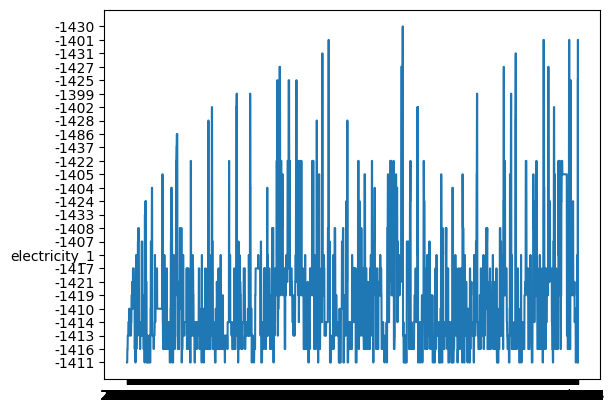
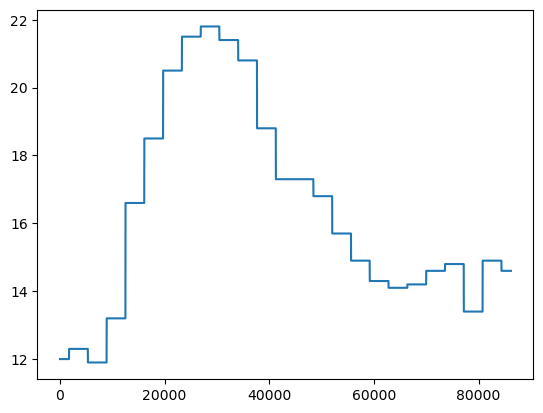
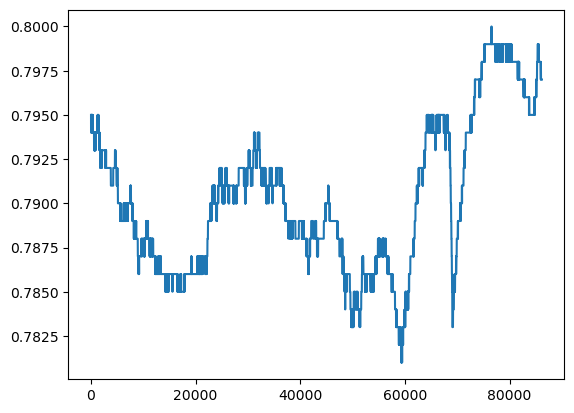
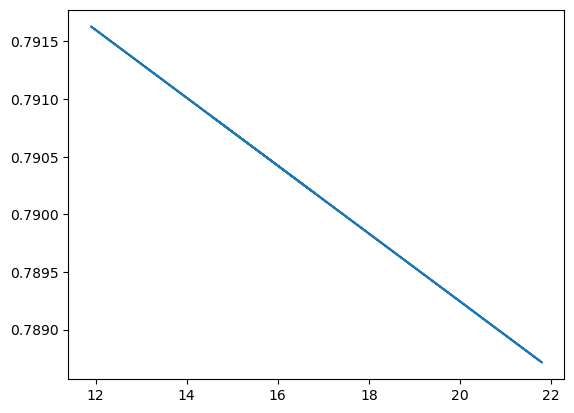
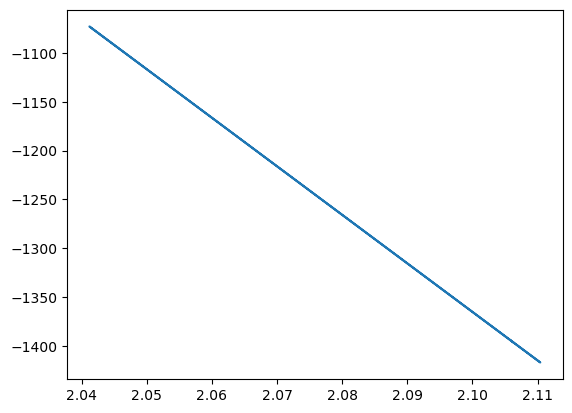
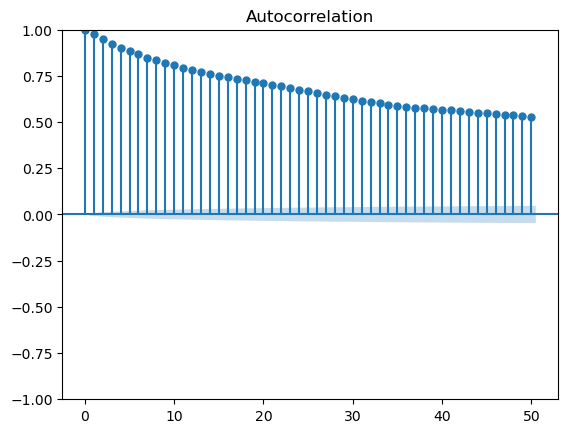
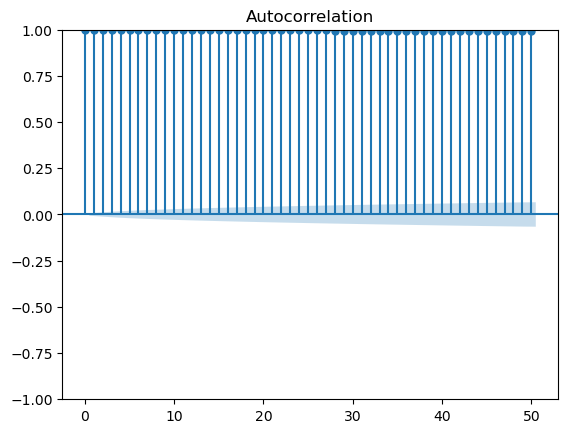
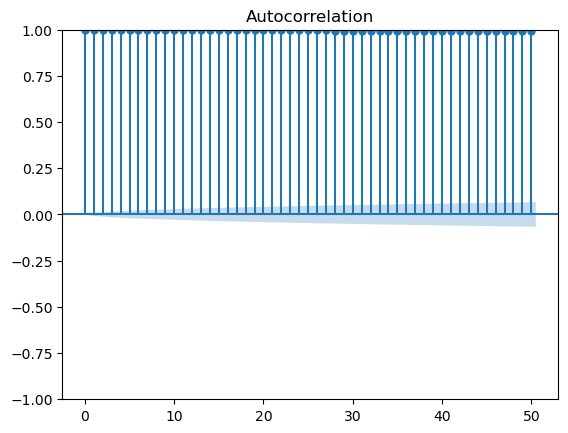
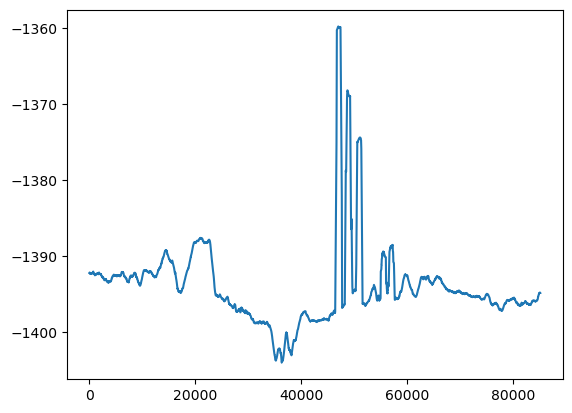
**CNCITY Dataset - Exploratory Data Analysis**

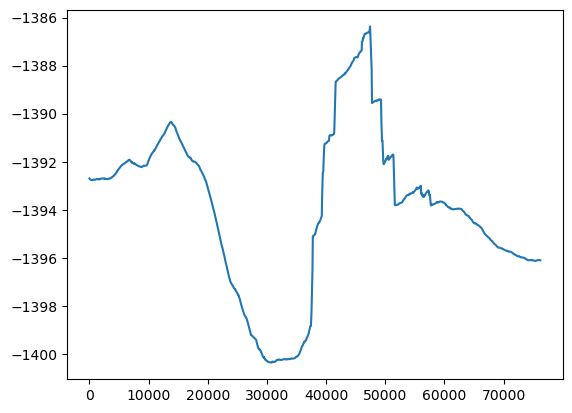
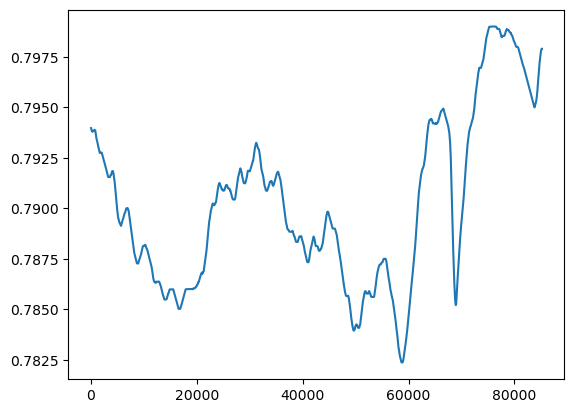
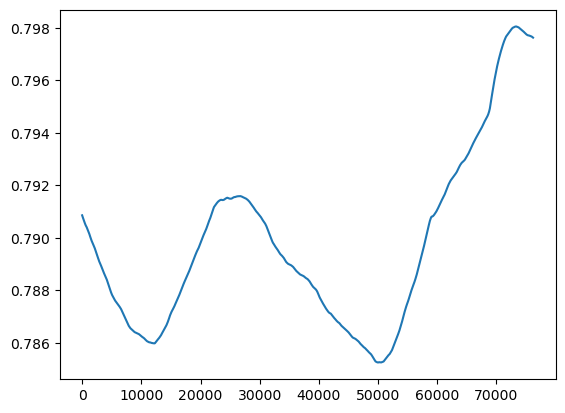
1. **Parsing and tabulating the data**
   1. Done through Cassandra\_query and other python packages
2. **Plotting each variable (stationarity, moving average, seasonality, etc)**
   1. Electricity\_1: seems relatively stationary, does not display seasonality 
   2. Outside\_temperature: ranges from 12 to 22 degrees Celsius
   3. Pressure\_1: 
3. **Linear Regression**
   1. Outside\_temperature and pressure\_1
      1. Split the data frames into train and test sets
      2. Train the model using x\_train and y\_train, then run it on x\_test to evaluate and further improve accuracy
      3. 
      4. Temperature and pressure are negatively correlated
      5. y = -0.0002944x + 0.795132
   2. Electricity\_1 and pressure\_2
      1. 
4. **Autocorrelation (Test for stationarity)**
   1. 
      1. Electricity\_1 - non-stationary
      2. Blue-shaded region: confidence interval
      3. **Suggests that the dataset might be non-stationary**
         1. Slow decay of autocorrelation

* The autocorrelation plot tends to drop quickly in a stationary time series   
  + - 1. Persistent correlations
* Our dataset displays higher autocorrelation values at longer lags
  + 1. Need to make the series stationary before analyzing it (moving average method)
  1. **Pressure\_1 - non-stationary (log transformation did not work)**

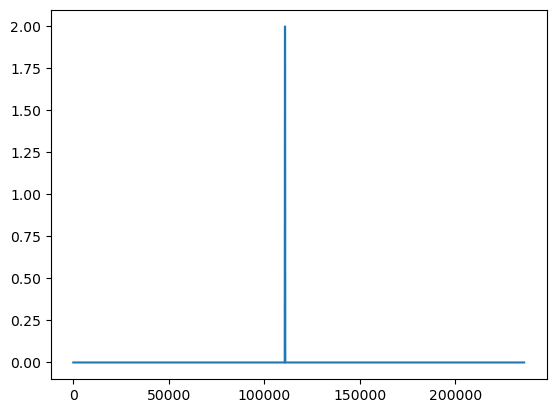
 (after applying log transformation)

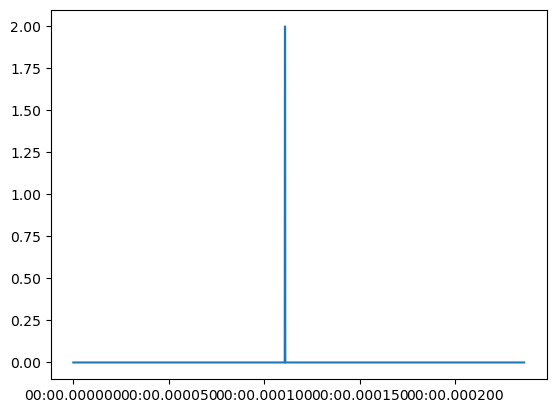
1. **Moving average (Rolling average)**
   1. Electricity\_1 (window = 1,000)



* 1. Window = 10,000  
     
     1. The steep rise and fall between indices 30,000 and 50,000 indicate periods of higher volatility
  2. Pressure\_1 (rolling average)
     1. Window = 1,000  
        
     2. Window = 10,000  
          
        

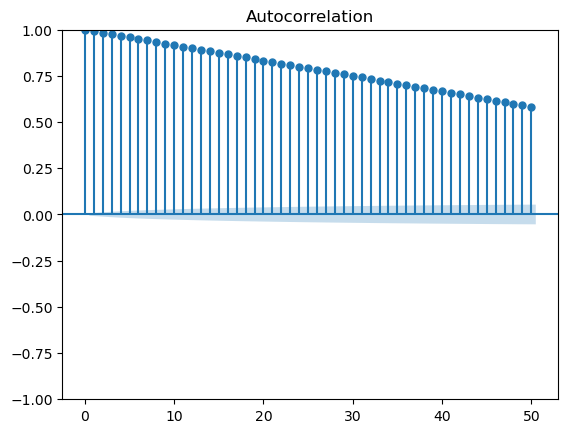
Plotted df[‘is\_maintenance’] and df[‘time’]



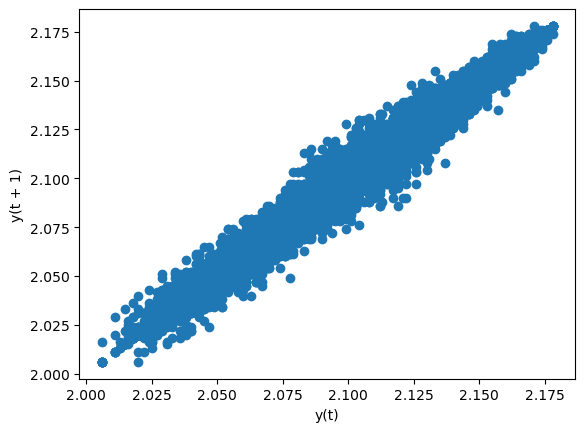


(maintenance\_df → the indices are different)

Autocorrelation plot (is\_maintenance)



Pressure\_2: (lag plot)



**6/13/2024:**

df['pressure\_1'].corr(df['pressure\_1'].shift(1000)): autoregression, lag = 1,000

Output: 0.9184823349674819

df['pressure\_1'].corr(df['pressure\_1'].shift(1000)): autoregression, lag = 1,000

Devised a different approach to preprocessing the data:

* Wrote a function called ‘**process\_dataframe**’ that takes a dataframe and returns a new one.
* Our dataset has a column called ‘Time’ that starts at 0 and ends at 235959 (midnight)
* ‘Process\_dataframe’ takes the dataset and groups the data by hour based on the ‘time’ column
* Then it computes the mean, median, mode, or max of the other columns for each hour group
* After that it creates a new dataframe with all these values and returns the new dataframe

Result\_df: our new dataset (consolidated), has a new column called ‘Hour’

0 0

1 1

2 2

3 3

4 4

5 5

6 6

7 7

8 8

9 9

10 10

11 11

12 12

13 13

14 14

15 15

16 16

17 17

18 18

19 19

20 20

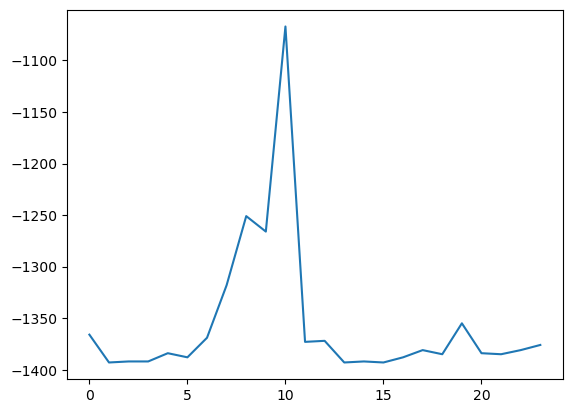
21 21

22 22

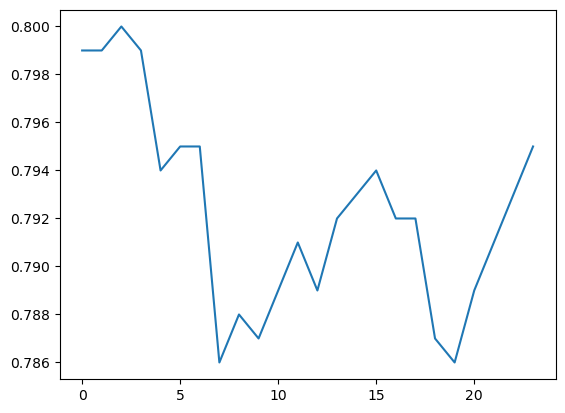
23 23

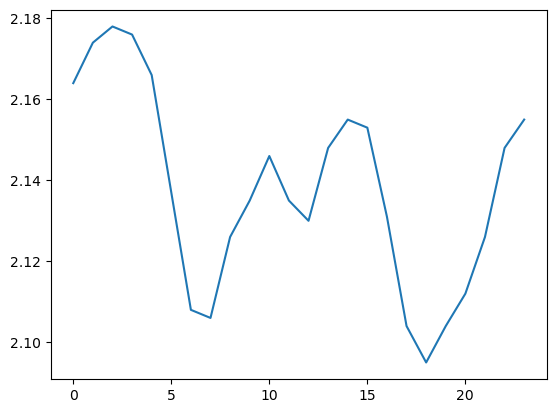
**Visualization: (graphs)**

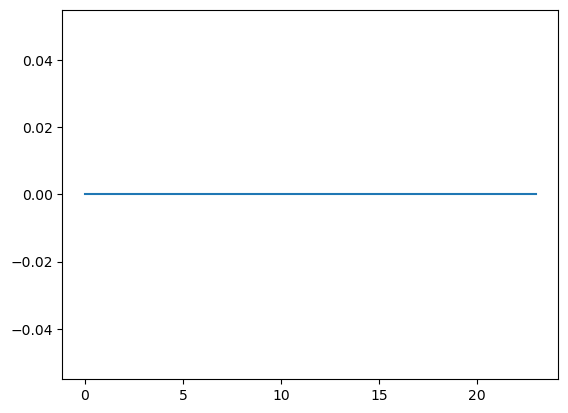
1. **Electricity\_1**

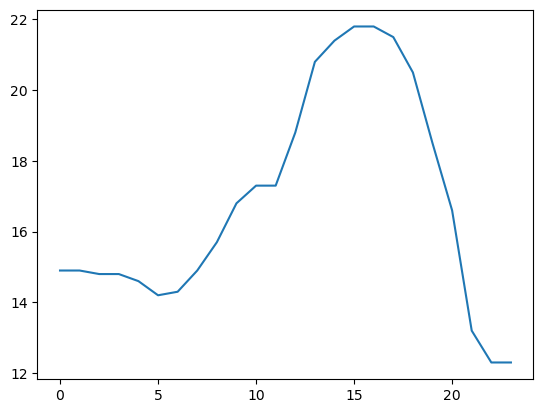
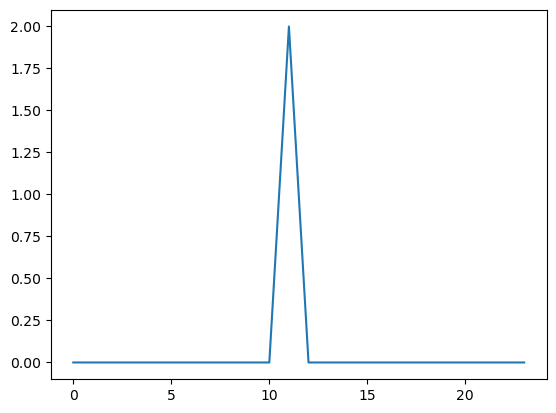
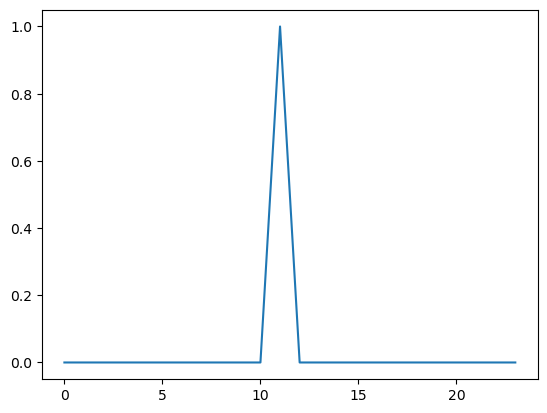


1. **Pressure\_1**

****

1. **Pressure\_2  
   **
2. **Electricity\_2**

****

1. **Outside\_temperature  
   **
2. **Is\_maintenance  
   **
3. **Door\_open  
   **