Time Series Forecasting

Done By: Toh Kien Yu (P2222291)

Background Research
I will be forecasting Gas, Electricity and Water Consumption for this project.

1. Forecasting gas consumption is essential for setting up and running of a sustainable system.
2. Forecasting electricity is important is crucial as it serves as a foundation to make decisions in power sector planning.
3. Forecasting water consumption is also important to ensure people do not face service interruptions are able to maintain an affordable rate while being dependable.

Overall, forecasting allows us to better plan and make decisions when managing our resource.

Nature of Dataset

<u>Numeric Data</u>

1. Gas Consumption (tons)
2. Electricity Consumption (MWh)
3. Water Consumption (tons)

<u>Categorical Data</u>

1. DATE

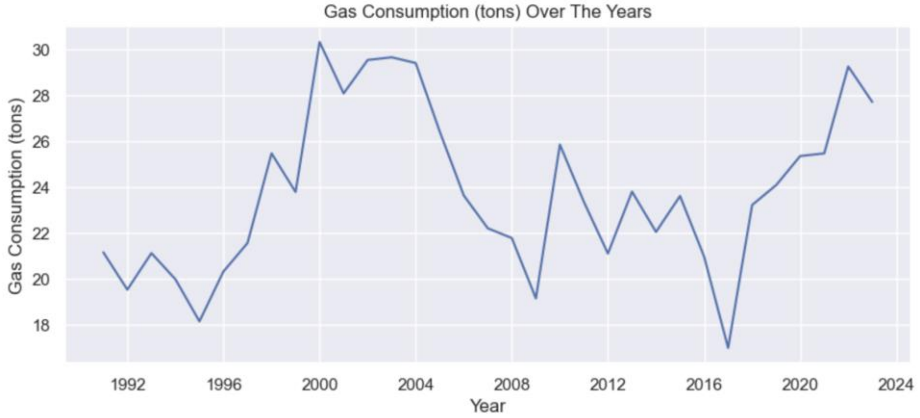
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 397 entries, 1990-01-01 to 2023-01-01
Data columns (total 3 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Gas Consumption (tons)         397 non-null    float64
 1   Electricity Consumption (MWh)  397 non-null    float64
 2   Water Consumption (tons)       397 non-null    float64
dtypes: float64(3)
memory usage: 12.4 KB
```

```
# No Null Values
df.isnull().sum()
```

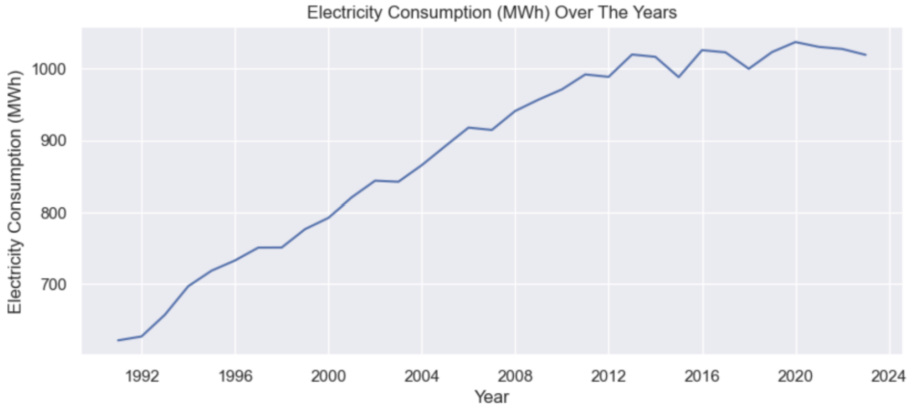
```
Gas Consumption (tons)           0
Electricity Consumption (MWh)    0
Water Consumption (tons)         0
dtype: int64
```

```
Dataset Shape
(397, 3)
```

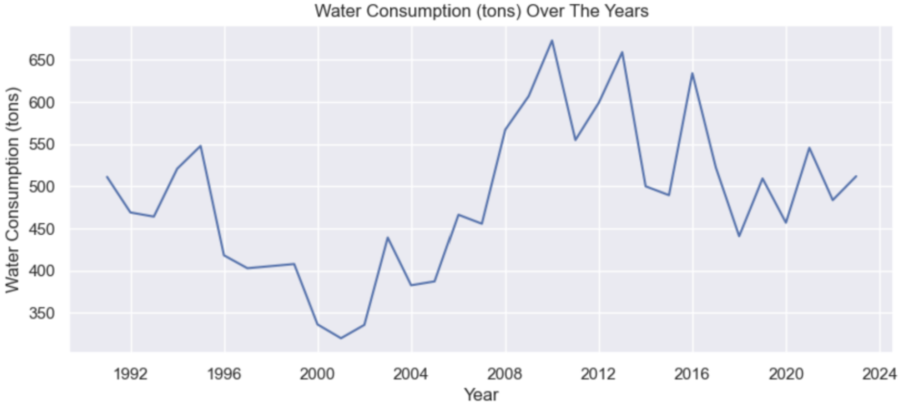
Data Exploration



There is an increase of about 7 tons of gas consumption over the years with fluctuations occurring during this period.

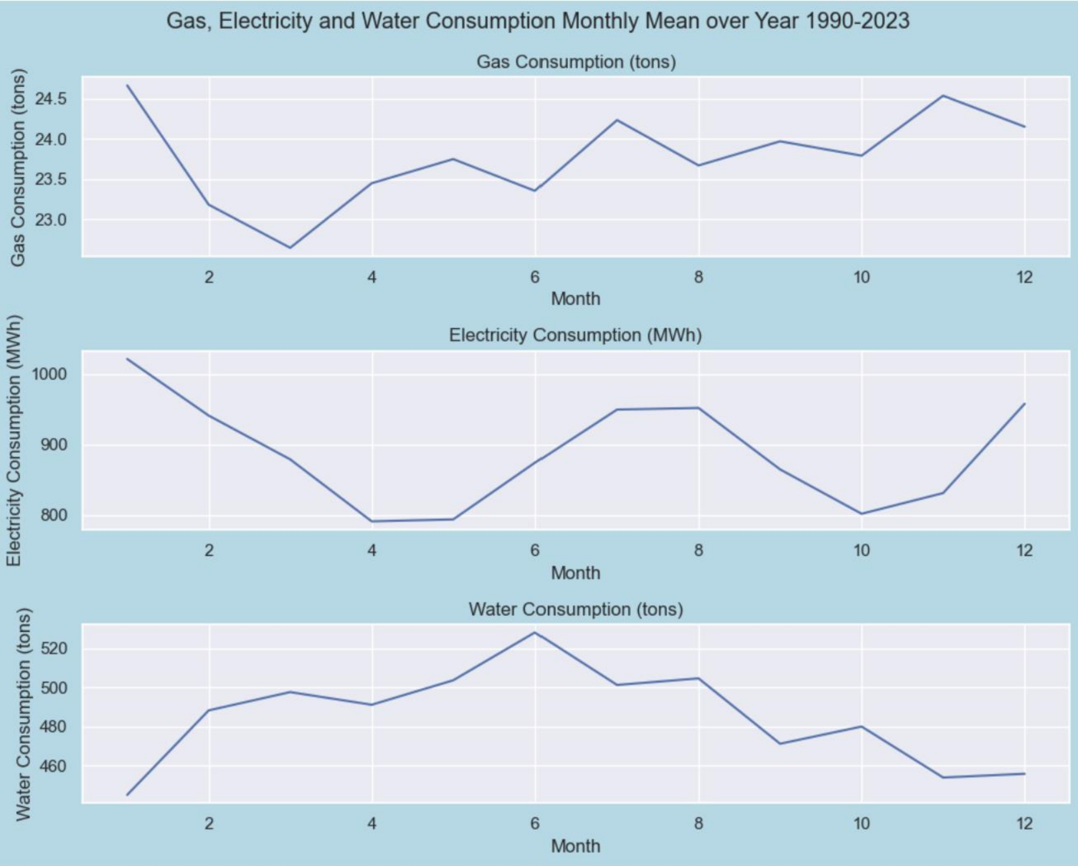


Electricity Consumption has been increasing over the years which can be possibly attributed to the global population increase.



While there is a general fall in water consumption, fluctuations occurred greatly where water consumption peaked at 675 tons in 2010.

Data Exploration



From the graph above,

1. Gas Consumption and Electricity Consumption tend to peak in January
2. Water Consumption is highest in June.

From the graph above, it provides data in recent years 2021 and 2022. Electricity and Water Consumption usually peaks in August

Stationarity

H0: The time series can be represented by a unit root, that is not stationary.
H1:The time series is stationary

```
# Augmented Dickey-Fuller Test
from statsmodels.tsa.stattools import adfuller
result1 = adfuller(df['Gas Consumption (tons)'])
result2 = adfuller(df['Electricity Consumption (MWh)'])
result3 = adfuller(df['Water Consumption (tons)'])
print('Gas Consumption (tons) p-value: %f' % result1[1])
print('Electricity Consumption (MWh) p-value: %f' % result2[1])
print('Water Consumption (tons) p-value: %f' % result3[1])
```

```
Gas Consumption (tons) p-value: 0.010811
Electricity Consumption (MWh) p-value: 0.186218
Water Consumption (tons) p-value: 0.000090
```

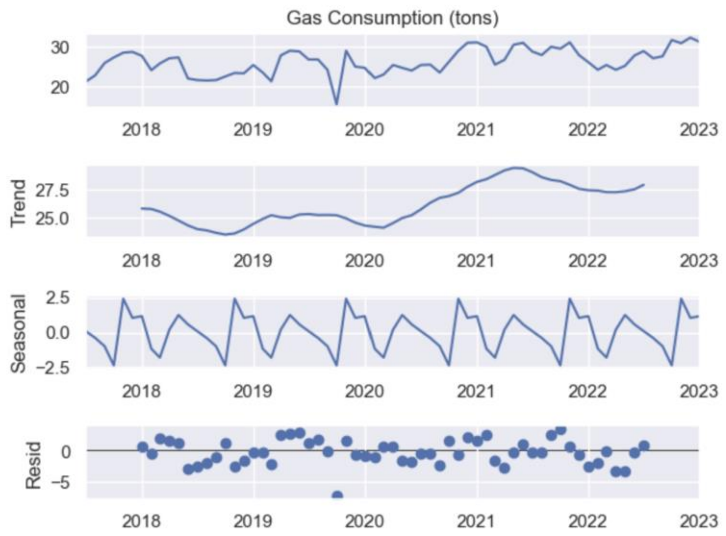
- Gas Consumption (tons) p value is 0.010811 which is < 0.05. Reject the null hypothesis (H0), the data does not have unit root and is stationary

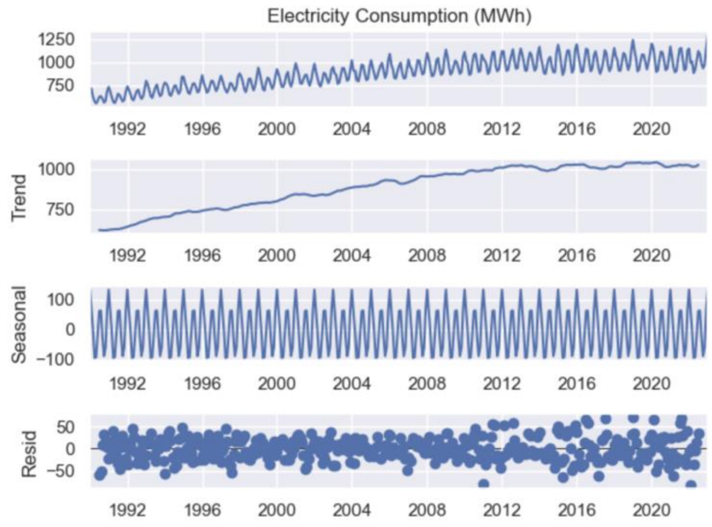
- Electricity Consumption (MWh) p value is 0.186218 which is > 0.05. Fail the reject the null hypothesis (H0), the data has a unit root and is non-stationary.

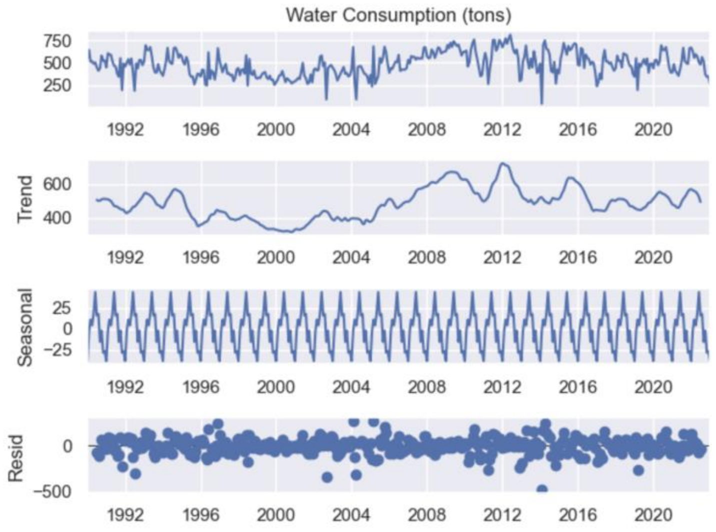
- Water Consumption (tons) p value is 0.000090 which is < 0.05. Reject the null hypothesis (H0), the data does not have unit root and is stationary

Electricity Consumption is differenced in order to make the time series stationary.

Seasonal Decomposition







- Seasonal Period observed every 12 months

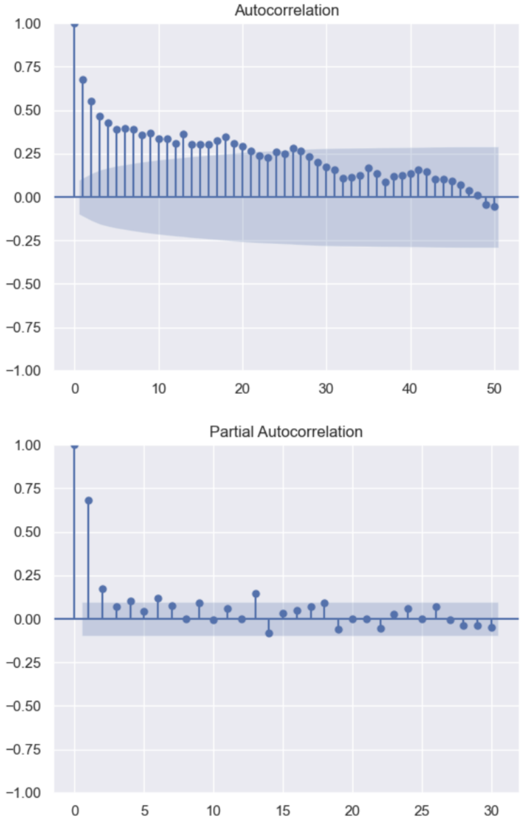
ACF and PACF Plots

```
hDF = df.copy()
train_data = hDF[hDF.index<'2019-11']
test_data = hDF[hDF.index>='2019-11']
(len(train_data)/(len(test_data) + len(train_data))) * 100
```

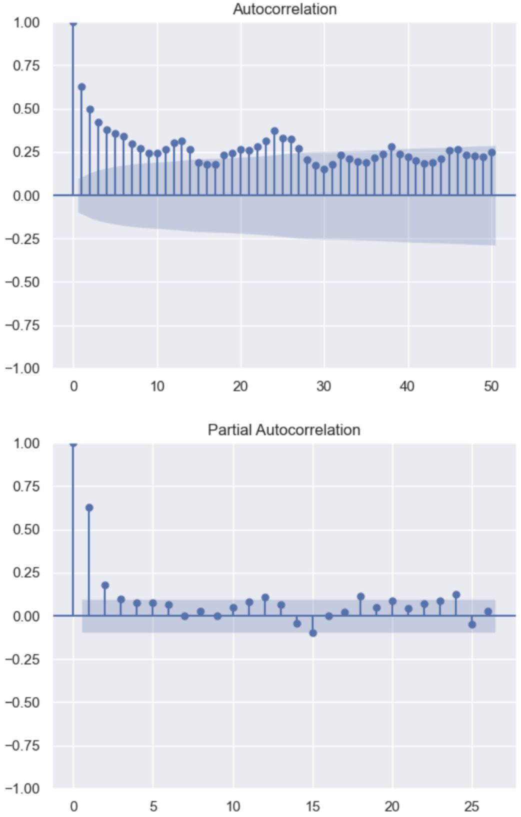
```
90.17632241813602
```

An 90:10 ratio between training and test data is chosen where data before Year 2019-11 is used for training and data from Year 2019-11 is used for testing

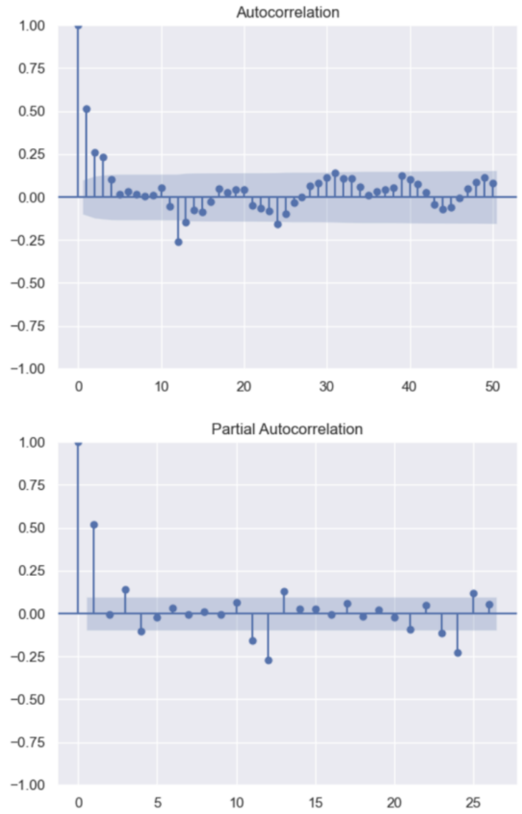
Gas Consumption (tons)



Water Consumption (tons)



Electricity Usage (MWh)



Gas Consumption and Water Consumption: Tail off at ACF Plot, which indicates an AR model. From PACF plot, cut off happens at lag 2. This illustrates an AR(2) model

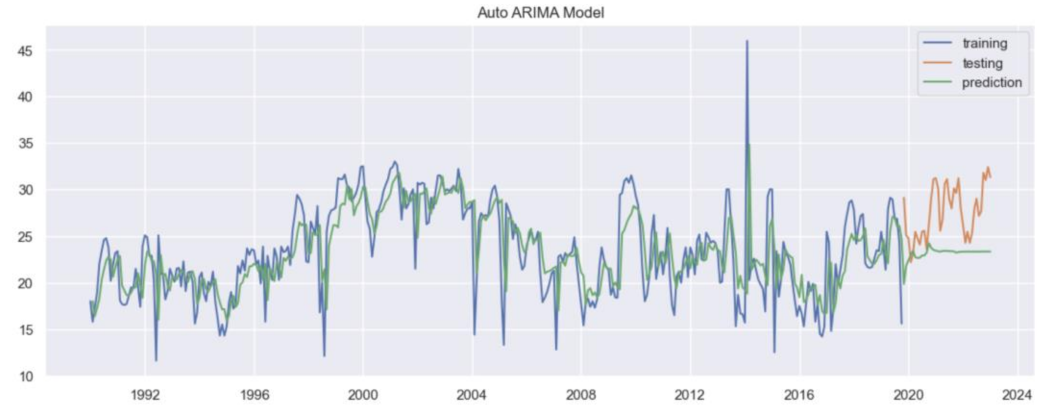
Electricity Consumption: Tail off at ACF Plot, which indicates an AR model. From PACF plot, cut off happens at lag 1. This illustrates an AR(1) model

Modelling (Gas Consumption)

Auto ARIMA Model

Model: SARIMAX(1, 1, 1)x(1, 0, [], 12)

AIC	1922.691
BIC	1938.202




```
Model Mean Absolute Percentage Error on training data is 11.19%
Model Mean Absolute Percentage Error on testing data is 15.39%
```

AR(2) Model

Model: `AutoReg(2)`

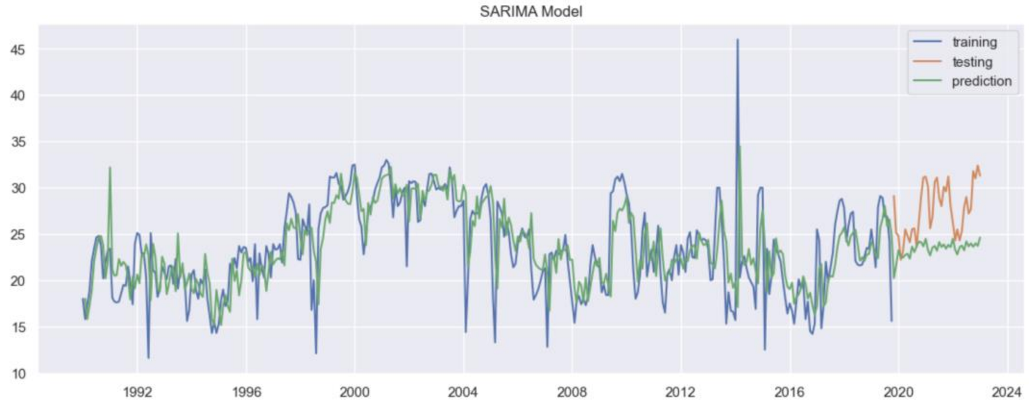
AIC: 1929.124
BIC: 1944.624


```
Model Mean Absolute Percentage Error on training data is 11.33%
Model Mean Absolute Percentage Error on testing data is 16.15%
```

SARIMA Model

Model: SARIMAX(2, 1, 1)x(1, 1, 1, 12)

AIC	1908.079
BIC	1931.141




```
Model Mean Absolute Percentage Error on training data is 11.86%
Model Mean Absolute Percentage Error on testing data is 14.45%
```

First, I fitted an Auto ARIMA model to find the most ideal parameters as a starting point.

Secondly, I implemented a SARIMA model and an AR model. I attempted to log transform the data but there was no improvement to the MAPE.

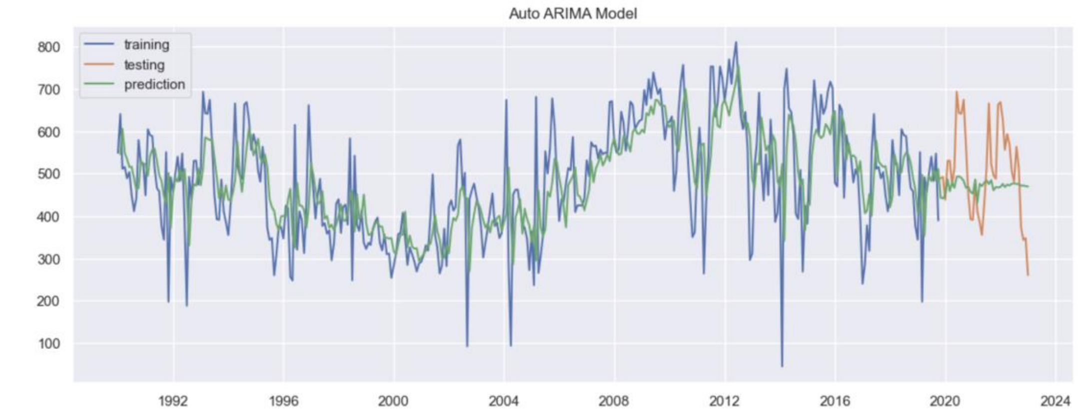
Since our SARIMA model has the lowest AIC and BIC as well as the lowest MAPE, we will further improve our model by finding the best order and seasonal order for it.

Modelling (Water Consumption)

Auto ARIMA Model

Model: SARIMAX(1, 1, 2)x(2, 0, [], 12)

AIC	4325.259
BIC	4348.525




```
Model Mean Absolute Percentage Error on training data is 21.53%
Model Mean Absolute Percentage Error on testing data is 17.50%
```

AR(2) Model

Model: `AutoReg(2)`

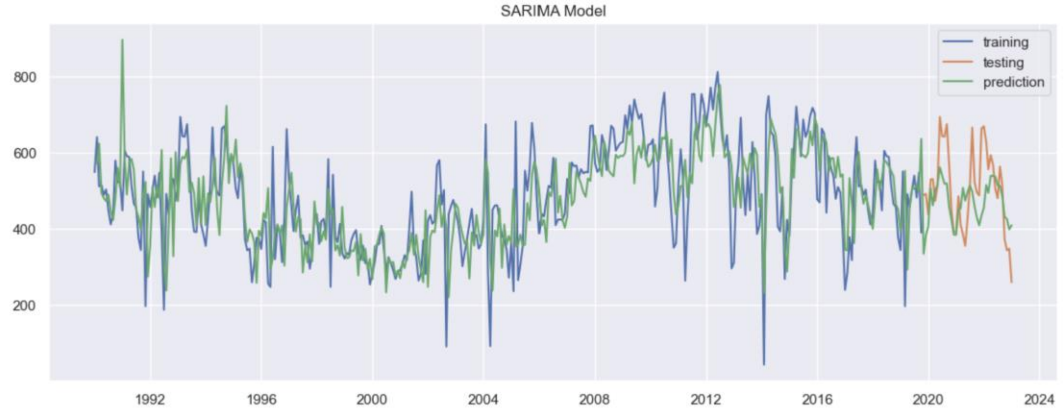
AIC: 4329.892
BIC: 4345.392


```
Model Mean Absolute Percentage Error on training data is 22.05%
Model Mean Absolute Percentage Error on testing data is 18.32%
```

SARIMA Model

Model: SARIMAX(1, 1, 2)x(2, 1, [1], 12)

AIC	4176.362
BIC	4207.111




```
Model Mean Absolute Percentage Error on training data is 19.53%
Model Mean Absolute Percentage Error on testing data is 15.52%
```

First, I fitted an Auto ARIMA model to find the most ideal parameters as a starting point.

Secondly, I implemented a SARIMA model with exogenous variables (gas) and an AR model. Exogenous variables are external factors that might indirectly affect water's time series.

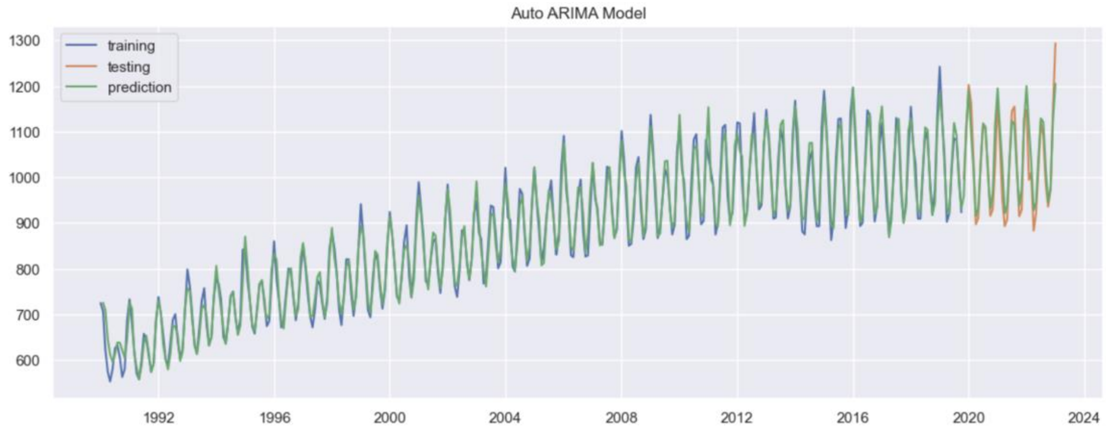
Since our SARIMA model has the lowest AIC and BIC as well as the lowest MAPE, we will further improve our model by finding the best order and seasonal order for it.

Modelling (Electricity Usage)

Auto ARIMA Model

Model: SARIMAX(2, 1, 1)x(1, 0, 1, 12)

AIC	3274.831
BIC	3301.975

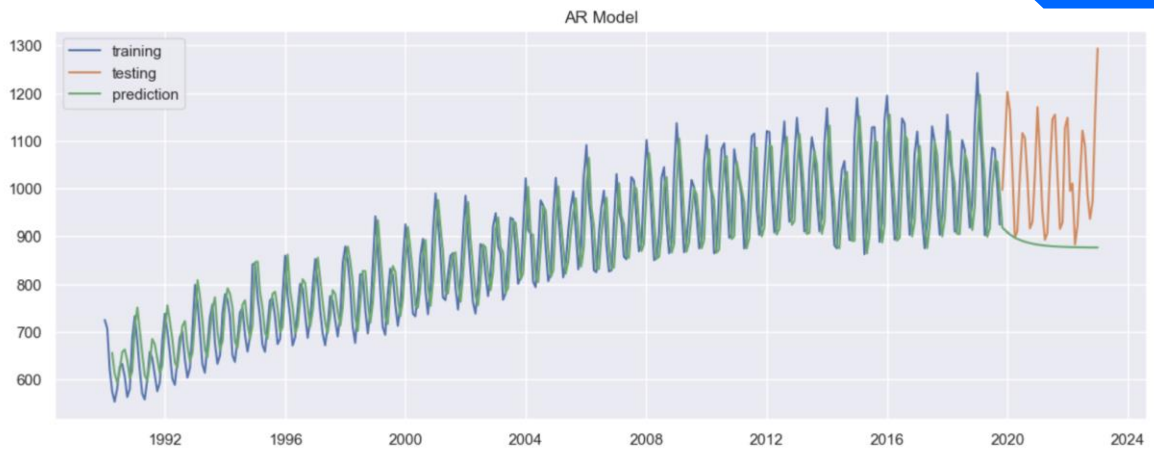



```
Model Mean Absolute Percentage Error on training data is 2.36%
Model Mean Absolute Percentage Error on testing data is 2.95%
```

AR(1) Model

Model: AutoReg(1)

AIC: 4076.064
BIC: 4087.697

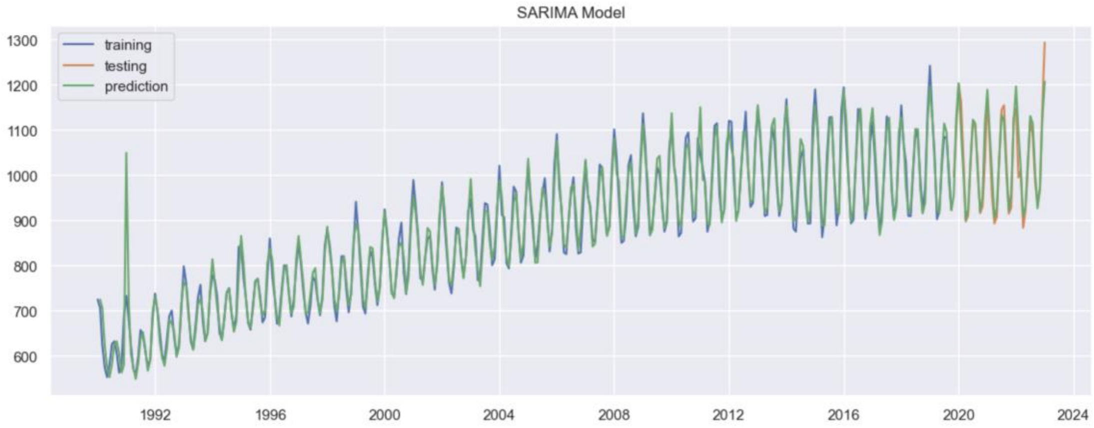



```
Model Mean Absolute Percentage Error on training data is 6.98%
Model Mean Absolute Percentage Error on testing data is 13.56%
```

SARIMA Model

Model: SARIMAX(2, 1, 1)x(2, 1, [1, 2], 12)

AIC	3132.950
BIC	3163.698




```
Model Mean Absolute Percentage Error on training data is 2.16%
Model Mean Absolute Percentage Error on testing data is 2.40%
```

First, I fitted an Auto ARIMA model to find the most ideal parameters as a starting point.

Secondly, I implemented a SARIMA model and an AR model.

Since our SARIMA model has the lowest AIC and BIC as well as the lowest MAPE, we will further improve our model by finding the best order and seasonal order for it.

‘For’ Loop Function

```python
p_values = range(1,4)
d_values = range(1,2)
q_values = range(1,4)
P_values = range(1,4)
D_values = range(1,2)
Q_values = range(1,4)
seasonality = [12]
bestMapeOrder = None
bestAICOrder= None
bestBICOrder = None
bestAIC = 100000000
bestMape = 10000000
bestBIC = 1000000
for p in p_values:
    for d in d_values:
        for q in q_values:
            for P in P_values:
                for D in D_values:
                    for Q in Q_values:
                        for s in seasonality:
                            order=(p,d,q)
                            seasonalOrder=(P,D,Q,s)
                        try:

                            model = SARIMAX(electricityTrain,order=order,seasonal_order=seasonalOrder).fit()
                            predictions = model.forecast(steps=len(electricityTest))
                            mape = mean_absolute_percentage_error(electricityTest, predictions)

                            if(model.aic<bestAIC):
                                bestAIC = model.aic
                                bestAICOrder= f'({order},{seasonalOrder})'
                            if(model.bic<bestBIC):
                                bestBIC = model.bic
                                bestBICOrder= f'({order},{seasonalOrder})'
                            if(mape<bestMape):
                                bestMapeaic = model.aic
                                bestMape = mape
                                bestMapeOrder= f'({order},{seasonalOrder})'


                        except Exception as e:
                            print(f'LU Decomposition Error: {order},{seasonalOrder}')


print(f'Order for Best AIC:{bestAICOrder}')
print(f'Order for Best BIC:{bestBICOrder}')
print(f'Order for Best MAPE:{bestMapeOrder}')
```

I made a for loop to further improve my model and find the best possible order for lowest AIC,BIC and MAPE.

Output:

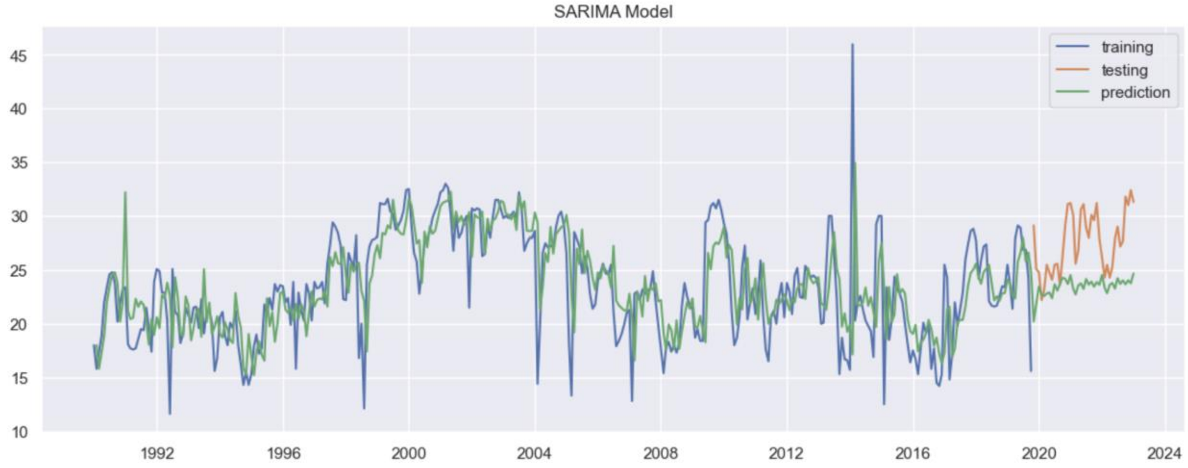
```
Order for Best AIC:((3, 1, 3),(2, 1, 2, 12))
Order for Best BIC:((1, 1, 1),(2, 1, 1, 12))
Order for Best MAPE:((3, 1, 2),(3, 1, 2, 12))
```

Final Model (Improved)

Gas Consumption (SARIMA Improved)

AIC	1906.345
BIC	1925.563

Model: SARIMAX(1, 1, 1)x(1, 1, 1, 12)

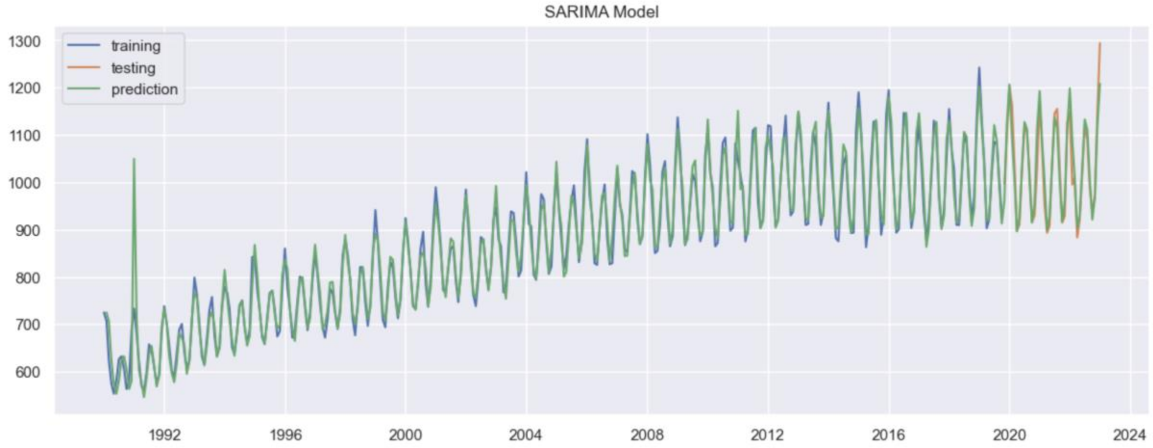



```
Model Mean Absolute Percentage Error on training data is 11.86%
Model Mean Absolute Percentage Error on testing data is 14.25%
```

Electricity Usage (SARIMA Improved)

AIC	3131.233
BIC	3173.512

Model: SARIMAX(3, 1, 3)x(2, 1, [1, 2], 12)

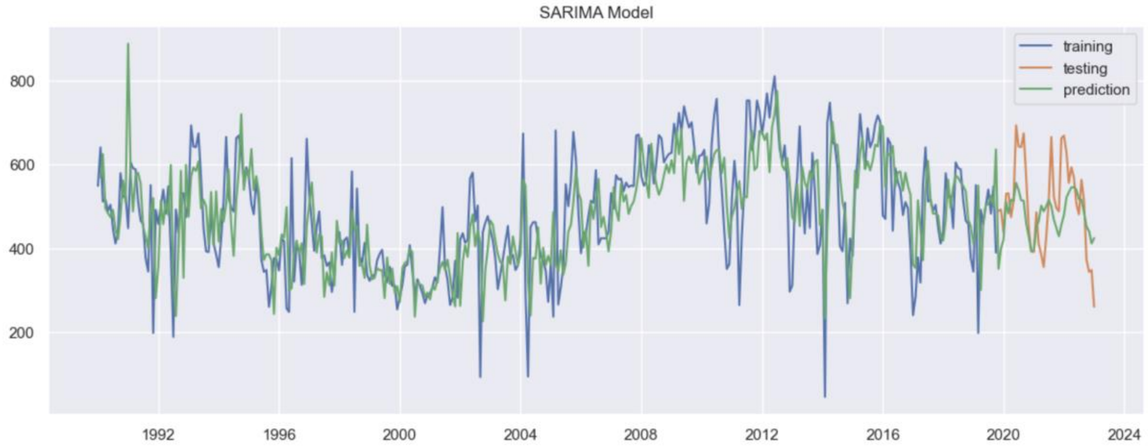



```
Model Mean Absolute Percentage Error on training data is 2.14%
Model Mean Absolute Percentage Error on testing data is 2.31%
```

Water Consumption (SARIMA Improved)

AIC	4183.705
BIC	4225.984

Model: SARIMAX(2, 1, 3)x(2, 1, [1, 2], 12)




```
Model Mean Absolute Percentage Error on training data is 19.63%
Model Mean Absolute Percentage Error on testing data is 15.15%
```

For model improvement, I made a for loop function to find the best order of every SARIMAX Model

1. For Gas Consumption, the AIC dropped from 1908 to 1906 and BIC dropped from 1931 to 1925. The MAPE on test data decreased by 0.20% from 14.45% to 14.25%.
2. For Water Consumption, although the AIC increased from 4176 to 4183 and BIC increased from 4207 to 4225. The MAPE on test data decreased by 0.25% from 15.52% to 15.15%. Upon weighing the trade-off between model complexity and accuracy, I will prioritise a lower MAPE as it reflects a more accurate prediction to the real world.
3. For Electricity Usage, the AIC decreased from 3132 to 3131 and BIC increased from 3163 to 3173. The MAPE decreased by 0.09% from 2.40% to 2.31%

Conclusion

Final Model Chosen:

Gas Consumption: SARIMAX(1,1,1)x(1,1,1,12)

```
Model Mean Absolute Percentage Error on training data is 11.86%
Model Mean Absolute Percentage Error on testing data is 14.25%
```

Water Consumption: SARIMAX(2,1,3)x(2,1,2,12)

```
Model Mean Absolute Percentage Error on training data is 19.63%
Model Mean Absolute Percentage Error on testing data is 15.15%
```

Electricity Consumption: SARIMAX(3,1,3)x(2,1,2,12)

```
Model Mean Absolute Percentage Error on training data is 2.14%
Model Mean Absolute Percentage Error on testing data is 2.31%
```

Exponential Smoothing (Used as Baseline)

Gas: Test Data Mean Absolute Percentage Error: 23.13%

Water: Test Data Mean Absolute Percentage Error: 18.47%

Electricity: Test Data Mean Absolute Percentage Error: 3.99%

My electricity consumption forecast is the most accurate with the lowest MAPE score of 2.14% on training data and 2.31% on testing data. Compared to the exponential smoothing that was used as a baseline, my MAPE has decreased significantly which implies that my model is more accurate when forecasting.

All in all, this time series project has allowed me to learn more in-depth about forecasting a time series and how forecasting is important for decision-makers to make the best decision