records and 370 features in total. Each feature represents a unique client. Each row of the dataset represents a specific timestamp between 2011 and 2014, so there are no duplicate records. Notice that the gap between two consecutive timestamps is 15 min, and therefore the unit of values in the dataset is kW per 15 min.

By looking at the dataset, we found that values are read as strings and there is a comma in most of the values. So, we replace the comma with a decimal point to make the data readable in Python. There are no missing values in the dataset, but for those clients that are generated after January first, 2011, the value of consumptions before the date they were generated is recorded as 0. So we looked at the first non-zero entries in each column and grouped them by patterns, presenting the top 4 values below

|  |  |
| --- | --- |
| First non-zero date index | Count |
| 35040 | 160 |
| 0 | 159 |
| 106464 | 11 |
| 70176 | 3 |

The above table indicates that almost half of all columns do not have data in the whole 2011 year. To avoid the noise from missing values, we need to handle these unknown values.

Moreover, as the records were taken every 15 minutes, there are 96 records in a day. But one thing that needs to be noticed here: for every year in March time change day the values between 1:00 am and 2:00 am are zero for all points; for every year in October time change day, the values between 1:00 am and 2:00 am aggregate the consumption of two hours.

## Data Exploration

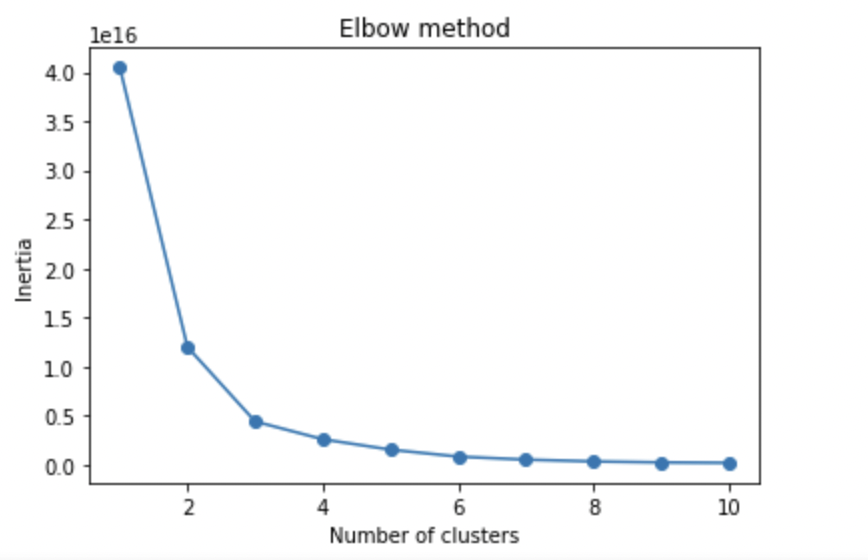
We first plotted the graph for the daily electric load of all clients from 2011 to 2014. The graph was shown below:

图表, 直方图

描述已自动生成

It indicated that although the variation and mean value of each client's daily electricity load is different, there were some clients’ electric loads that have similar periodic changes. Therefore, to have a better prediction of each client, we split the clients into different groups based on their historical data.

By using the K-mean algorithm, based on Euclidean distance, as the graph shows below, we identified 3 groups behaving differently. Therefore, we split the dataset into 3 clusters.



## Data Preparation

In order to do the prediction of electric load for clients, the original dataset was cleaned, and new features were generated to improve the accuracy of the prediction.

For data cleaning, we first convert the data from string to float and replace the comma with a decimal point to make the data readable in Python. Moreover, to make the data more reasonable and improve the accuracy of prediction, instead of using units in kW per 15 min, we used the daily electricity load as the unit of measurement. Finally, we converted the date from string to datetime.

For feature engineering, we generated two types of new features for each client. First, the 10 lagged values are added to the dataset (the energy consumption in the last 10 days). Another group of features is time-related features, which include year, month name, and day of the week that are created using datetime.

# Target Variables

Since this is a prediction project, the target variable is the value of electric load in the future. For the purpose of this project, we used 2 types of target variables. We performed time series forecasting on

1. Individual client meters. For each group of clients, we choose the most representative client in the group to predict their consumption, thanks to the “affinity matrix” property of spectral clustering.
2. The total electricity consumption of each of the 3 groups.

# Predictive variables

There are two types of predictive variables in this project:

1. Direct variable – These variables were directly from the dataset that was provided by direct customers
2. Derived variable – These variables were created by manipulating the direct variables

**Variable List:**

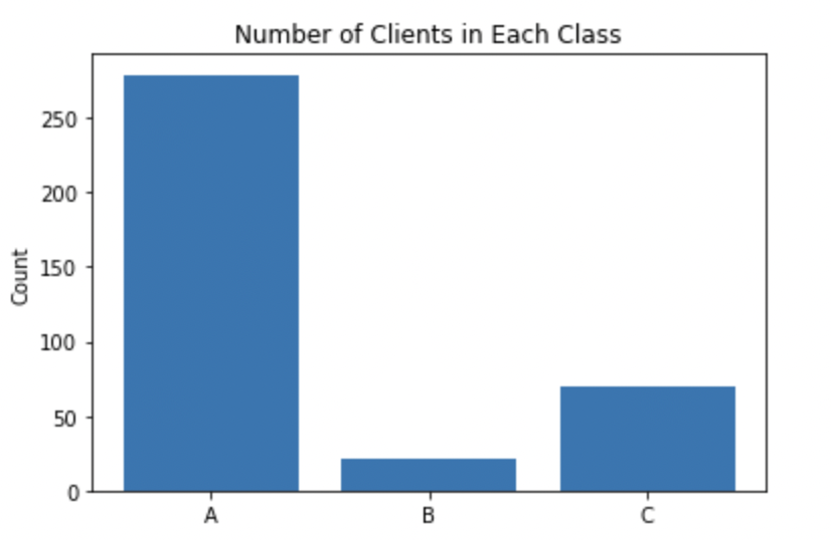
* Client\_ID: the value of the electric load on a specific date for the specific client
* Lagged\_x: quantity of electric load on the day x days before the specific date, x ranges from 1 to 10, these predictors are equivalent to AR(1) to AR(10) terms.
* Year: year of the date
* Month: month of the date
* Day: day of the date
* Weekday : Monday to Sunday

# Pre-Modeling

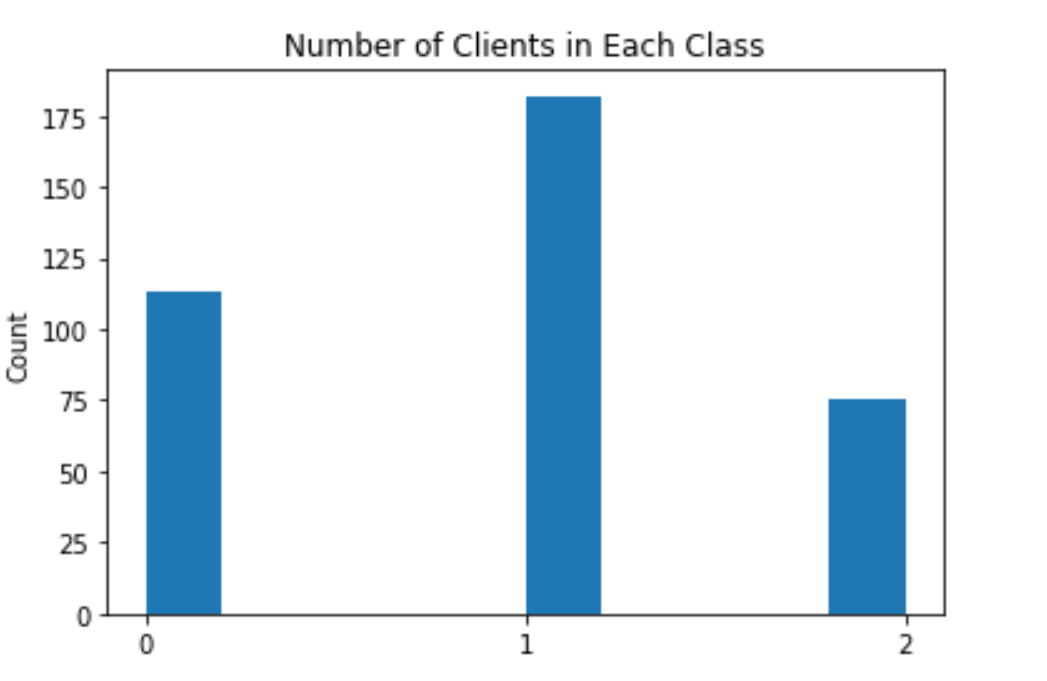
## Classification of Clients

To split the clients into different groups, we used spectral clustering. Based on their historical electricity load, we split them into 3 different groups.

The team before us was using Spectral Clustering with ‘rbf’ affinity metric to make this clustering. However, this method leads to very unbalanced clusters, with one big group containing almost all the clients and 2 tiny groups aside.



After trying different clustering methods, we find Spectral Clustering performs the best. So we change the affinity metric to ‘nearest\_neighbors’ to get much more balanced clusters. As the figure shows below, there are 113 clients in the first group, 182 clients in the second group, and 75 clients in the third group.



After assigning clients to different groups, we choose clients ‘MT\_171’, ‘MT\_047’, and ‘MT\_250’ as the representative to build the predictive models for groups A, B, and C respectively. The clients are selected based on the degree of freedom using the affinity matrix created by the spectral clustering instead of selecting the first client for each group.

## 

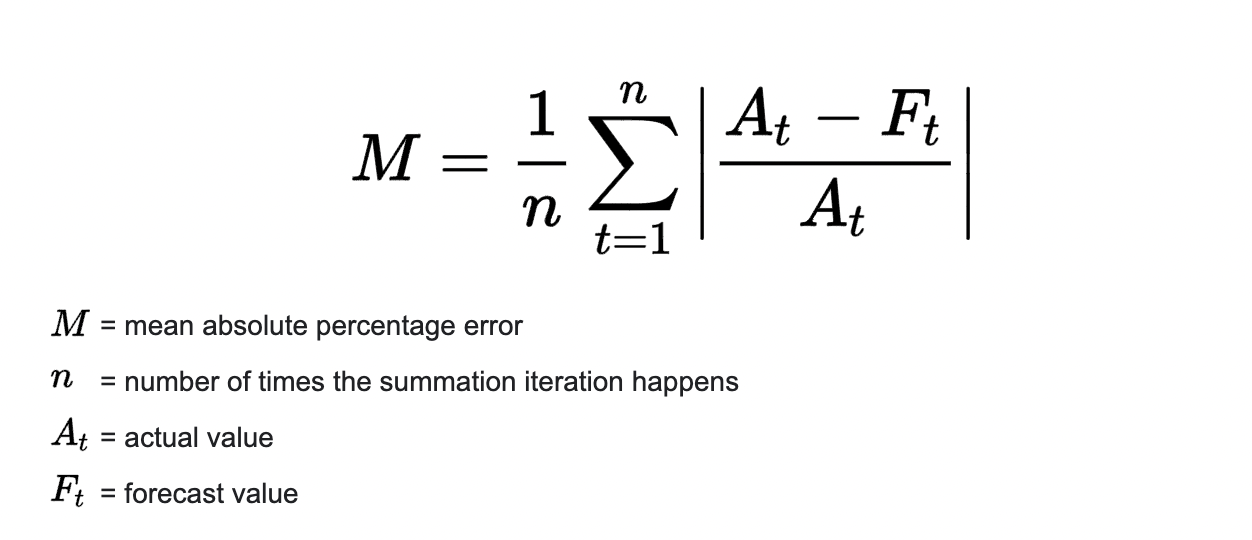
## Train & Test Split

We have 1461 records left, which are records for 4 years (from 2011 to 2015). To make the prediction more accurate by avoiding overfitting, we split the data into training and testing dataset along with years : the 70% first records are assigned to the training set, the next 30% to the test set.

We tested 6 different models : Linear regression, XGBoost, Random Forest, Prophet, SARIMA and LSTM. For the 6 models, we split the training set into development and validation to optimize some of the hyperparameters: L1 and L2 regularization for the linear regression, number of estimators and depth of the trees for XGBoost and random forest, layers sizes for LSTM, change\_points for Prophet, and (p, d, q, P, D, Q) for SARIMA.

## Evaluation Metric

In this project, we use MAPE as our evaluation matrix. MAPE represents the mean or average of the absolute percentage errors of forecasts. The function of it is as below:



# Modeling

After the data was cleaned and new features were generated, we built 2x6=12 models based on the training dataset and tested the performance of the test model.

The Algorithmic Solution was built for two major targets:

* Predict the next one-year electric load for the representative client and the rest clients from each group
* Predict the total electric load for each group

## 

## Models used for prediction

### First target: individual client

* + **GLM**

|  |  |  |
| --- | --- | --- |
| **Model** | **Pros** | **Cons** |
| **Linear regression** | Simple model, easy to interpret | Bad at capture nonlinear correlations |
| **XGBoost** | Better capture nonlinear relationship, and process fast | Cannot capture the weekly trend of the time series |
| **Random Forest** | Easy to interpret | Loss some correlation between features |

The dataset itself contains every person's usage per day, along with the features including: different Months, different days in week, and time difference from the first day that customer started using, a total of 21 features. The target variable is solely the usage of each person per day. We used spectral clustering, divided this group of people into three clusters, and trained a model for each group.

However, since three clusters couldn’t capture every type of customer, there’s a large fluctuation inside each group, which gave us a large error in MAPE. The results are inside the result section. To test out if this method works, we even tried to make a dummy variable for every customer, which means consider each person’s identity as a feature, and that didn’t improve our model much as well.

* **Time Series Models**
  + **Model Descriptions**

**1. SARIMA Model Using Auto-ARIMA**

* + Used the SARIMA Model since it considers the trend and seasonal effect based on the ARIMA model.
  + Order of the AR term (p).
  + Order of the MA term (q).
  + Order of the differencing (d).
  + Seasonality (m)
  + AIC was selected for model selection.

**2. Facebook Prophet with Exogenous Features**

* + Used Facebook Prophet which decomposes a time series into trend, seasonality, and holiday effects.
  + Added the weekday, month, and day as exogenous features.
  + **Training Methods and Results**

Since the two models are time series models, we can’t break the data structure to train the model as GLM models do. It means every time we train the model, only one time series is allowed to be passed into and the time order can’t be changed. Here we use two methods to train the model.

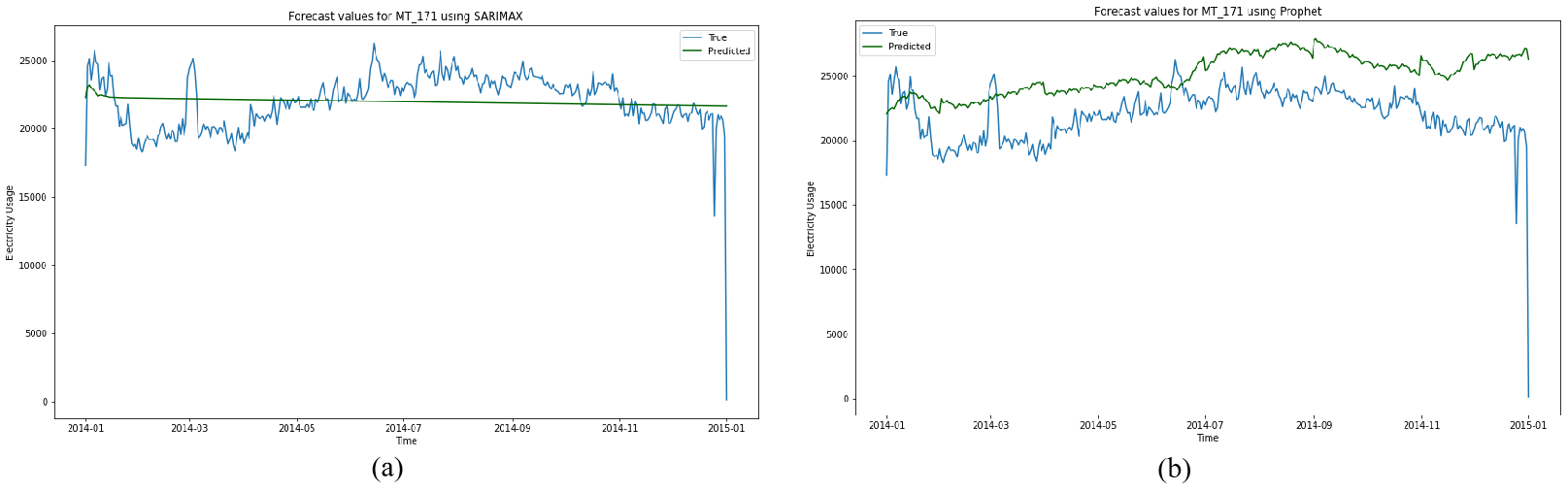
**1. Use one client to train the model and use it to predict the others.**

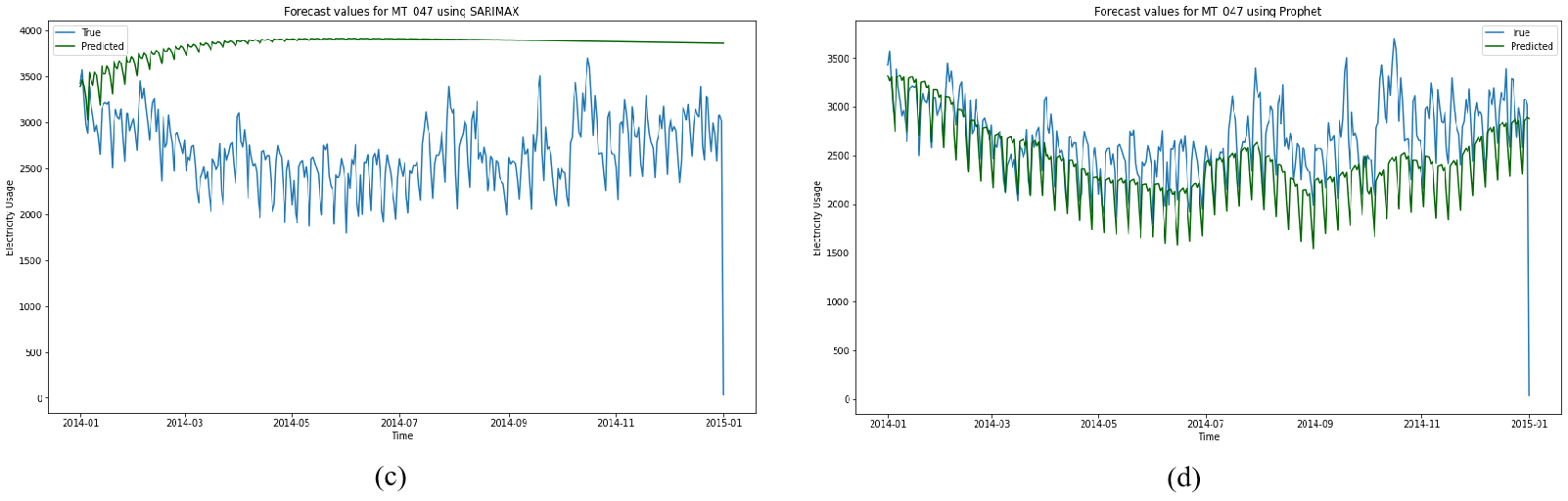
The method is similar to what the previous group used. We use the selected client to train the model for each group and deploy the model to predict the usage of all the other clients for each group.

For example, for clients in class A, we use MT\_171 to represent the clients in the group and build models based on MT\_171’s historical data (from 2012 to 2014). Then we can get the predicted usage of client MT\_171 from 2014 to 2015 and it will also be the prediction for all the other clients’ usage from 2014 to 2015 because in time series models, the feature is the time. So the features for all the clients are the same in the same time period . Similarly, for clients in class B, we use MT\_047 and MT\_250 for clients in class B and class C respectively.

**But we have a different evaluation method from the previous group.** **In their previous work, they evaluated the overall performance by choosing the first 6 clients for each cluster and calculating the mean of their MAPEs to represent other clients for each group. But we think only 6 clients cannot represent the whole group. As long as we are to evaluate the performance of the model on other clients for each group, we should use all the other clients. Also, the mean metric can be easily affected by the outliers. So in our model evaluation part, we use the median of the MAPE for all the other clients to represent the overall MAPE. We think it’s a more reasonable method.**

Below are the graphs for showing the performance of the prediction based on SARIMA and Prophet for the three clients.

****

****

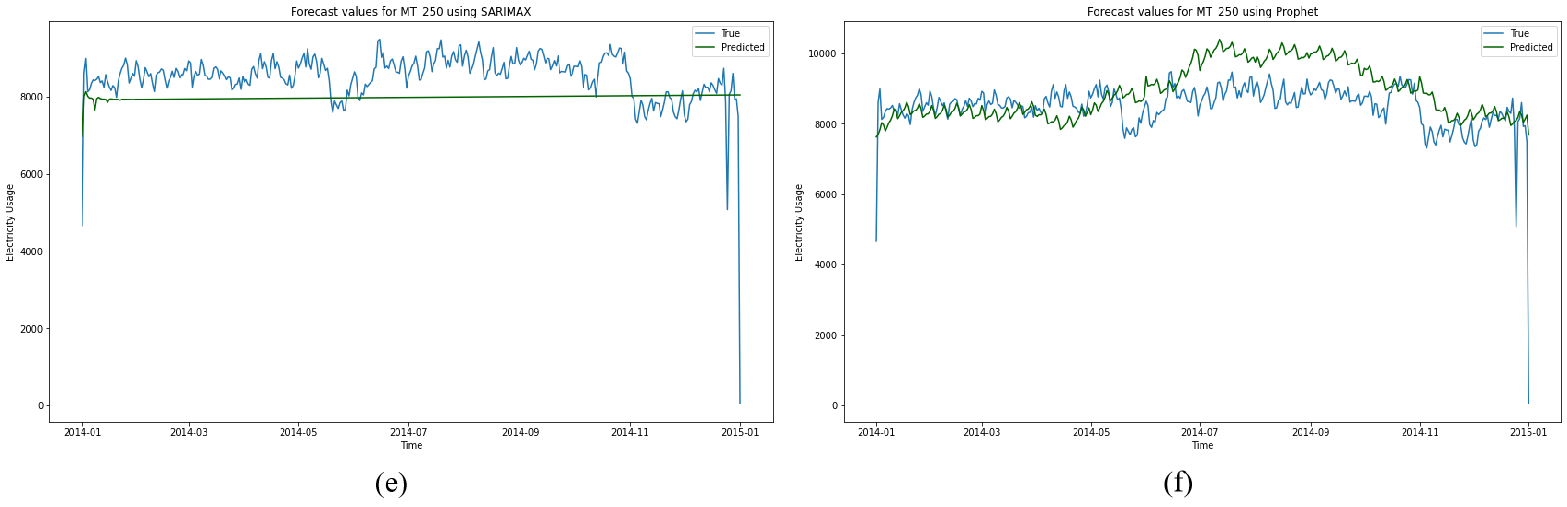
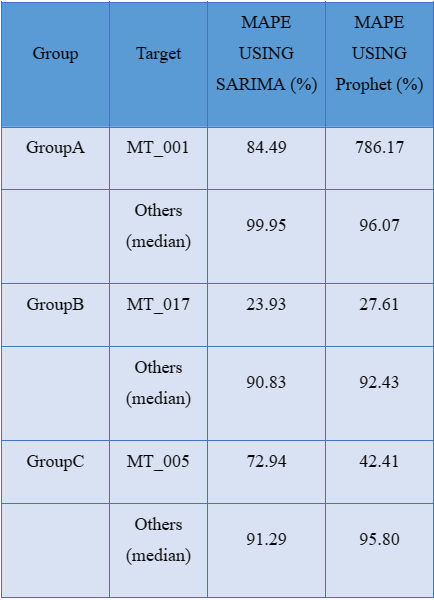
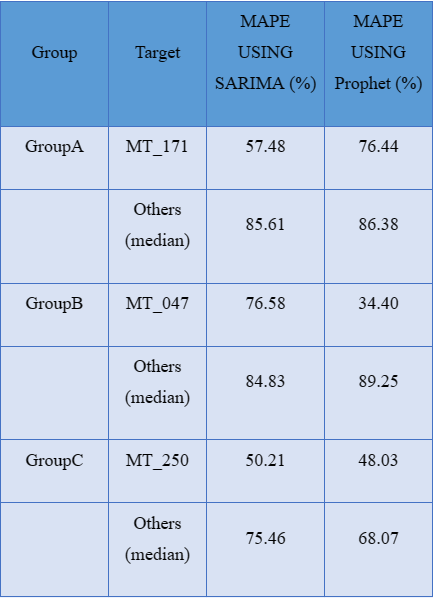
****

Figure (a), (c) and (e) show the predicted electricity usage for MT\_171, MT\_047 and MT\_250 respectively using SARIMA. From the figures, we can see that the predicted values don’t have much fluctuation. They are more like lines with small slopes. But in figure (b), (d) and (f), which are the results using Prophet, the predictions have similar trends with the true values, especially for the predictions for the clients in class B and class C. The predictions for MT\_171 capture the trend but the values are higher than the truth. Therefore, comparing SARIMA with Prophet, we can conclude that for MT\_171, SARIMA has a better result than Prophet while for the other two, Prophet has better fit.

The MAPE is shown in the following table. **On the left is our result and on the right is the result from the previous group.**



From our result, we can see that the MAPEs for the three clients are smaller than the MAPES for the others respectively. It’s consistent with our common knowledge because the models are built based on the three clients. Also, although in the previous figures, SARIMA doesn’t capture the trend of the data, it performs slightly better than Prophet in GroupB. Perhaps because Prophet is more client-targeted, it has worse adaptability. Both SARIMA and Prophet perform the best on GroupC.

**Since we changed the clustering method, the way to select the representative clients and the evaluation metrics, it’s difficult to directly compare our result with theirs. So we use their clustering method, their clients and our evaluation metrics to compare our performance. The result is shown on the right. We can focus on the predictions for other clients for each group. Compared with their results, our results are obviously better than theirs. So we think our improvement on clustering and selecting clients do make a difference.**

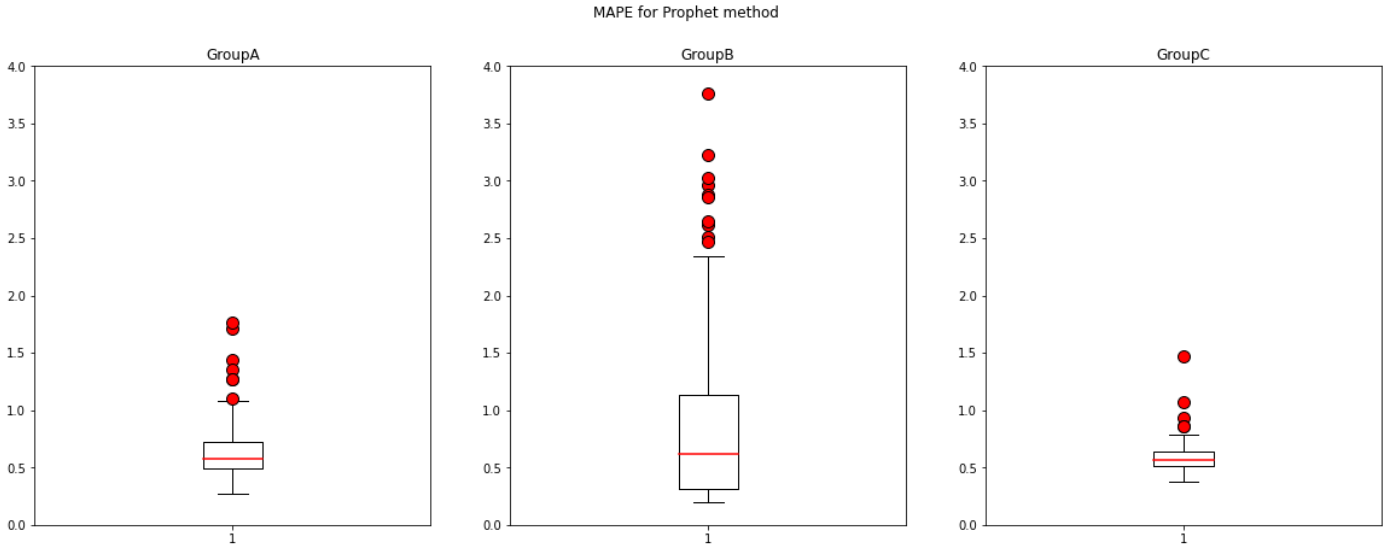
**2. Create a model for every client.**

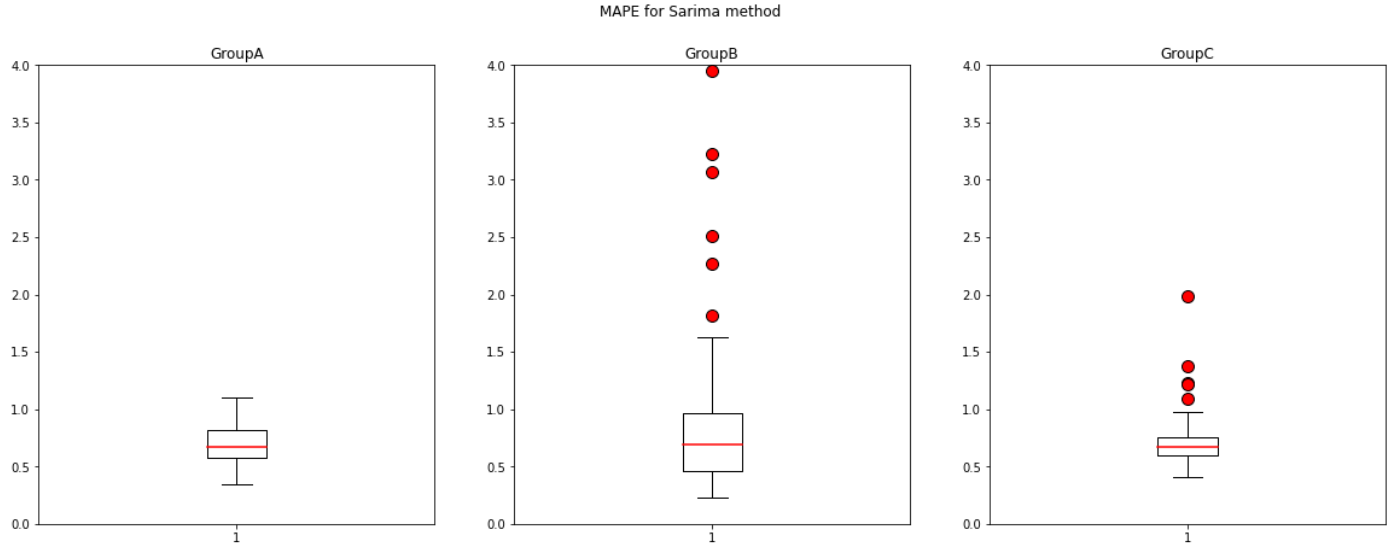
**This method is proposed because we think it’s difficult to just use one client’s predictions to represent others’. So we create a model for each client to predict their electricity usage and use the median of the MAPEs to represent the performance.** We use SARIMA and Prophet respectively and the result is shown in the table below.

|  |  |  |
| --- | --- | --- |
| Group | SARIMA | Prophet |
| GroupA | 67.00% | 57.48% |
| GroupB | 68.79% | 61.97% |
| GroupC | 66.82% | 56.55% |

**From the table, we can find that the performance is obviously better than the first method. But it will take more time to train.** Both SARIMA and Prophet perform the best on GroupC, which is consistent with our result of the first method. So the patterns of the electricity usage of clients in GroupC may be easier to capture than GroupA and GroupB.

We also create boxplots to show the distribution of the MAPEs for each group instead of just showing the median. The first figure is the MAPEs of Prophet models and the second figure is the MAPEs of SARIMA models.

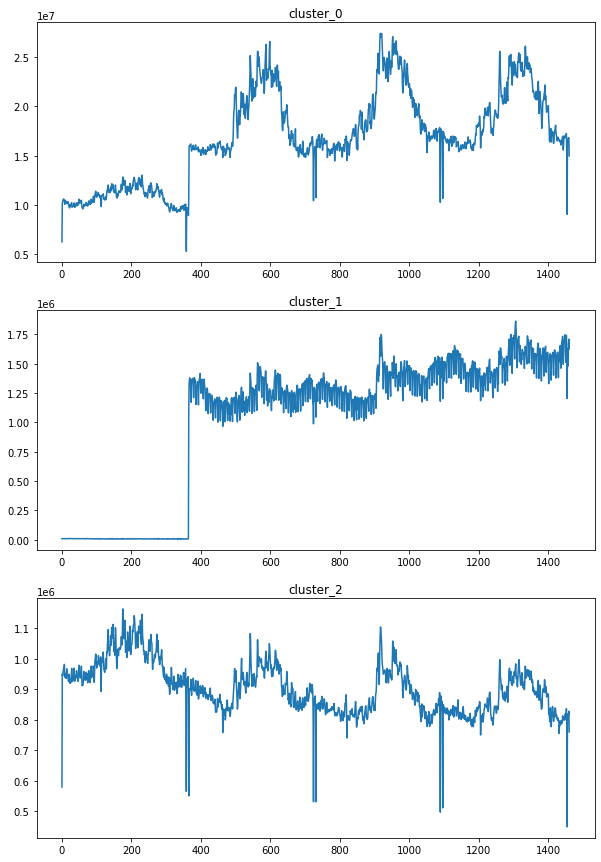




Through these figures, we can see that although the median of Prophet is smaller than the median of SARIMA, SARIMA has fewer outliers and the MAPEs are gathered more closely (the bounds are smaller). The two methods both have a poor performance on GroupB and a good performance on GroupC.

### Second target: total electricity consumption

For each group, we also trained the models on the total electricity consumption (usage). The total electricity consumption is calculated by taking the sum of all the client’s daily consumption in the group.



Now, our target variable is the total daily electricity consumption. Then, we added four features: time spent (current time minus start time), weekday, month, and the number of people. Four models are used and the one with lowest MAPE is selected as the best model.

|  |  |  |
| --- | --- | --- |
| **Model** | **Pros** | **Cons** |
| **Linear regression** | Simple model, easy to interpret | Bad at capture nonlinear correlations |
| **XGBoost** | Better capture nonlinear relationship, and process fast | Cannot capture the weekly trend of the time series |
| **Facebook Prophet** | Automatic and effective model exclusive for time series | Performance varies between different training data set |
| **SARIMA** | Good at capturing the seasonality | Overfitting for data without seasonality |

The addition of the number of people to the features significantly improved our model result. It is one of the biggest improvements we made. Given that it is very natural to think the amount of people has a positive effect on the total daily electricity consumption, we computed the number of people present by counting clients with only positive electricity consumption in the day. After adding this feature, it turned out to significantly improve our model’s performance.

Conclusion: for all three groups, the linear regression has the best prediction with MAPE smaller than 10%.

# 

# Algorithmic Solution Result

## The Previous Group

The previous group used MT\_001 to represent Group A, MT\_017 to represent Group B, MT\_005 to represent Group C, then built the models for different groups based on these three representative clients. The models can capture general periodicity of the electric load of clients in all three groups. but they sometimes cannot be accurate enough to capture the fluctuation especially when the electricity load is high.

### MAPE Value (The Previous Group)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | Target | Linear regression (rolling) | SARIMA | Prophet |
| Group A | MT\_001 | 155.5462 | 63.3110 | 778.6575 |
|  | Other clients in group A | 91.2347 | 93.6691 | 82.9968 |
| Group B | MT\_017 | 23.0439 | 39.0796 | 27.0015 |
|  | Other clients in group B | 71.5003 | 85.8627 | 56.7704 |
| Group C | MT\_005 | 28.1081 | 63.6468 | 42.8828 |
|  | Other clients in group C | 51.8436 | 52.2571 | 52.6164 |
| Predict total |  | 676809445.0432 | 61.1313 | 58.2253 |

### 

### MAPE in Different Stages (The Previous Group)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | Model | 1st 4 Months | 2nd 4 Months | 3rd 4 Months |
| Group A | Linear Regression | 57.3101 | 80.2631 | 329.0655 |
|  | SARIMA | 26.5873 | 58.0655 | 105.2804 |
|  | Prophet | 435.5251 | 470.2678 | 1430.1794 |
| Group B | Linear Regression | 3.6215 | 2.7701 | 62.7402 |
|  | SARIMA | 19.0486 | 22.7028 | 75.4872 |
|  | Prophet | 5.8465 | 14.2343 | 60.9238 |
| Group C | Linear Regression | 5.2779 | 7.0391 | 72.0074 |
|  | SARIMA | 21.3223 | 65.6995 | 103.9187 |
|  | Prophet | 12.0889 | 25.1482 | 91.4113 |

In most cases, the value of MAPE is increasing as the time goes by, and that is reasonable since we use sliding windows in our predictions, so the prediction of the next four months is dependent on the prediction in the previous four months. Therefore, the prediction error increases along with time.

## Our Group

We applied GLM and the two time series models listed in the **modeling/First target** to predict the first target variable: individual client’s usage/consumption per day. We used their months, weekday, time lasting features. For the second target variable of total electricity consumption, we use the four models listed in the **modeling/Second target** section. The results are listed below.

### MAPE Value for individual client

#### GLM

The table below shows the median MAPE for each group under different models.

|  |  |  |  |
| --- | --- | --- | --- |
| Group | Linear regression | XG boost | Random Forest |
| Group A | 1503.46% | 1421.87% | 1451.33% |
| Group B | 2398.88% | 2384.42% | 2505.61% |
| Group C | 1314.54% | 1263.67% | 1362.39% |

#### Time Series Models

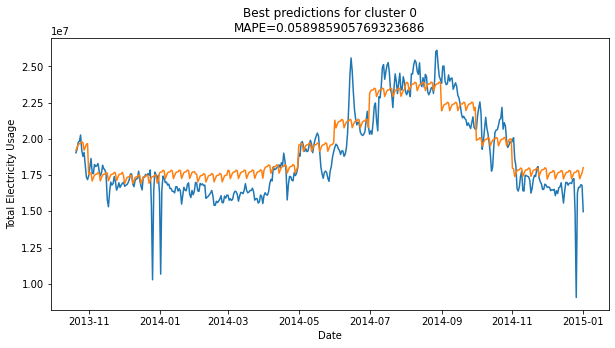
| Group | SARIMA | Prophet |
| --- | --- | --- |
| GroupA | 67.00% | 57.48% |
| GroupB | 68.79% | 61.97% |
| GroupC | 66.82% | 56.55% |

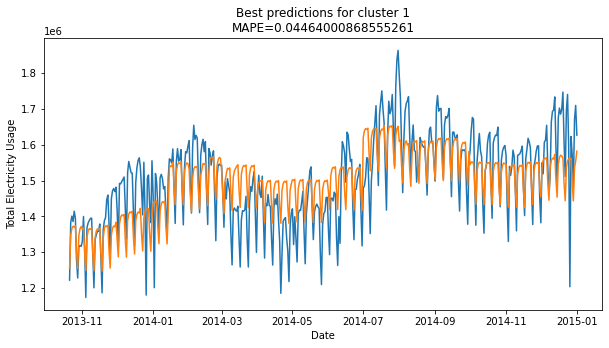
### MAPE in Different Stages for total electricity consumption

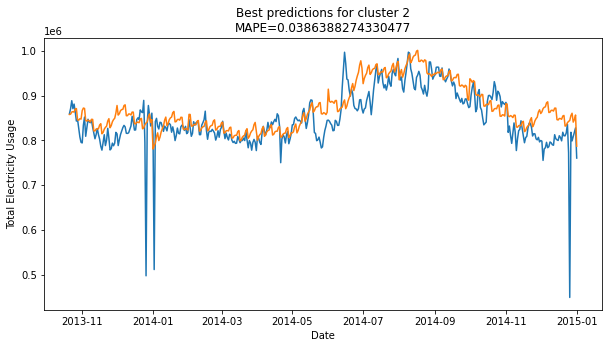
For each group, best model is selected and the MAPE in the three different stages are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | Best Model | 1st 4 Months | 2nd 4 Months | 3rd 4 Months |
| Group A | Linear Regression | 6.3% | 5.6% | 5.8% |
| Group B | Linear Regression | 4.1% | 5.7% | 3.5% |
| Group C | Prophet | 3.5% | 3.5% | 4.6% |

Overall prediction for the three different groups:







### Conclusion

**Based on the comparison, we achieve significant improvement on:**

1. **Changing the spectral clustering parameter to ‘nearest\_neighbors’ to get a more balanced division of clusters.**
2. **Get a slightly better prediction on SARIMA and facebook prophet as we select a more “center” client instead of randoming choosing the first one by the name.**
3. **Significantly improve the result to around 5% MAPE for prediction of the total electricity, by adding two features: time spent and the amount of present clients.**

**We fail to make a significant improvement on:**

1. **Improving the result of individual prediction using GLM.**

# Discussion

We’ve already got a good result for predicting overall electricity consumption. However, we cannot get a good prediction on individual usages. The best MAPE is still around 50%. The future suggestion would be to find a way to personalize a model that can fit well on each client in the group. One thought is to create more groups in order to decrease the difference within each group.