W8_PearsonCorrMatrix

January 12, 2021

[1]:
```
!chown 16095065:zero W8_PearsonCorrMatrix.ipynb
!chmod 660 'W8_PearsonCorrMatrix.ipynb'
!chown 16095065:zero .ipynb_checkpoints/W8_PearsonCorrMatrix-checkpoint.ipynb
```

Niels van Drunen 18062814
Jefry el Bhwash 16095065

1 Pearson Correlation Matrix

Parameter to predict: - Production of tomorrow

[2]:
```
#modules
import numpy as np
import pandas as pd
from tqdm import tqdm
import matplotlib.pyplot as plt
import glob
import seaborn as sns

import sklearn
#from sklearn.model_selection import train_test_split

# import sys
# sys.path.insert(0, '/home/16095065/notebooks/tests/scripts/')
# import jFunc as jF
```

[3]:
```
#functions
def housesListDeltaEnergy(dictio, start, end):
    '''Creates a list of the specified range of houses with the specified sheets
        This function:
        - Resamples
        - Calculates the difference of the cumulative energy
        - Fills NaN with 0
        - Dict format: {strSheet: [intCol, strColSuffix]}'''
    houses = list()
    for i in range(start,end):
```

```
        df_delta = pd.DataFrame()
        df_norma = pd.DataFrame()
        for sheetname, column in dictio.items():
            df = pd.DataFrame(np.load(loadpath + sheetname + '_' + f'{i:03}' +␣
↪'.npy'))
            df = df.set_index(pd.DatetimeIndex(pd.to_datetime(df[0],unit='s').
↪values))
            df = df.resample('5min').sum()
            if sheetname == 'smartMeter':
                col = df[6].shift(-1) - df[6]
                col = col.shift(1)
                df_delta[str(sheetname)+'In'+'_delta'] = col

            col = df[column[0]].shift(-1) - df[column[0]]
            col = col.shift(1)
            df_delta[str(sheetname)+column[1]+'_delta'] = col
            df_delta=df_delta.fillna(0)
        houses.append(df_delta)
    return houses
```

1.0.1 Imports of files

[4]:
```
#KNMI
knmi = pd.read_pickle('KNMI_DF_W8')

#Houses with solar data
loadpath = '/home/16095065/notebooks/zero/DATA/'
house_dict = {'solar': [3, 'Out']}
houses = list()
houses = housesListDeltaEnergy(house_dict, 1,3)
solar1 = houses[0].resample('60min').sum().shift(1)
```

1.0.2 Compile dataframe and plot

[5]:
```
df = pd.DataFrame()
df['Solar'] = solar1['solarOut_delta']
df['SolarGist'] = df['Solar'].shift(24)
df['Temp'] = knmi['temperature']
df['Irr'] = knmi['straling']
df.head(50)
print(df.corr(method="spearman"))
# df.plot()
# plt.ylim(0,350)
```

```
               Solar  SolarGist      Temp       Irr
Solar      1.000000   0.877819  0.477427  0.888041
SolarGist  0.877819   1.000000  0.475669  0.861993
Temp       0.477427   0.475669  1.000000  0.488540
Irr        0.888041   0.861993  0.488540  1.000000
```