In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.linear_model import LinearRegression
```

In [2]:

```
#Read the Data

url="https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction"
df= pd.read_csv(url, error_bad_lines=False)
```

In [3]:

```
from sklearn.preprocessing import MinMaxScaler
scaler= MinMaxScaler()
dropped_df=df.drop(columns=["date", "lights"])
normalised_df=pd.DataFrame(scaler.fit_transform(dropped_df),columns=dropped_df.columns)
features_df=normalised_df.drop(columns=["Appliances"])
appliances_df=normalised_df["Appliances"]
```

In [7]:

```
# Question 12
t2=np.array(normalised_df["T2"]).reshape(-1,1)
t6=np.array(normalised_df["T6"]).reshape(-1,1)
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
lmodel=LinearRegression()
x_train, x_test, y_train, y_test = train_test_split(t2, t6, test_size=0.3,
random_state=42)
lmodel.fit(x_train,y_train)
predicted_values=lmodel.predict(x_test)
from sklearn.metrics import r2_score
rsquare= r2_score(y_test,predicted_values)
round(rsquare,3)
```

Out[7]:

```
0.643
```

In [12]:

```
#Question 13
x_train, x_test, y_train, y_test = train_test_split(features_df, appliances_df,
test_size=0.3, random_state=42)
from sklearn.linear_model import LinearRegression
lmodel=LinearRegression()
lmodel.fit(x_train,y_train)
predicted_values=lmodel.predict(x_test)
from sklearn.metrics import mean_absolute_error
mae=mean_absolute_error(y_test,predicted_values)
print(round(mae,3))
```

```
0.05
```

In [16]:

```
#Question 14
rss = np.sum(np.square(y_test-predicted_values))
rss
```

Out[16]:

```
45.34762967266377
```

In [14]:

```
#Question 15
from sklearn.metrics import mean_squared_error
rmse=np.sqrt(mean_squared_error(y_test,predicted_values))
round(rmse,3)
```

Out[14]:

```
0.088
```

In [17]:

```
#Question 16
rsquare= r2_score(y_test,predicted_values)
round(rsquare,3)
```

Out[17]:

```
0.149
```

In [20]:

```
ridge_reg=Ridge(alpha=0.4)
ridge_reg=Ridge(alpha=0.4)
dr= ridge_reg.fit(x_train,y_train).coef_
print(ridge_reg.fit(x_train,y_train))
print(dr)

predicted_values_ridge=ridge_reg.predict(x_test)
mae_ridge=mean_absolute_error(y_test,predicted_values_ridge)
rmse_ridge=np.sqrt(mean_squared_error(y_test,predicted_values_ridge))
print("the ridge rmse is %.3f"%round(rmse_ridge,3))
#The Rmse is the same 0.088 , question 18
```

```
Ridge(alpha=0.4, copy_X=True, fit_intercept=True, max_iter=None,
      normalize=False, random_state=None, solver='auto', tol=0.001)
[-0.01840621  0.5195253  -0.20139673 -0.41107123  0.28808681  0.0951346
  0.02738389  0.02457853 -0.01985322  0.01615237  0.21729178  0.03551862
  0.01009781 -0.04597696  0.10102815 -0.15683005 -0.1889163  -0.04136654
 -0.26217227  0.00658387 -0.05472365  0.03026762  0.01207649  0.08312757
  0.00074817  0.00074817]
the ridge rmse is 0.088
```

In [22]:

```
#question 20
lasso_reg=Lasso(alpha=0.001)
lasso_reg.fit(x_train,y_train)
predicted_values_lasso=lasso_reg.predict(x_test)

rmse_lasso=np.sqrt(mean_squared_error(y_test,predicted_values_lasso))
print("the lasso rmse is %.3f"%round(rmse_lasso,3))
```

```
the lasso rmse is 0.094
```

In [25]:

```
#Question 19
def get_weights_df(model, feat, col_name):
#this function returns the weight of every feature
    weights = pd.Series(model.coef_, feat.columns).sort_values()
    weights_df = pd.DataFrame(weights).reset_index()
    weights_df.columns = ['Features', col_name]
    weights_df[col_name].round(3)
    return weights_df
linear_model_weights = get_weights_df(lmodel, x_train, 'Linear_Model_Weight')
ridge_weights_df = get_weights_df(ridge_reg, x_train, 'Ridge_Weight')
lasso_weights_df = get_weights_df(lasso_reg, x_train, 'Lasso_weight')
final_weights = pd.merge(linear_model_weights, ridge_weights_df, on='Features')
final_weights = pd.merge(final_weights, lasso_weights_df, on='Features')
final_weights
#for the lasso weights, there are 4 non-zero feature weights
```

Out[25]:

| | Features | Linear_Model_Weight | Ridge_Weight | Lasso_weight |
|---|---|---|---|---|
| **0** | RH_2 | -0.456698 | -0.411071 | -0.000000 |
| **1** | T_out | -0.321860 | -0.262172 | 0.000000 |
| **2** | T2 | -0.236178 | -0.201397 | 0.000000 |
| **3** | T9 | -0.189941 | -0.188916 | -0.000000 |
| **4** | RH_8 | -0.157595 | -0.156830 | -0.000110 |
| **5** | RH_out | -0.077671 | -0.054724 | -0.049557 |
| **6** | RH_7 | -0.044614 | -0.045977 | -0.000000 |
| **7** | RH_9 | -0.039800 | -0.041367 | -0.000000 |
| **8** | T5 | -0.015657 | -0.019853 | -0.000000 |
| **9** | T1 | -0.003281 | -0.018406 | 0.000000 |
| **10** | rv1 | 0.000770 | 0.000748 | -0.000000 |
| **11** | rv2 | 0.000770 | 0.000748 | -0.000000 |
| **12** | Press_mm_hg | 0.006839 | 0.006584 | -0.000000 |
| **13** | T7 | 0.010319 | 0.010098 | -0.000000 |
| **14** | Visibility | 0.012307 | 0.012076 | 0.000000 |
| **15** | RH_5 | 0.016006 | 0.016152 | 0.000000 |
| **16** | RH_4 | 0.026386 | 0.024579 | 0.000000 |
| **17** | T4 | 0.028981 | 0.027384 | -0.000000 |
| **18** | Windspeed | 0.029183 | 0.030268 | 0.002912 |
| **19** | RH_6 | 0.038049 | 0.035519 | -0.000000 |
| **20** | RH_3 | 0.096048 | 0.095135 | 0.000000 |
| **21** | T8 | 0.101995 | 0.101028 | 0.000000 |
| **22** | Tdewpoint | 0.117758 | 0.083128 | 0.000000 |
| **23** | T6 | 0.236425 | 0.217292 | 0.000000 |
| **24** | T3 | 0.290627 | 0.288087 | 0.000000 |
| **25** | RH_1 | 0.553547 | 0.519525 | 0.017880 |

In [ ]:

In [ ]: