In [4]:
```
import pandas as pd
import numpy as np
import pylab as py
import seaborn as sns
```

In [5]:
```
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [3]:
```
data=pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/00374/energydata_complete.csv')
```

In [4]:
```
data.head()
```

Out[4]:

	date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	...	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpoint	rv1	rv2
0	2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.790000	19.79	44.730000	19.000000	...	17.033333	45.53	6.600000	733.5	92.0	7.000000	63.000000	5.3	13.275433	13.275433
1	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	...	17.066667	45.56	6.483333	733.6	92.0	6.666667	59.166667	5.2	18.606195	18.606195
2	2016-01-11 17:20:00	50	30	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	...	17.000000	45.50	6.366667	733.7	92.0	6.333333	55.333333	5.1	28.642668	28.642668
3	2016-01-11 17:30:00	50	40	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	...	17.000000	45.40	6.250000	733.8	92.0	6.000000	51.500000	5.0	45.410389	45.410389
4	2016-01-11 17:40:00	60	40	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	...	17.000000	45.40	6.133333	733.9	92.0	5.666667	47.666667	4.9	10.084097	10.084097

5 rows × 29 columns

In [5]:
```
data.to_csv('HAMOYE_STAGE_B.csv',index_label=False)
```

In [6]:
```
df=pd.read_csv('HAMOYE_STAGE_B.csv')
```

In [7]:
```
df.head()
```

Out[7]:

	date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	...	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpoint	rv1	rv2
0	2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.790000	19.79	44.730000	19.000000	...	17.033333	45.53	6.600000	733.5	92.0	7.000000	63.000000	5.3	13.275433	13.275433
1	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	...	17.066667	45.56	6.483333	733.6	92.0	6.666667	59.166667	5.2	18.606195	18.606195
2	2016-01-11 17:20:00	50	30	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	...	17.000000	45.50	6.366667	733.7	92.0	6.333333	55.333333	5.1	28.642668	28.642668
3	2016-01-11 17:30:00	50	40	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	...	17.000000	45.40	6.250000	733.8	92.0	6.000000	51.500000	5.0	45.410389	45.410389
4	2016-01-11 17:40:00	60	40	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	...	17.000000	45.40	6.133333	733.9	92.0	5.666667	47.666667	4.9	10.084097	10.084097

5 rows × 29 columns

In [8]:
```
df.isnull().any()
```

Out[8]:
```
date           False
Appliances     False
lights         False
T1             False
RH_1           False
T2             False
RH_2           False
T3             False
RH_3           False
T4             False
RH_4           False
T5             False
RH_5           False
T6             False
RH_6           False
T7             False
RH_7           False
T8             False
RH_8           False
T9             False
RH_9           False
T_out          False
Press_mm_hg    False
RH_out         False
Windspeed      False
Visibility     False
Tdewpoint      False
rv1            False
rv2            False
dtype: bool
```

In [9]:
```
mis=[col for col in df.columns if df[col].isnull().any()]
```

In [10]:
```
mis
```

Out[10]:
```
[]
```

In [11]:
```
type(df.date.loc[1])
```

Out[11]:
```
str
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['date', 'Appliances', 'lights', 'T1', 'RH_1', 'T2', 'RH_2', 'T3',
       'RH_3', 'T4', 'RH_4', 'T5', 'RH_5', 'T6', 'RH_6', 'T7', 'RH_7', 'T8',
       'RH_8', 'T9', 'RH_9', 'T_out', 'Press_mm_hg', 'RH_out', 'Windspeed',
       'Visibility', 'Tdewpoint', 'rv1', 'rv2'],
      dtype='object')
```

In [13]:
```
'lights' in df.columns
```

Out[13]:
```
True
```

In [14]:
```
cdf=df.drop(['date','lights'],axis=1)
```

In [15]:
```
cdf.head()
```

Out[15]:

	Appliances	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	T5	...	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpoint	rv1	rv2
0	60	19.89	47.596667	19.2	44.790000	19.79	44.730000	19.000000	45.566667	17.166667	...	17.033333	45.53	6.600000	733.5	92.0	7.000000	63.000000	5.3	13.275433	13.275433
1	60	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	45.992500	17.166667	...	17.066667	45.56	6.483333	733.6	92.0	6.666667	59.166667	5.2	18.606195	18.606195
2	50	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	45.890000	17.166667	...	17.000000	45.50	6.366667	733.7	92.0	6.333333	55.333333	5.1	28.642668	28.642668
3	50	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	45.723333	17.166667	...	17.000000	45.40	6.250000	733.8	92.0	6.000000	51.500000	5.0	45.410389	45.410389
4	60	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	45.530000	17.200000	...	17.000000	45.40	6.133333	733.9	92.0	5.666667	47.666667	4.9	10.084097	10.084097

5 rows × 27 columns

In [16]:
```
from sklearn.preprocessing import MinMaxScaler
norms=MinMaxScaler()
```

In [17]:
```
trans=pd.DataFrame(norms.fit_transform(cdf),columns=cdf.columns)
```

In [18]:
```
X=trans.drop('Appliances',axis=1)
y=trans.Appliances
```

In [65]:
```
from sklearn.model_selection import train_test_split
```

In [66]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7,test_size=0.3, random_state=42)
```

In [67]:
```
from sklearn.linear_model import LinearRegression
lin=LinearRegression()
```

In [68]:
```
lin.fit(X_train,y_train)
```

Out[68]:
```
LinearRegression()
```

In [69]:
```
from sklearn.linear_model import LinearRegression
lin=LinearRegression()
lin.fit(X_train,y_train)
```

Out[69]:
```
LinearRegression()
```

In [70]:
```
predict=lin.predict(X_test)
```

In [71]:
```
predict
```

Out[71]:
```
array([0.03322207, 0.24411599, 0.03400024, ..., 0.06844707, 0.10032325,
       0.05722198])
```

In [72]:
```
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
```

In [73]:
```
r2_score(X_train.T2,y_train)
```

Out[73]:
```
-2.233222777237864 8
```

In [62]:
```
mean_absolute_error(X_train.T2,y_train)
```

Out[62]:
```
0.24274876594062624
```

In [33]:
```
np.sum(np.square(X_train.T2-y_train))
```

Out[33]:
```
1136.5633215814557
```

In [63]:
```
np.sqrt(mean_squared_error(X_train.T2,y_train))
```

Out[63]:
```
0.28683826868672
```

In [35]:
```
lin.coef_
```

Out[35]:
```
array([-0.00328105,  0.5535466 , -0.23617792, -0.45669795,  0.29062714,
        0.09604827,  0.028981  ,  0.02638578, -0.01565684,  0.01600579,
        0.23642491,  0.03804865,  0.01031878, -0.04461364,  0.10195505,
       -0.15759548, -0.18994077, -0.03980832, -0.32185067,  0.00683933,
       -0.07767065,  0.02918313,  0.01230661,  0.11775773,  0.0007701 ,
        0.0007701 ])
```

In [36]:
```
X.columns
```

Out[36]:
```
Index(['T1', 'RH_1', 'T2', 'RH_2', 'T3', 'RH_3', 'T4', 'RH_4', 'T5', 'RH_5',
       'T6', 'RH_6', 'T7', 'RH_7', 'T8', 'RH_8', 'T9', 'RH_9', 'T_out',
       'Press_mm_hg', 'RH_out', 'Windspeed', 'Visibility', 'Tdewpoint', 'rv1',
       'rv2'],
      dtype='object')
```

In [39]:
```
cd=pd.DataFrame(lin.coef_,X.columns,columns=['coef'])
```

In [42]:
```
cd.sort_values('coef')
```

Out[42]:

	coef
RH_2	-0.456698
T_out	-0.321860
T2	-0.236178
T9	-0.189941
RH_8	-0.157595
RH_out	-0.077671
RH_7	-0.044614
RH_9	-0.039800
T5	-0.015657
T1	-0.003281
rv1	0.000770
rv2	0.000770
Press_mm_hg	0.006839
T7	0.010319
Visibility	0.012307
RH_5	0.016006
RH_4	0.026386
T4	0.028981
Windspeed	0.029183
RH_6	0.038049
RH_3	0.096048
T8	0.101995
Tdewpoint	0.117758
T6	0.236425
T3	0.290627
RH_1	0.553547

In [46]:
```
from sklearn.linear_model import Lasso
las=Lasso(alpha=0.001)
```

In [47]:
```
las.fit(X_train,y_train)
```

Out[47]:
```
Lasso(alpha=0.001)
```

In [52]:
```
la=pd.DataFrame(las.coef_)
```

In [57]:
```
la
```

Out[57]:

	0
0	0.000000
1	0.017880
2	0.000000
3	-0.000000
4	0.000000
5	0.000000
6	-0.000000
7	0.000000
8	-0.000000
9	0.000000
10	0.000000
11	-0.000000
12	-0.000000
13	-0.000000
14	0.000000
15	-0.000110
16	-0.000000
17	-0.000000
18	0.000000
19	-0.000000
20	-0.049557
21	0.002912
22	0.000000
23	0.000000
24	-0.000000
25	-0.000000

In [58]:
```
las_pre=las.predict(X_test)
```

In [64]:
```
np.sqrt(mean_absolute_error(y_test,las_pre))
```

Out[64]:
```
0.23506730912924118
```

In [ ]:
```
```