W10_MVLR_DataMean

January 12, 2021

1 MVLR: with a new dataset

Niels van Drunen 18062814

Jefry el Bhwash 16095065

This week 10 version changed the dataset from the sum method to the mean method data
It looks like the model favors the production of 3 days before, which is bad

[2]:
```
#modules
import numpy as np
import pandas as pd
from tqdm import tqdm
import matplotlib.pyplot as plt
import glob
from sklearn.metrics import r2_score
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_validate
import seaborn as sns
```

1.0.1 Import of data

[3]:
```
loadpath = '/home/16095065/notebooks/zero/datasetP_mean/'
greathouses = [37,40,41,42,51,53,54,55,56,57,58,60,70,72,99,100,105,108,114,115]
houses = {}
for h in greathouses:
    houses[h] = pd.read_pickle(loadpath + 'Train_' +str(h)).fillna(0)
```

[4]:
```
houses[37].head()
```

[4]:
```
                     s_delta  solar_T-24  solar_T-48  solar_T-72  \
DateTime
2019-01-01 00:00:00      0.0         0.0         0.0         0.0
2019-01-01 01:00:00      0.0         0.0         0.0         0.0
2019-01-01 02:00:00      0.0         0.0         0.0         0.0
2019-01-01 03:00:00      0.0         0.0         0.0         0.0
```

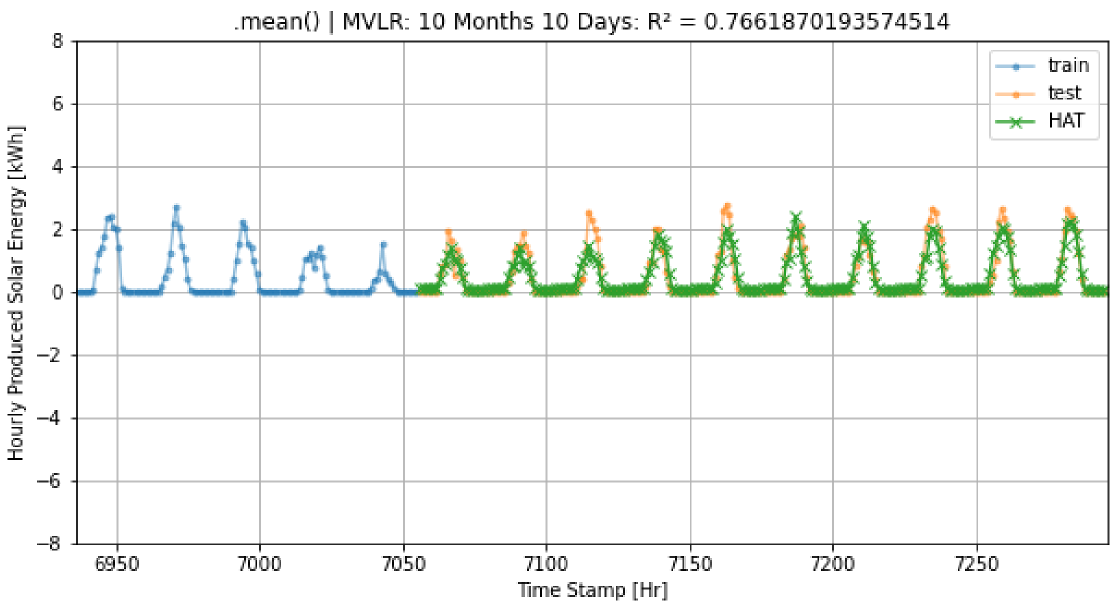
```
2019-01-01 04:00:00      0.0         0.0         0.0         0.0

                     straling_T-24  straling_T-48  straling_T-72  \
DateTime
2019-01-01 00:00:00            0.0            0.0            0.0
2019-01-01 01:00:00            0.0            0.0            0.0
2019-01-01 02:00:00            0.0            0.0            0.0
2019-01-01 03:00:00            0.0            0.0            0.0
2019-01-01 04:00:00            0.0            0.0            0.0

                     temperature_T-24  temperature_T-48  temperature_T-72
DateTime
2019-01-01 00:00:00              93.0               0.0               0.0
2019-01-01 01:00:00              95.0               0.0               0.0
2019-01-01 02:00:00              92.0               0.0               0.0
2019-01-01 03:00:00              90.0               0.0               0.0
2019-01-01 04:00:00              90.0               0.0               0.0
```

2 Model

[8]:
```
%matplotlib inline
from sklearn import linear_model
days = 10

df = houses[37]
df = df['2019-01-01':'2019-10-31']
df['hour'] = df.index.hour

features = ['solar_T-24','solar_T-48', 'solar_T-72', 'straling_T-24',␣
 ↪'straling_T-48', 'straling_T-72', 'hour'] #, 'temperature_T-24',␣
 ↪'temperature_T-48', 'temperature_T-72'
target = 's_delta'

X = df[features].values.reshape(-1, len(features))
y = df[target].values
y = y.reshape(y.shape[0], 1)
print(X.shape)
print(y.shape)
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=(1/len(df.
 ↪index)*24)*days, random_state=0,shuffle=False)

regr = linear_model.LinearRegression()
regr.fit(X_train,y_train)

print('Intercept: \n', regr.intercept_)
print('Coefficients: \n', regr.coef_)
y_hat =  regr.predict(X_test)

plt.figure(figsize=(10,5))
plt.plot(np.arange(X_train.shape[0]), y_train, ".-", label='train', alpha=0.5)
plt.plot(np.arange(X_train.shape[0], X_train.shape[0]+X_test.shape[0]), y_test,␣
 ↪ ".-", label='test', alpha=0.5)
plt.plot(np.arange(X_train.shape[0], X_train.shape[0]+X_test.shape[0]), y_hat, ␣
 ↪ "x-", label='HAT')
plt.xlabel('Time Stamp [Hr]')
plt.ylabel('Hourly Produced Solar Energy [kWh]')
plt.title('.mean() | MVLR: 10 Months 10 Days: R\u00b2 = ' + str(r2_score(y_hat,␣
 ↪y_test)))

plt.ylim([-8,8])
plt.xlim([X_train.shape[0]-(24*5),X_train.shape[0]+X_test.shape[0]]) #lastpart
#plt.xlim([0,X_train.shape[0]+X_test.shape[0]]) #year
plt.grid()
plt.legend()
## R^2 functie toepassen op yhat vs y
print('R\u00b2 score: ', r2_score(y_hat, y_test))
#plt.savefig('W9_MVLR_month.png', dpi=600)
```

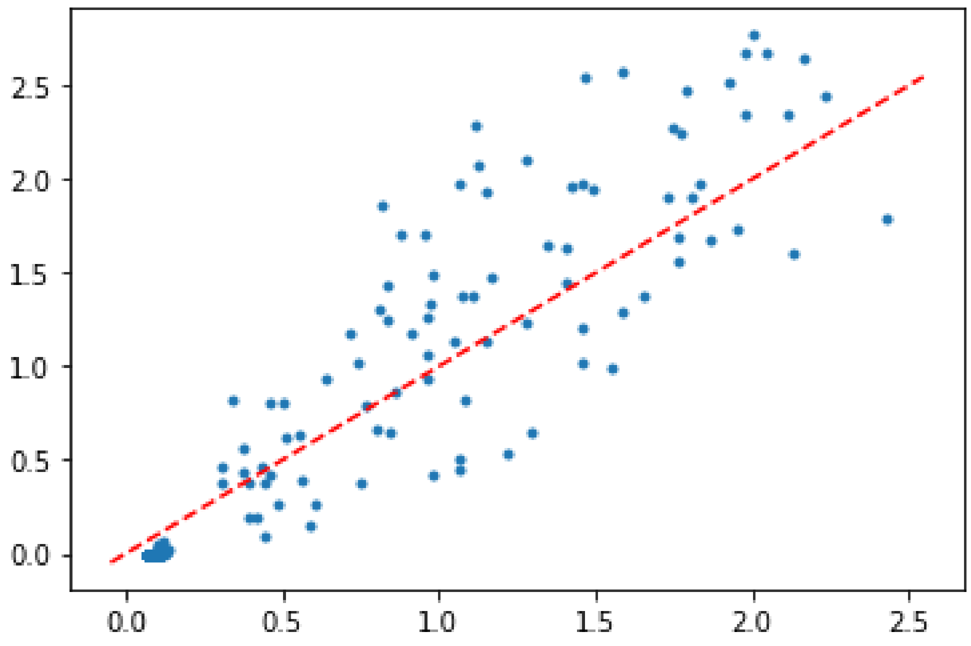
```
(7296, 7)
(7296, 1)
Intercept:
 [0.10683605]
Coefficients:
 [[ 0.26781293  0.10688183  0.22408689  0.00196572  0.0040875   0.00029817
  -0.00159458]]
R² score:  0.7661870193574514
```

```
/opt/jupyterhub/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
```



```
[6]: plt.plot(y_hat, y_test, ".")
     plt.plot(plt.xlim(), plt.xlim(), ls="--", c='r', label="$y$=$\hat{y}$")
     #plt.savefig('W9_MVLR_month_y_yhat.png', dpi=600)
```

```
[6]: [<matplotlib.lines.Line2D at 0x7f47d6f70cc0>]
```



[7]:
```
cv_results = cross_validate(regr, X, y, cv=2)
print(cv_results)
```

```
{'fit_time': array([0.00240397, 0.00142002]), 'score_time': array([0.00066638,
0.00060248]), 'test_score': array([0.76298522, 0.75078212])}
```

[ ]:
```

```