W9_SVR

January 12, 2021

1 SVR model

1.0.1 importeer modules

[1]:
```
#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
from sklearn import linear_model
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
```

1.1 Data klaarmaken

[4]:
```
loadpath = '/home/16095065/notebooks/zero/datasetP/'
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)
```

[5]:
```
df
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-00cf07b74dcd> in <module>
----> 1 df

NameError: name 'df' is not defined
```

[6]:
```
%matplotlib notebook
df = houses[37]
df = df['2019-10-01':'2019-10-31']
#df = df['2019-06-01':'2019-06-30']
#df = df['2019']

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

#scale the data


#Splits de data in train test split:
X = df[features].values.reshape(-1, len(features))
y = df[target].values
y = y.reshape(y.shape[0], 1)
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=(1/len(df.
 ↪index)*24)*10, random_state=False,shuffle=False)
```

```
/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
```

1.2 SVR Trainen

[7]:
```
#train the model:
svr = SVR()
svr.fit(X_train,y_train)

#make data plot ready:
test = np.arange(X_train.shape[0], X_train.shape[0]+X_test.shape[0])
train = np.arange(X_train.shape[0])
```

```
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
```

```
(n_samples, ), for example using ravel().
  return f(**kwargs)
```

1.2.1 Plotje

[8]:
```
%matplotlib notebook
plt.subplots(figsize=(10,5))
plt.plot(train,svr.predict(X_train),color='black',label="SVR fit")
plt.plot(test,svr.predict(X_test),color='g',label="SVR prediction")
plt.scatter(train,y_train,color="r",label="Training Data")
plt.scatter(test,y_test,color="b",label="Testing Data")

#Nice layout:
plt.xlabel("Datapoint [hr]")
plt.ylabel("Solar production hourly [kWh]")
plt.legend(loc="upper right")
plt.title("R^2 = "+str(svr.score(X_test,y_test)))
plt.ylim([0,4])
#plt.xlim([X_train.shape[0]-(24*5),X_train.shape[0]+X_test.shape[0]])
plt.xlim([0,X_train.shape[0]+X_test.shape[0]])
plt.grid()
plt.show()

#print de score:
print("R2-score test data:")
print(svr.score(X_test,y_test))
```

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

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

```
R2-score test data:
0.7233646543700721
```

[6]:
```
print(df['2019-10-01':'2019-10-31']['s_delta'].sum())
print(svr.predict(X_test).sum())
print(df['2019-10-01':'2019-10-31']['s_delta'].sum()/svr.predict(X_test).sum())
print()
print(df['2019-01-01':'2019-10-01']['s_delta'].sum())
print(svr.predict(X_train).sum())
print(df['2019-01-01':'2019-10-01']['s_delta'].sum()/svr.predict(X_train).sum())
```

```
338.0
91.13248355533071
3.7088860833556097

10.2400000000016
```

```
198.90822536231903
0.0514810284056833
```

1.3 Evaluatie

[12]:
```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
y = y_test
yhat = svr.predict(X_test)

r2 = r2_score(yhat, y)
rmse = np.sqrt(mean_squared_error(yhat, y))
actual, pred = np.array(y), np.array(yhat)
mape = np.mean(np.abs((actual - pred) / actual)) * 100
mae = mean_absolute_error(yhat, y)
print('R\u00b2: \t%.2f \nRMSE: \t%.2f \nMAPE: \t%.2f \nMAE: \t%.2f' %␣
 ↪(r2,rmse,mape,mae))
```

```
R²:     0.42
RMSE:   0.41
MAPE:   inf
MAE:    0.23
```

```
/opt/jupyterhub/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:9:
RuntimeWarning: divide by zero encountered in true_divide
  if __name__ == '__main__':
```

[9]:
```
%matplotlib notebook
plt.scatter(y_test,svr.predict(X_test))
plt.xlabel("y [kWh]")
plt.ylabel("y_hat [kWh]")
#plt.legend(loc="upper right")
plt.title("R^2= "+str(r2_score(y_test,svr.predict(X_test))))
plt.plot(plt.xlim(), plt.xlim(), ls="--", c='r', label="$y$=$\hat{y}$")
plt.grid()
plt.show()
```

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

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

[ ]:
```
```