Forecasting Electricity Consumption Using Seasonal Autoregressive Integrated Moving Average (SARIMA) Method

Abu Kaoser
Student ID: 200488293
mkk544@uregina.ca

Thuy Nguyen
Student ID: 200490798
ttn070@uregina.ca

Project Summary

Time series forecasting is a prediction technique widely used in many fields of study and applications, ranging from weather forecasting to earthquake prediction to pattern recognition[1]. This project aims to predict electricity consumption in a region of the United States, allowing power companies to make better preparations for electricity production and supply management to meet customer demand, avoid power shortages during peak hours, or reduce energy waste. Due to the seasonality of the hourly electricity consumption data, the Seasonal Autoregressive Integrated Moving Average (SARIMA) method is used to forecast electricity usage over the next 24 hours. There are 1469 hourly observations from November to December 2023 to fit and build the model. The best model is automatically chosen using a stepwise approach, resulting in a forecasting model accuracy of **93.10%** (MAPE = 0.069).

Table of Contents

[1] https://www.kdnuggets.com/2020/01/predict-electricity-consumption-time-series-analysis.html

1. Introduction

Recently, Canada has experienced extreme cold weather, with some regions, particularly in Manitoba, Saskatchewan, and Alberta, seeing temperatures as low as -40^{0}C, leading to electricity shortages[2]. The main reason for this power outage is the dramatic increase in demand exceeded the capacity of the electricity supply sources[3]. In this case, if Alberta can predict its electricity consumption in a specific time interval, they might be able to find a solution that will timely coordinate supply to fulfill the demand.

Inspired by the power outage situation in Alberta, the project aims to predict energy consumption over a month, allowing energy companies to regulate supply to keep up with demand and avoid power outages like Alberta's. Various machine learning algorithms for time series data analysis have been applied to forecast electricity consumption, including autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), long short-term memory (LSTM), or the simple recurrent neural network (RNN). Each model has distinct features and is applied to specific data sets based on the nature of the data. We first plan to apply the ARIMA model to forecast electricity consumption over a **month** based on hourly time series data. However, the data exhibits a seasonal pattern, so we decided to choose the seasonal autoregressive integrated moving average (SARIMA) method, an extension of the ARIMA model, to predict data over the next **24 hours**.

2. SARIMA Method[4]

ARIMA is one of the most popular forecasting methods for time series data forecasting. However, it does not work effectively with seasonal data, which exhibits a seasonal pattern or cycle, such as monthly, quarterly, or yearly. SARIMA or Seasonal ARIMA, an extension of ARIMA helps deal with this problem.

[2] https://www.cbc.ca/news/science/extreme-cold-climate-change-1.7087754
[3] https://globalnews.ca/news/10225294/alberta-emergency-power-alert-lessons/
[4] https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/

SARIMA adds four more seasonal hyperparameters P, D, Q, and m, in addition to the three p, d, and q that are the same as ARIMA. The SARIMA model can be expressed as SARIMA (p, d, q) (P, D, Q) m, in which:

- p: Trend autoregression order.
- d: Trend difference order.
- q: Trend moving average order.
- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period.

3. Data Discovery and Preparation

This project uses a time series dataset collected by PJM Interconnection LLC (PJM), "a regional transmission organization (RTO) that coordinates the movement of wholesale electricity in all or parts of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia and the District of Columbia."[5] The electricity consumption data has been collected every hourly since 1993. However, to ensure data accuracy, we decided to use the data for 15 years starting from 2008. PJM allows us to export CSV data from their website and support us downloading data using API through their portal (https://apiportal.pjm.com/). Below is a detailed description of each field in the data file, which is published on their website.

Field name	Data type	Description
Datetime Beginning UTC	DateTime	Datetime Beginning UTC
Datetime Beginning EPT	DateTime	Datetime Beginning EPT
NERC Region	String	NERC Region
Market Region	String	Market Region
Transmission Zone	String	Transmission Zone Location
Load Area	String	Fully Metered Electric Distribution Company
MW	Number	Load in MW
Company Verified	Boolean	Indicates whether the metered load has been verified by the Electric Distribution Company

[5] https://www.pjm.com

In this phase, we merge all 16 CSV data files and do some cross checks to make sure of the quality of the data, including checking data duplication, data missing, and outliers. We first select the data of the region named “DOM” for our study. However, when we double checked data using the Python visualization tools, we found that the data in Q4 2016, June and Q2 2019 are totally missed (see Figures 1, 2, 3 for more information).

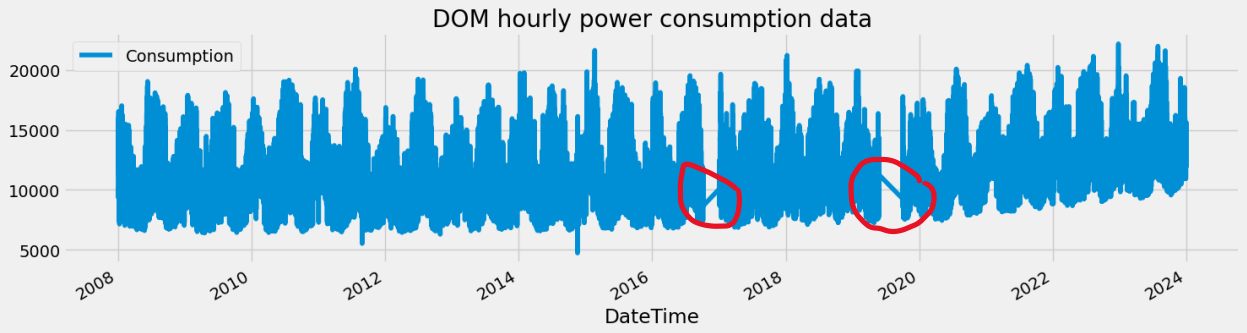

Figure 1 – Missing DOM data in Q4 2016 and Q2 2019

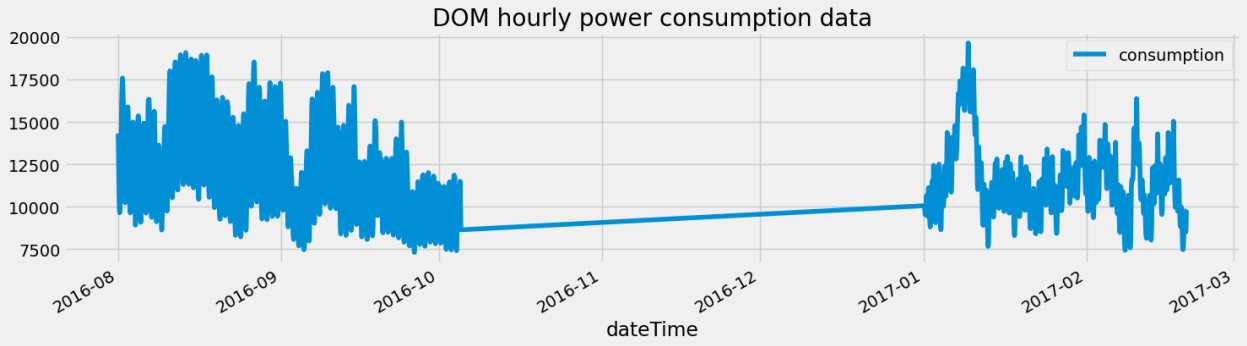

Figure 2 – Missing DOM data in Q4 2016

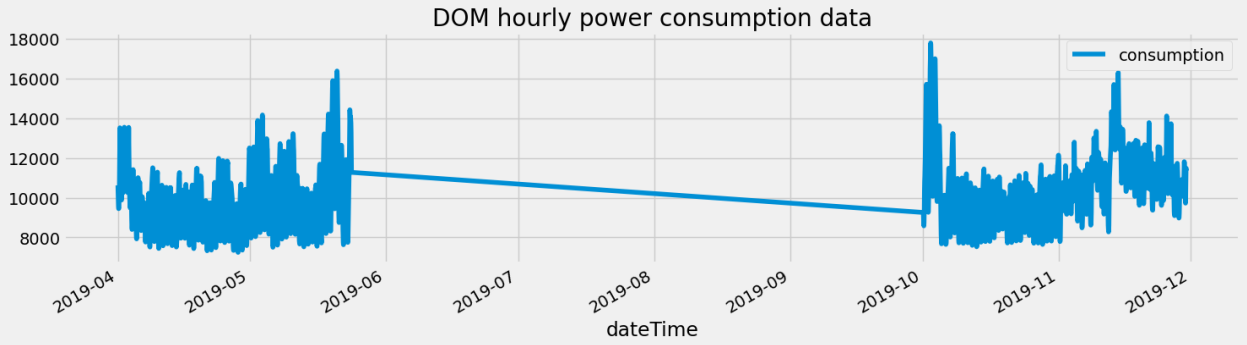

Figure 3 – Missing DOM data in June and Q2 2019

For study purposes, we decided to choose the data of the region named "**AP**" instead of imputing the missing data of the DOM region mentioned above.

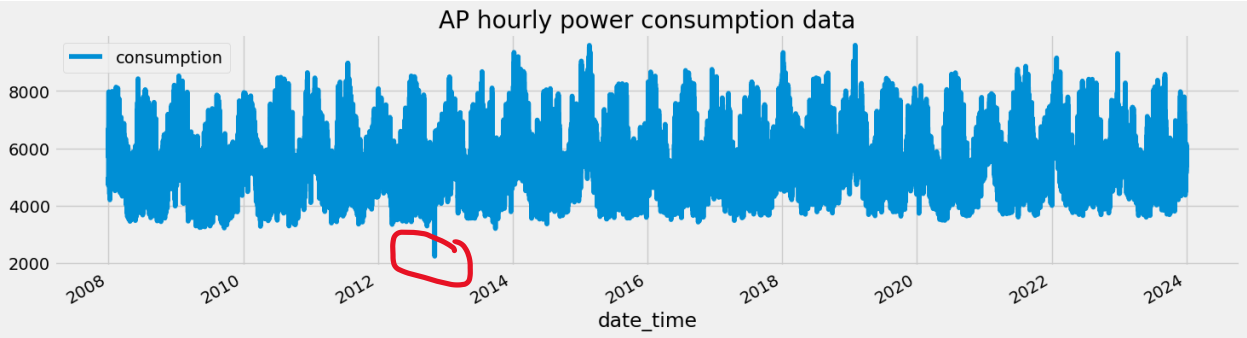

Figure 4 – The AP region data

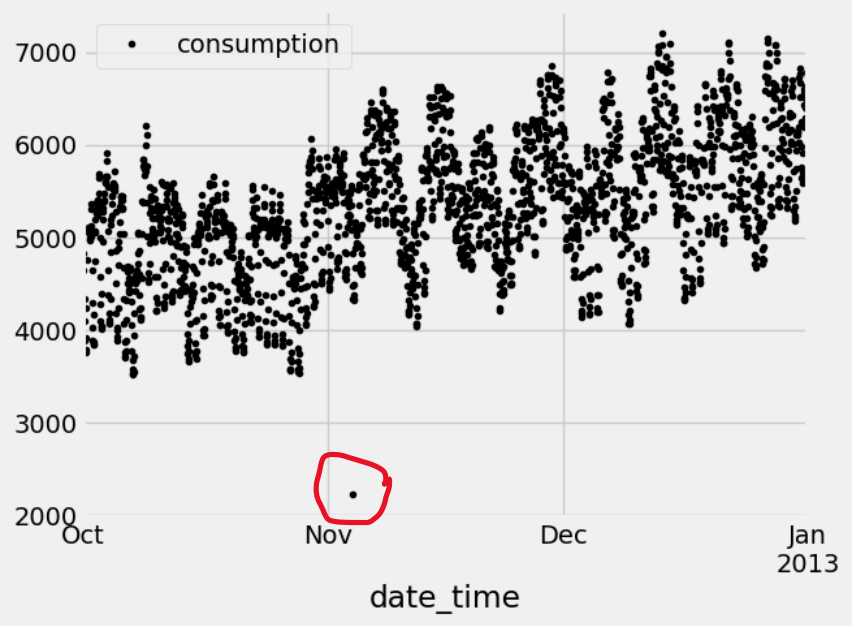

Figure 5 – The outlier data of the AP region

For the AP region, we found an outlier shown in figures 4 and 5. Also, when checking the data column **"date_time_ept"**, we found that there are 32 duplicates, 1 data row missing each year (16 data rows missing in total), and 672 data rows missing in each February each year (see Figures 6, the Excel files named duplicate.xlsx, and missing_data.xlsx in the Output folder of the project code for more details).

date_time_utc	date_time_ept	consumption	date_time
11/2/2008 5:00	11/2/2008 1:00	3693.066	2008-11-02 01:00:00
11/2/2008 6:00	11/2/2008 1:00	3748.793	2008-11-02 01:00:00
11/1/2009 5:00	11/1/2009 1:00	3819.498	2009-11-01 01:00:00
11/1/2009 6:00	11/1/2009 1:00	3623.467	2009-11-01 01:00:00
11/7/2010 5:00	11/7/2010 1:00	4688.179	2010-11-07 01:00:00
11/7/2010 6:00	11/7/2010 1:00	4488.572	2010-11-07 01:00:00
11/6/2011 5:00	11/6/2011 1:00	4575.514	2011-11-06 01:00:00
11/6/2011 6:00	11/6/2011 1:00	4537.101	2011-11-06 01:00:00
11/4/2012 5:00	11/4/2012 1:00	4477.774	2012-11-04 01:00:00
11/4/2012 6:00	11/4/2012 1:00	4333.208	2012-11-04 01:00:00
11/3/2013 5:00	11/3/2013 1:00	4071.842	2013-11-03 01:00:00
11/3/2013 6:00	11/3/2013 1:00	4034.939	2013-11-03 01:00:00
11/2/2014 5:00	11/2/2014 1:00	4571.107	2014-11-02 01:00:00
11/2/2014 6:00	11/2/2014 1:00	4612.613	2014-11-02 01:00:00
11/1/2015 5:00	11/1/2015 1:00	3926.62	2015-11-01 01:00:00
11/1/2015 6:00	11/1/2015 1:00	3847.953	2015-11-01 01:00:00
11/6/2016 5:00	11/6/2016 1:00	4088.621	2016-11-06 01:00:00
11/6/2016 6:00	11/6/2016 1:00	4113.775	2016-11-06 01:00:00
11/5/2017 5:00	11/5/2017 1:00	4041.907	2017-11-05 01:00:00
11/5/2017 6:00	11/5/2017 1:00	3983.532	2017-11-05 01:00:00
11/4/2018 5:00	11/4/2018 1:00	4752.774	2018-11-04 01:00:00
11/4/2018 6:00	11/4/2018 1:00	4572.838	2018-11-04 01:00:00
11/3/2019 5:00	11/3/2019 1:00	4678.695	2019-11-03 01:00:00
11/3/2019 6:00	11/3/2019 1:00	4636.395	2019-11-03 01:00:00
11/1/2020 5:00	11/1/2020 1:00	4586.007	2020-11-01 01:00:00

year_month_day	frequency	no. missing rows
2008_03_09	23	1
2009_03_08	23	1
2010_03_14	23	1
2011_03_13	23	1
2012_03_11	23	1
2013_03_10	23	1
2014_03_09	23	1
2015_03_08	23	1
2016_03_13	23	1
2017_03_12	23	1
2018_03_11	23	1
2019_03_10	23	1
2020_03_08	23	1
2021_03_14	23	1
2022_03_13	23	1
2023_03_12	23	1
	Total	16

year_month	frequency	no. missing rows
2008_02	696	24
2009_02	672	48
2010_02	672	48
2011_02	672	48
2012_02	696	24
2013_02	672	48
2014_02	672	48
2015_02	672	48
2016_02	696	24
2017_02	672	48
2018_02	672	48
2019_02	672	48
2020_02	696	24
2021_02	672	48
2022_02	672	48
2023_02	672	48
	Total	672

Figure 6 – Duplicates and missing data

However, we saw patterns when taking a closer look at the duplicates and missing data, and it turns out that they are not duplicates and missing data at all. For the duplicates (the left table), this is due to the daylight-saving time adjustments in November at 1 AM on a specific day each year, so there are two data rows with the same datetime values on these days. For the data missing in the right table, this is because there are only 28 or 29 days in February each year, they cannot be seen as missing cases. For the data missing in the middle table, this happens on the first or second week of March each year. There must be a special event such as Earth Hours or system maintenance because data is lost at a specific hour on these days. We suppose that they are not missing data as well.

For the reasons mentioned above, we decided not to impute these missing data and go ahead to the “Model Planning” step, skipping the “Data Preparation” one.

4. Model Planning

We started with the full data set in 16 years, from 2008 to 2023. The seasonal decomposition plot suggests that there is no trend in the data. Furthermore, the seasonal component differs from the others since it only shows a rectangle with no further information.

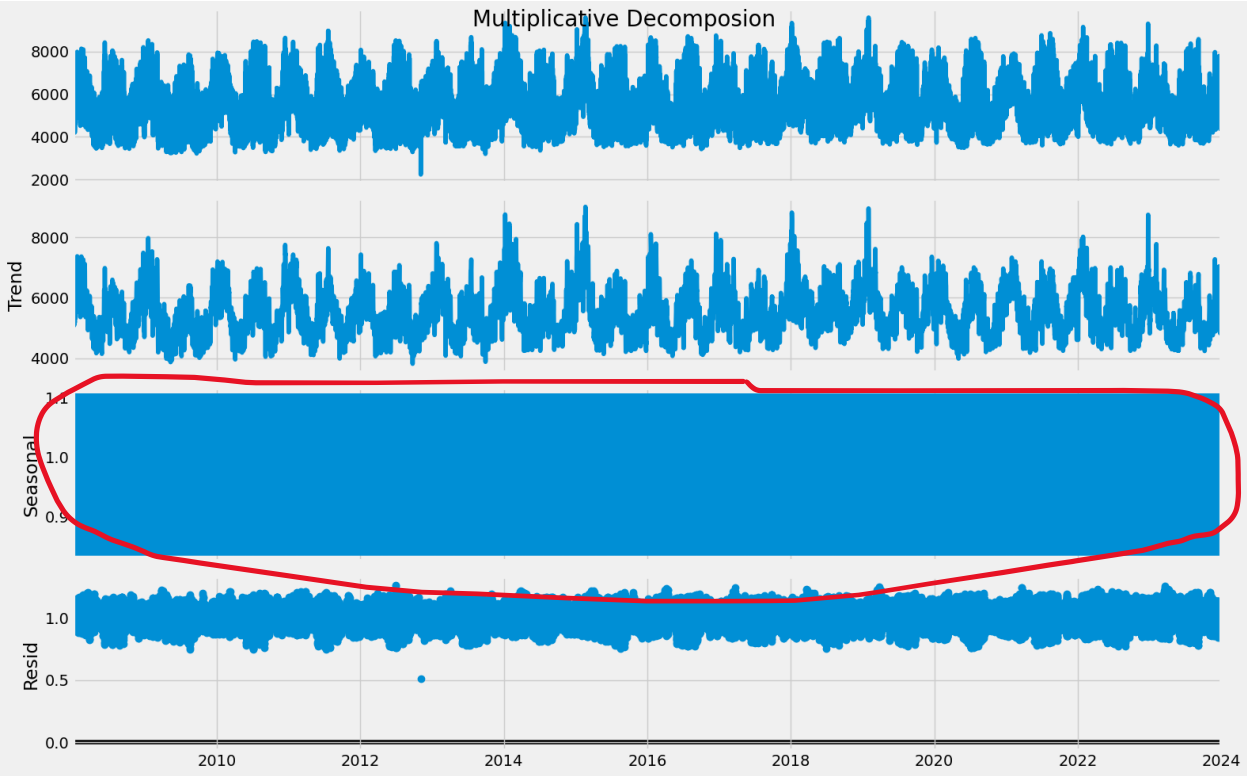

Figure 7 – Time series decomposition

We need to check if the time series is stationary or not by using the augmented Dickey-Fuller (ADF) test, in which the null and alternative hypothesis of the test is[6]:

Null Hypothesis: The series has a unit root (or the series is not stationary)

Alternate Hypothesis: The series has no unit root (or the series is stationary)

Below is the test result, in which the P-value is 0, less than 0.05. We can reject the null hypothesis, meaning that the time series is stationary already.

ADF (Augmented Dickey Fuller) Statistic: -19.56771155088631
p-value: **0.0**
Critical Values:
1%, -3.4303966498295004
Critical Values:
5%, -2.861560618558905
Critical Values:
10%, -2.5667809745260652

[6] https://www.kdnuggets.com/2020/01/predict-electricity-consumption-time-series-analysis.html

After splitting data into training and test sets with the ratio of 80% and 20% respectively, we go through all the next stages in the time series forecasting, including finding the best model, fitting the model, and finally predicting the model. However, because there are too many data rows (more than 14.000), we cannot automatically select the best model by using a grid search or stepwise approach. It crashed each time we ran our code to find the best model using the Google Colab service. After trying to manually fit the mode with 15 combinations of the hyperparameter set (p, d, q) (P, D, Q) m, we found the best model SARIMAX (2, 0, 1) $(1, 0, 1)_{24}$. We are not sure if it is the best model or not but continue to predict the data on the test set with this best model. Below is the plot showing the predicted result for the test set. Most of the predicted data are totally under the actual values, which means that the model is too bad to go ahead (Figure 8).

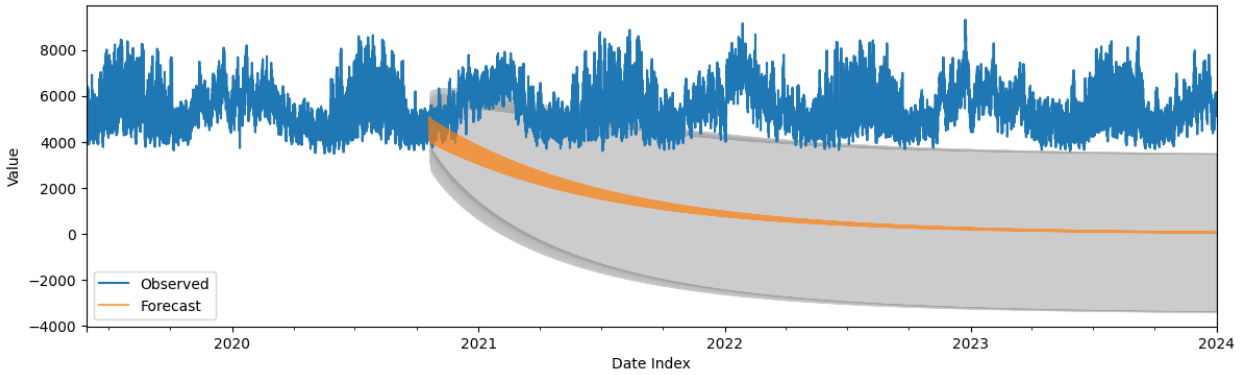

Figure 8 – Forecast and Actual with 95% significance level boundaries

For that reason, we decided to reduce the size of the data to only two months, from November to December 2023, which includes 1469 hourly data rows. Doing this will not only allow us to visualize data more clearly, thereby analyzing data more accurately, but also help us choose the best model using an automatic approach. Below are the details of each step in the time series forecasting using the 2-month data.

Visualizing the time series

As we analyzed before, it seems there is no trend in the data and the distribution is a near-normal distribution (see Figures 9 and 10).

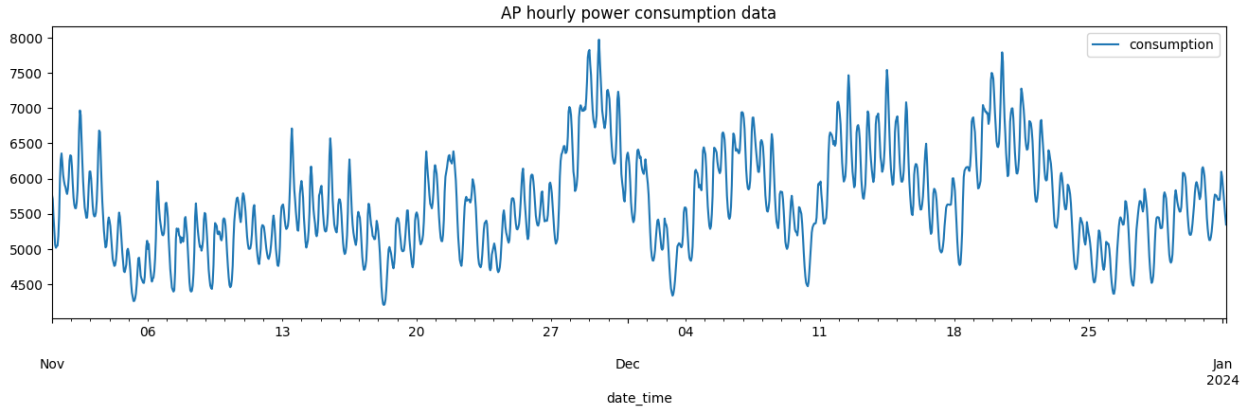

Figure 9 – Original time series data

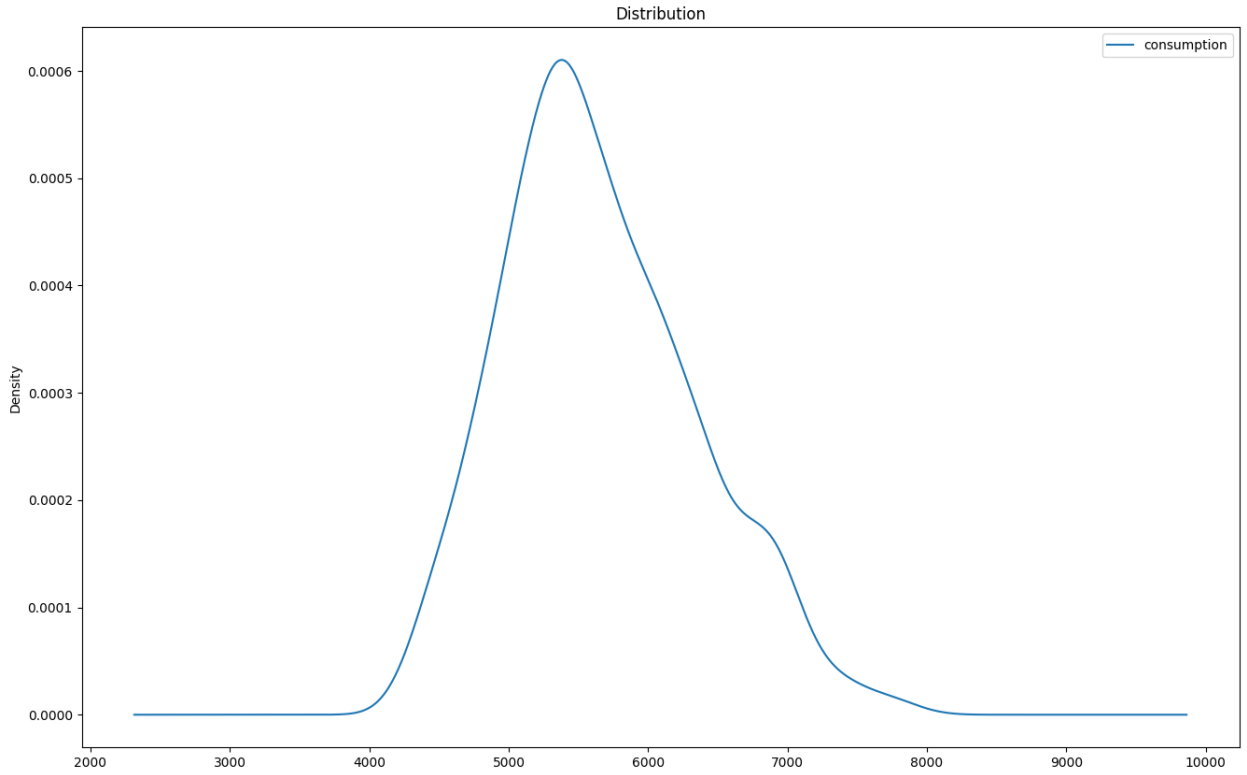

Figure 10 – The consumption distribution

Continue to decompose the time series data to see more details. Clearly, there is no trend but a seasonal pattern in our data (Figure 11).

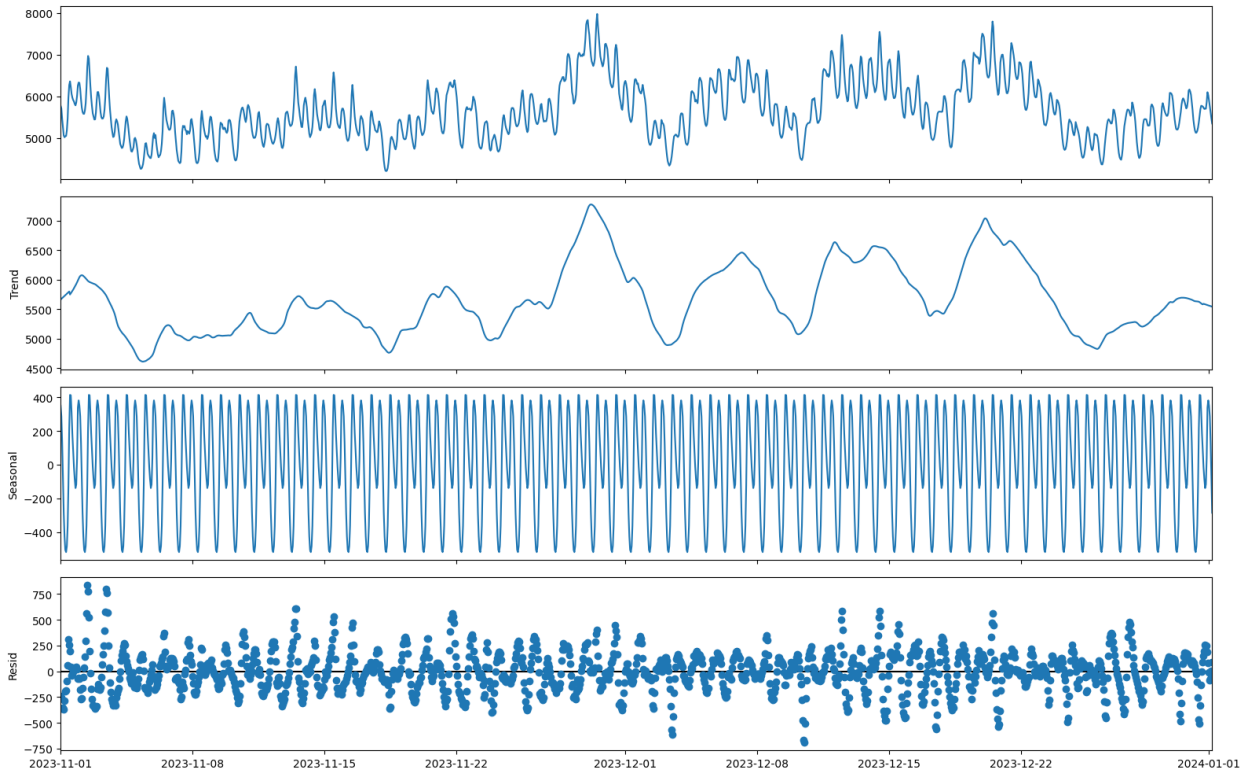

Figure 11 – Seasonal decomposition plot

Let us check if the time series is stationary or not by using the ADF test mentioned above.

```
ADF (Augmented Dickey Fuller) Statistic: -5.574091859424268
p-value: 1.4452988953541691e-06
Critical Values:
   1%, -3.434886677803751
Critical Values:
   5%, -2.8635436366589673
Critical Values:
   10%, -2.5678367211155533
```

The P-value is less than 0.05, we can reject the null hypothesis, meaning that the time series is stationary. We will go ahead with the next steps without making the time series stationary.

Finding the best parameters for the model

We need to look at the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots, which allows us to have an overview of the hyperparameters p, d, q of the ARIMA model. In our case, d is 0 since the time series is already stationary.

The ACF plot shows many significant ACF values (Figure 12), and they slowly decay at lags 12, 24, 36, and 48. This signal indicates a seasonal autoregressive pattern every 12 hours. Furthermore, the PACF values at lags 1, 2, 3, and some other lags after that are quite large, but we do not see a seasonal pattern here. Thus, we can say that a seasonal MA (**1**) model with a period of 12 will be considered. In other words, this is a SARIMA model with a seasonal period of 12, not ARIMA as planned to apply at the beginning of the project.

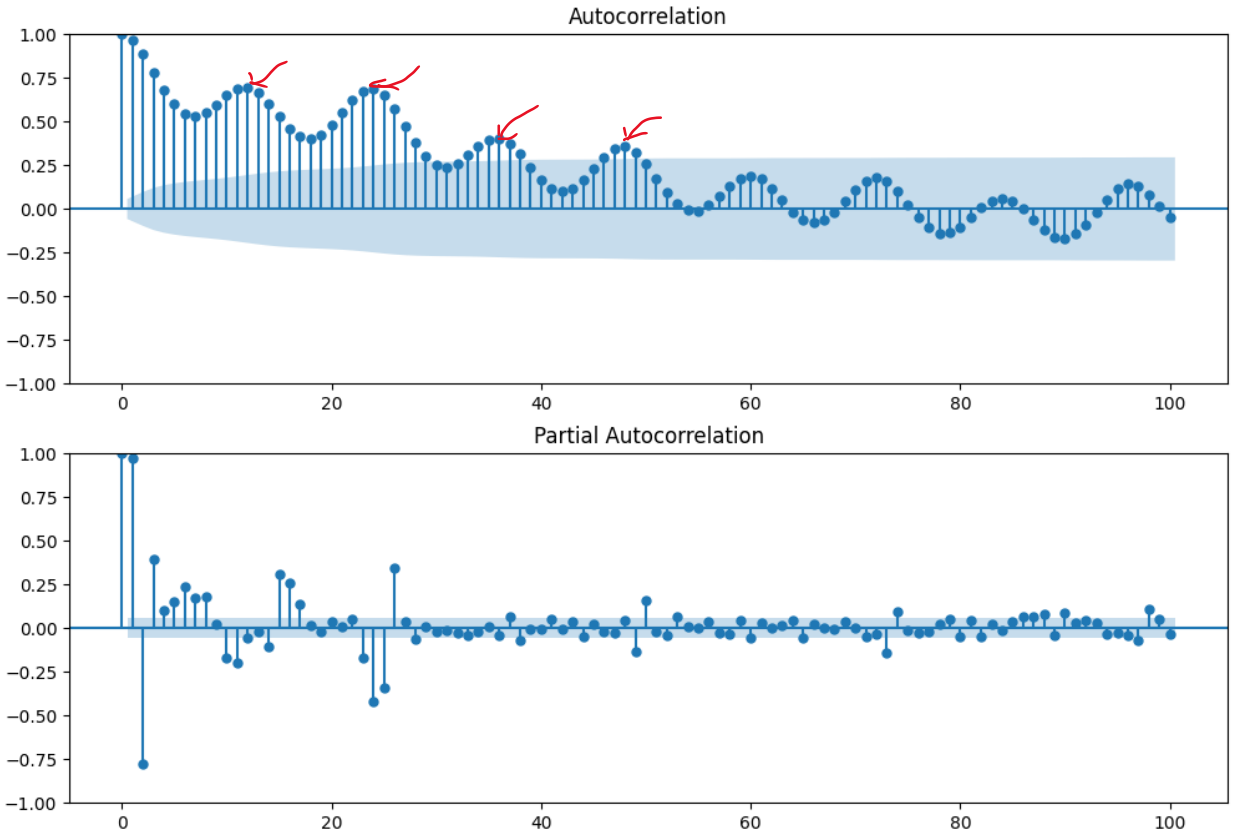

Figure 12 – ACP and PACF plots

Let perform a stepwise search to find the best model with the minimum Akaike information criterion (ACI). The best model found is SARIMAX (3, 0, 0) (2, 0, 1)$_{12}$.

```
 ARIMA(4,0,0)(2,0,1)[12] intercept   : AIC=13401.269, Time=10.41 sec
 ARIMA(3,0,1)(2,0,1)[12] intercept   : AIC=inf, Time=11.01 sec
 ARIMA(2,0,1)(2,0,1)[12] intercept   : AIC=13407.218, Time=8.28 sec
 ARIMA(4,0,1)(2,0,1)[12] intercept   : AIC=inf, Time=12.31 sec
 ARIMA(3,0,0)(2,0,1)[12]             : AIC=13365.903, Time=4.54 sec

Best model:  ARIMA(3,0,0)(2,0,1)[12] intercept
Total fit time: 188.654 seconds
```

SARIMAX Results

Dep. Variable:	y	No. Observations:	1175
Model:	SARIMAX(3, 0, 0)x(2, 0, [1], 12)	Log Likelihood	-6671.098
Date:	Sat, 30 Mar 2024	AIC	13358.196
Time:	00:24:19	BIC	13398.748
Sample:	11-01-2023	HQIC	13373.488
	- 12-19-2023		

Figure 13 – The best SARIMA model

Fitting the model

Let us fit the model using the training set with the best SARIMA model found above. We have the residual plots of the model as shown in Figure 14.

From the residual error plot (on the top left), it seems to fluctuate around a mean of zero and have a uniform variance. The density plot from the top right suggests a normal distribution with a zero mean. In the bottom right plot, most of the dots fall in line with the red line, except for several dots at the beginning and end of the red line. Finally, the bottom right plot indicates that no further terms needed to be considered in the SARIMA model, which means the residual errors are not autocorrelated. In conclusion, the model seems to be a good fit.

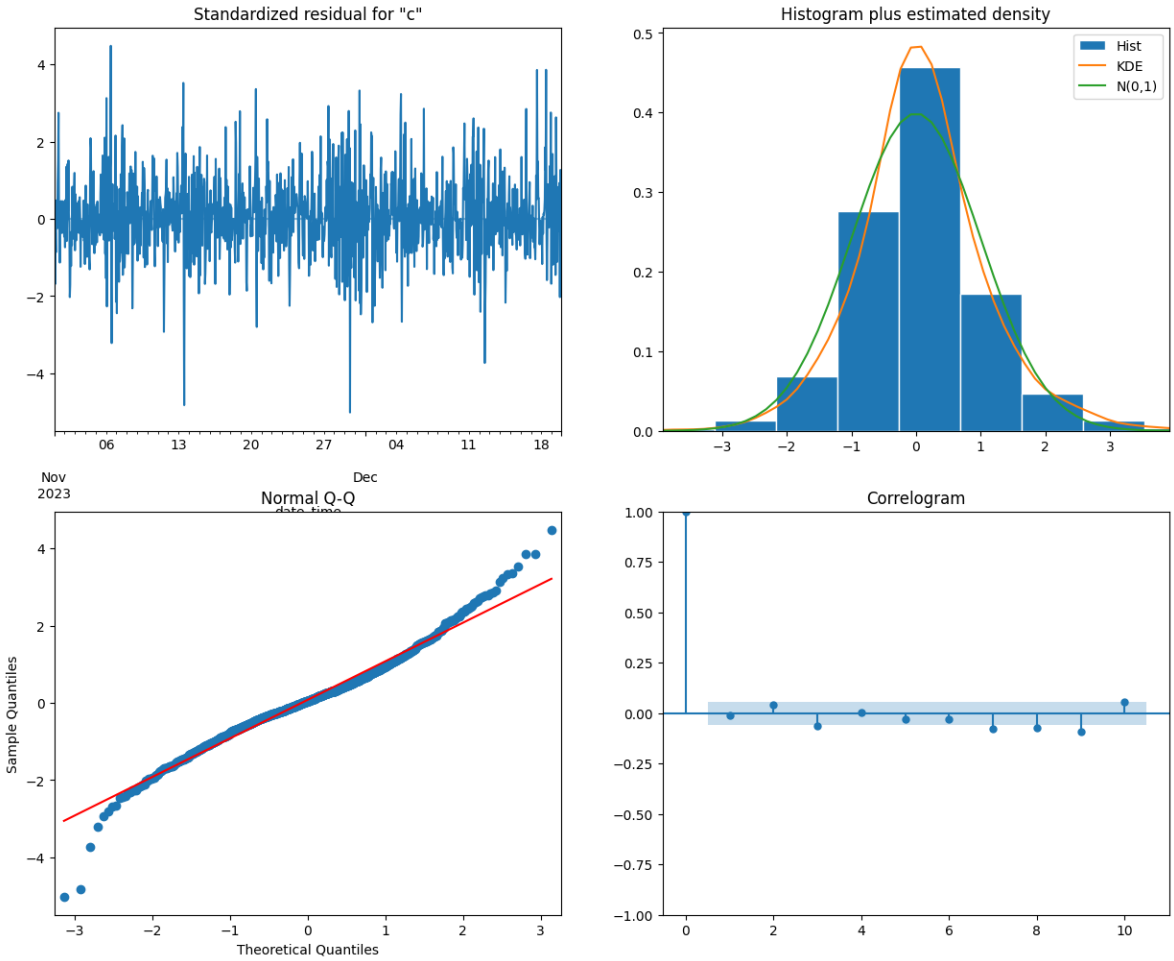

Figure 14 – Residual charts

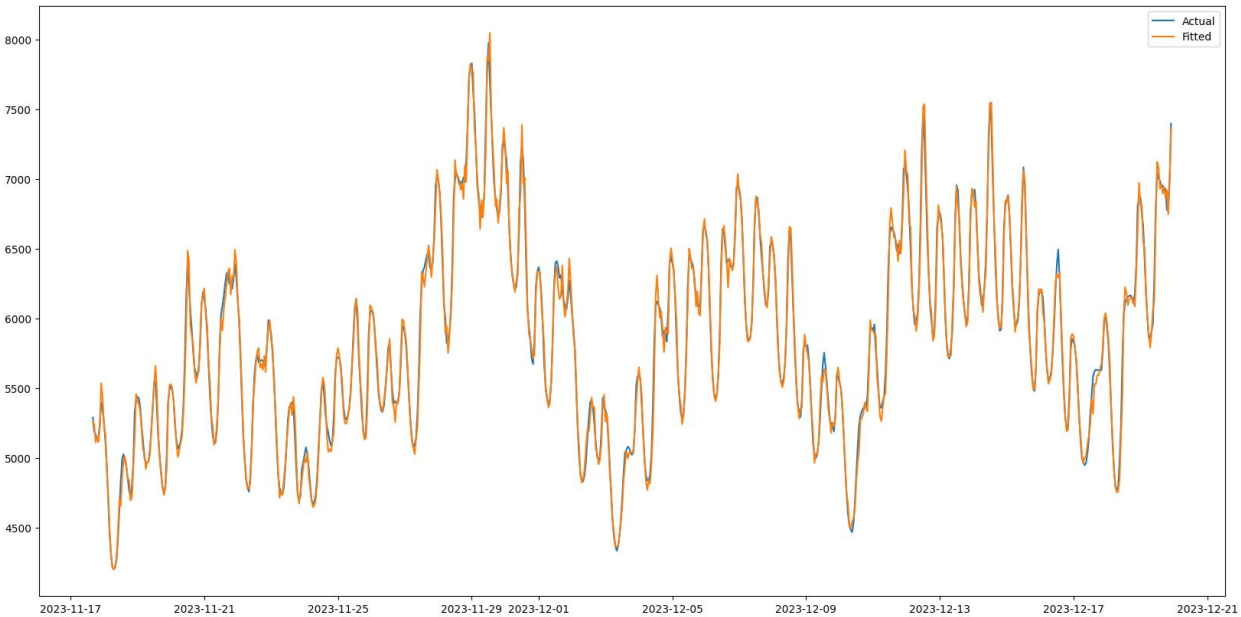

Figure 15 – Actual and fitted values

From Figure 15 above, which plots the actual and fitted values, we can see they are not much different. However, when looking at Figure 16, which visualizes the training, test, and forecasted values with the boundaries of a 95% confidence level, we can see that the model correctly predicted the downward trend in half of the test set but incorrectly for the rest. This is understandable because our model is a seasonal model with a period of 12 hours. It is better to predict data in the next 12 or 24 hours, not too far ahead.

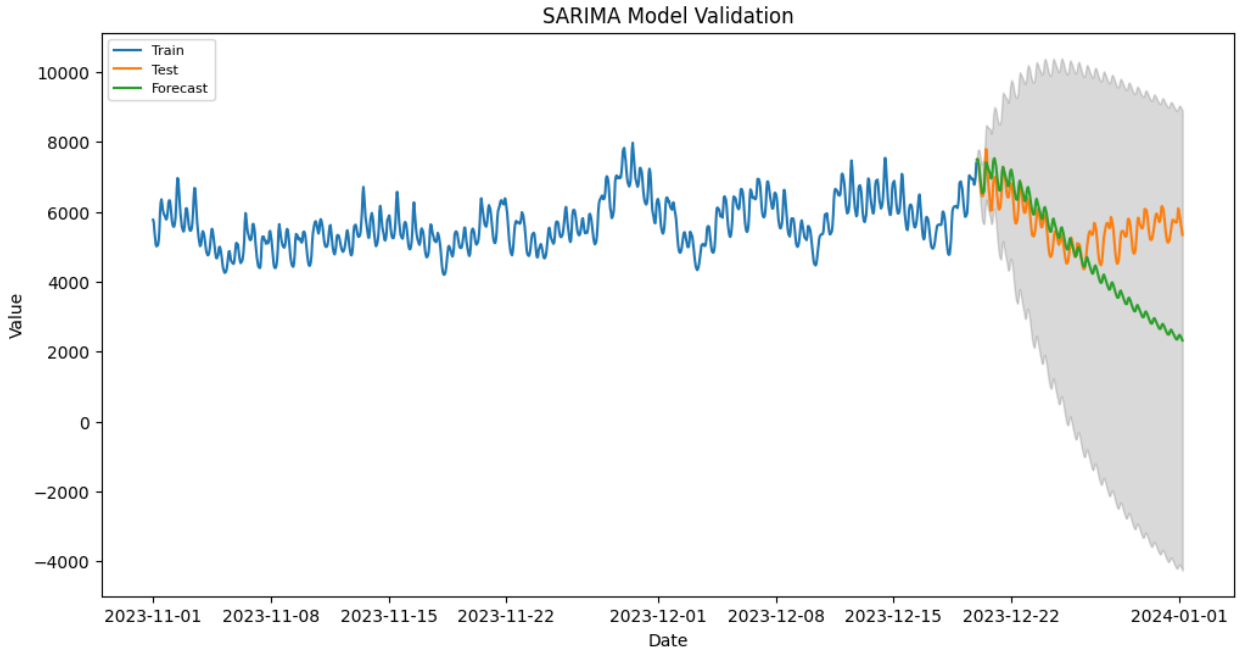

Figure 16 – SARIMA model validation

Finally, we should look at the accuracy metrics to validate our model's performance. The mean absolute percentage error (MAPE)[7] is 0.2087, meaning that the model predicted with about 79.13% accuracy. This ratio is not as good as our expectations, but we have good signals from analyzing some other plots mentioned above. Now we can go ahead with our predictions.

```
Accuracy metrics:
MAPE: 0.208750
MAE: 1163.182718
MSE: 2537408.273549
RMSE: 1592.924441
```

[7] https://machinelearningmastery.com/time-series-forecasting-performance-measures-with-python/

5. Model Building

Let's build the best model, SARIMA (3, 0, 0) (2, 0, 1)$_{12}$, on the whole data set. The model has significant P-values, except for the last one (Figure 17).

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                         consumption   No. Observations:                 1469
Model:             SARIMAX(3, 0, 0)x(2, 0, [1], 12)   Log Likelihood               -8283.930
Date:                              Sun, 31 Mar 2024   AIC                          16581.861
Time:                                      03:20:21   BIC                          16618.907
Sample:                                11-01-2023   HQIC                         16595.676
                                     - 01-01-2024
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.6762      0.019     86.542      0.000       1.638       1.714
ar.L2         -0.7736      0.034    -22.955      0.000      -0.840      -0.708
ar.L3          0.0941      0.019      4.929      0.000       0.057       0.131
ar.S.L12       0.0440      0.020      2.157      0.031       0.004       0.084
ar.S.L24       0.7604      0.014     53.695      0.000       0.733       0.788
ma.S.L12       0.0267      0.030      0.881      0.378      -0.033       0.086
sigma2      4439.6388    111.804     39.709      0.000    4220.506    4658.771
===================================================================================
Ljung-Box (L1) (Q):                   0.61   Jarque-Bera (JB):               378.40
Prob(Q):                              0.43   Prob(JB):                         0.00
Heteroskedasticity (H):               0.80   Skew:                            -0.02
Prob(H) (two-sided):                  0.02   Kurtosis:                         5.49
===================================================================================
```

Figure 17 – Build the SARIMA (3, 0, 0) (2, 0, 1)$_{12}$ model

Predictions

Let us forecast for the next 24 hours. We can see that the model works better when predicting data for the next 24 hours. It correctly predicts the seasonal pattern of the data, with a confidence level of 95% (see Figure 18).

Furthermore, when looking at the accuracy metrics below, the MAPE now is 0.069, meaning that our model predicts about **93.10%** accuracy for the next 24 hours. This ratio is significantly higher than that when validating the model for the test set.

```
Accuracy metrics:
MAPE: 0.069017
MAE: 421.792950
MSE: 260382.148474
RMSE: 510.276541
```

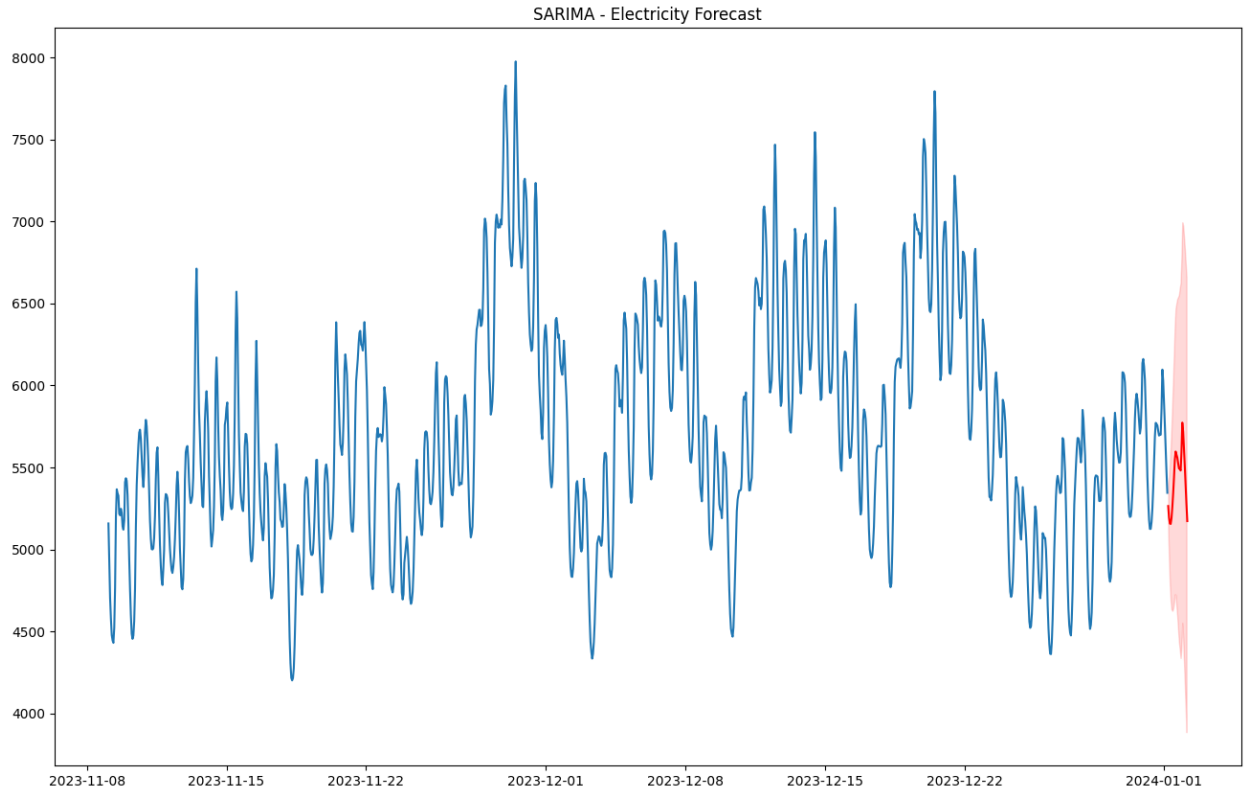

Figure 18 – SARIMA electricity consumption forecasting result for the next 24 hours

6. Conclusions

The model predicts electricity consumption for the next 24 hours with an accuracy of nearly 93.10% (MAPE = 0.069), higher than we expected in the project proposal (85–90%). The time series data has a seasonal component with a period of 12 hours. We believe that the model will work better to forecast electricity usage in the next 12 or 24 hours. When trying to predict the data for the next 48 and 72 hours (see Figures 19, 20, respectively), its accuracy is reduced to 87% and 83%, respectively, which is still a high ratio. However, we suppose that the forecasting period should not be more than 48 hours (2 days), which is good enough for power companies to prepare for their supply management to keep up with demand. In the future, if they want to predict

electricity consumption for a long period of time, we suggest using time series data in other time intervals, such as days or months.

```
Accuracy metrics:
MAPE: 0.131716
MAE: 846.669902
MSE: 1019985.102453
RMSE: 1009.943118
```

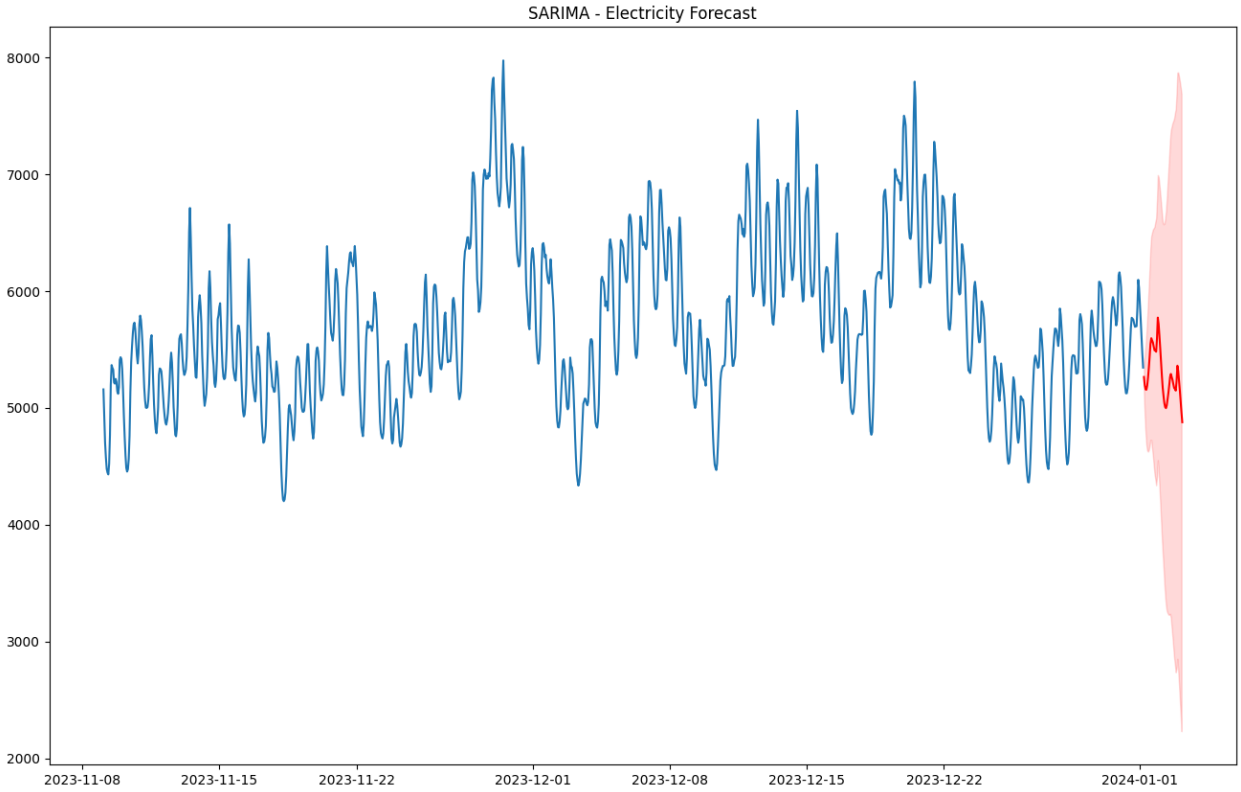

Figure 19 - SARIMA electricity consumption forecasting result for the next 48 hours

```
Accuracy metrics:
MAPE: 0.172002
MAE: 1107.230898
MSE: 1592247.967342
RMSE: 1261.843083
```

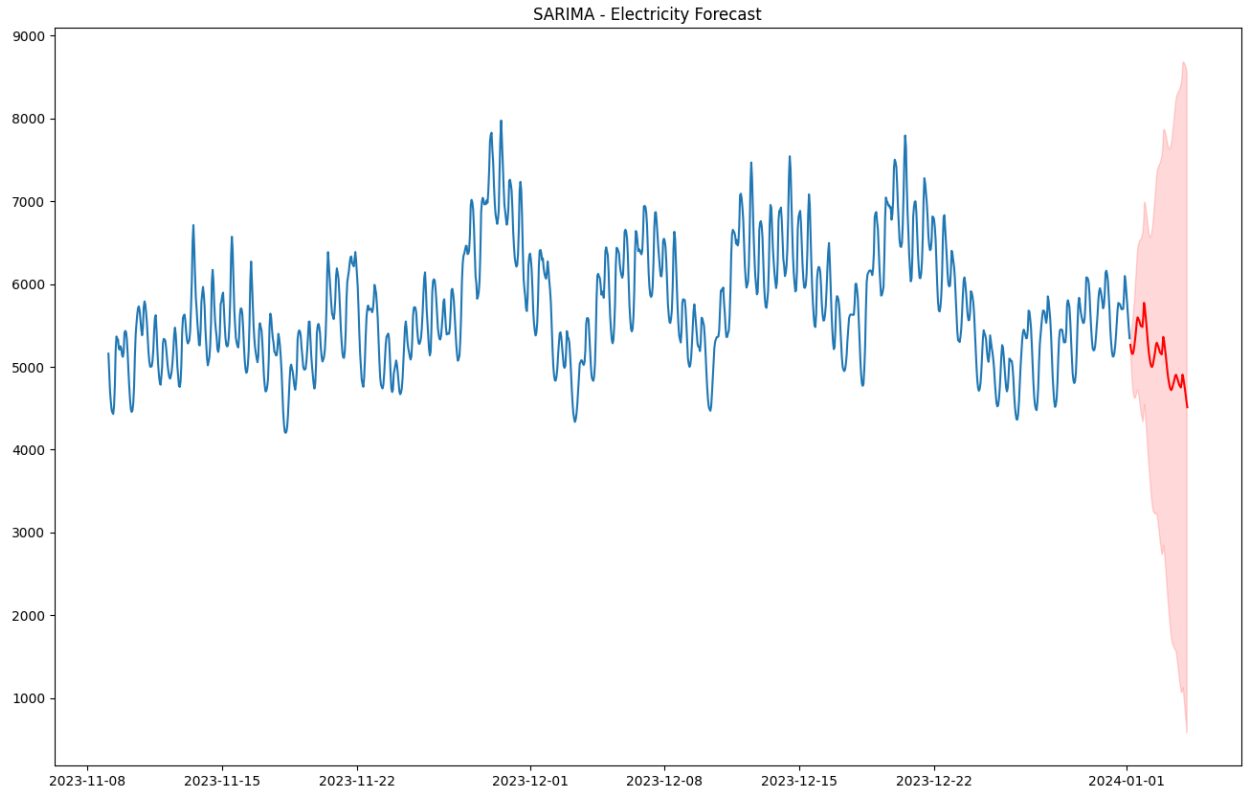

Figure 20 - SARIMA electricity consumption forecasting result for the next 72 hours

Lesson learned from the project

Going through the whole data science project like this brings us lots of useful lessons to learn from. They include:

- **The important of data visualization tools**

Data visualization is a critical step to getting insight into data. At the beginning of the project, we did some initial checks and found that our data was of high quality. However, when plotting data using visualization tools, it turned out that the data was not as good as our thinking. According to

Anscombe's Quartet[8], data visualization plays an important part before starting to analyze data. This is a critical point that we will keep in mind when doing any data analysis tasks in the future.

- **The nature of the data when choosing an appropriate machine learning algorithm**

It is clear that we were led by the nature of the data when doing this project. Firstly, we plan to use the ARIMA model, but then it turns out to be the ARMA one since the time series is already stationary. Finally, we end up with the SARIMA model because the time series exhibits a seasonal pattern with a period of 12 hours. This lesson emphasizes what we have learned throughout this course, which mentions that we should choose a machine learning algorithm based on the nature of the data.

- **Choosing the right size of data set**

We are not sure about the accuracy of this lesson, but we have a sense that there is no need to use the time series data for many years to only predict data in the next few hours or days. When it comes to time series data, let's look at the nature of the data, especially its seasonal pattern (if any), we will find the right size of the data to be used to build our model, avoiding wasting time with unexpected results.

7. References

Course book:

EMC^2 – Data Science and Big Data Analytics – Discovering, Analyzing, Visualizing and Presenting Data

Previous/similar projects or studies:

https://www.kdnuggets.com/2020/01/predict-electricity-consumption-time-series-analysis.html

https://www.kaggle.com/code/nageshsingh/predict-electricity-consumption/notebook

https://www.kaggle.com/code/msripooja/hourly-energy-consumption-time-series-rnn-lstm/notebook

[8] https://en.wikipedia.org/wiki/Anscombe%27s_quartet

https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/

https://medium.com/@tirthamutha/time-series-forecasting-using-sarima-in-python-8b75cd3366f2