Week8_SelectBestHouse

November 17, 2020

1 Kijk welk huis de beste data heeft

Er worden op meerdere dingen gechecked in dit programma: 1. Data freeze 2. Data jump 3. Data gap 4. Data reset

En de sheets waarnaar wordt gekeken zijn: - solar - energyHeatPump - energyImmersion - energyWtWreg - smartmeter

modules: Importeer de modules:

[1]:
```
import numpy as np
import pandas as pd
from tqdm import tqdm
import matplotlib.pyplot as plt
from statsmodels.graphics import tsaplots
import statsmodels.api as sm

#zelfgemaakte functies:
import sys
sys.path.insert(0,'/home/18005152/notebooks/zero/imports/')
import Load_data
```

variabelen: Zet alle variabelen op de juiste waardes:

[2]:
```
std_nmr = 18005152
req_sheets = {'solar': [3]}
```

1.1 Functions

Hier worden alle functies gemaakt.

[ ]:
```

```

1.2 Hoofd loop:

[24]:
```
houses[37].rolling(window=24).mean().plot()
houses[37].diff().resample("H").sum().plot()
houses[37].plot()
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd6a2ec01d0>
```

[5]:
```
houses = Load_data.load(37,38,req_sheets,std_nmr)
house = houses[37]['solar_3']#.diff().resample("H").sum()

%matplotlib notebook
for i in range(1,13):
    fig = tsaplots.
 ↪plot_acf(house["2019-"+str(i)],lags=7500,use_vlines=False,title="Autocorrelation␣
 ↪plot SolarEnergy"+str(i))
    plt.xlabel("N-keer 5min timesteps back")
    plt.ylabel("correlation")
    plt.grid()
plt.show()
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
```

[6]:
```
for i in tqdm(range(1,119)):
    for j in req_sheets.keys():
        try:
            Load_data.load(i,i+1,{j:req_sheets[j]},std_nmr)
        except KeyError:
            print(str(i)+":"+j)
```

```
  8%|          | 9/118 [01:05<13:02,  7.18s/it]
10:energyImmersion
  8%|          | 10/118 [01:12<12:44,  7.08s/it]
11:energyImmersion
 14%|         | 17/118 [02:00<11:42,  6.95s/it]
17:smartMeter
 15%|         | 18/118 [02:06<10:47,  6.47s/it]
18:smartMeter
 44%|        | 52/118 [06:02<07:47,  7.09s/it]
52:smartMeter
 45%|        | 53/118 [06:07<07:09,  6.61s/it]
53:smartMeter
 55%|        | 65/118 [07:33<06:08,  6.95s/it]
66:energyWtwReg
```

```
 56%|       | 66/118 [07:39<05:48,  6.70s/it]
67:energyWtwReg
 58%|       | 69/118 [07:59<05:29,  6.72s/it]
70:energyWtwReg
 59%|       | 70/118 [08:05<05:18,  6.63s/it]
71:energyWtwReg
 73%|      | 86/118 [09:56<03:31,  6.61s/it]
87:energyImmersion
 74%|      | 87/118 [10:02<03:22,  6.52s/it]
88:energyImmersion
 85%|      | 100/118 [11:35<02:04,  6.90s/it]
100:smartMeter
 86%|      | 101/118 [11:41<01:48,  6.41s/it]
101:smartMeter
100%|     | 118/118 [13:42<00:00,  6.97s/it]
```