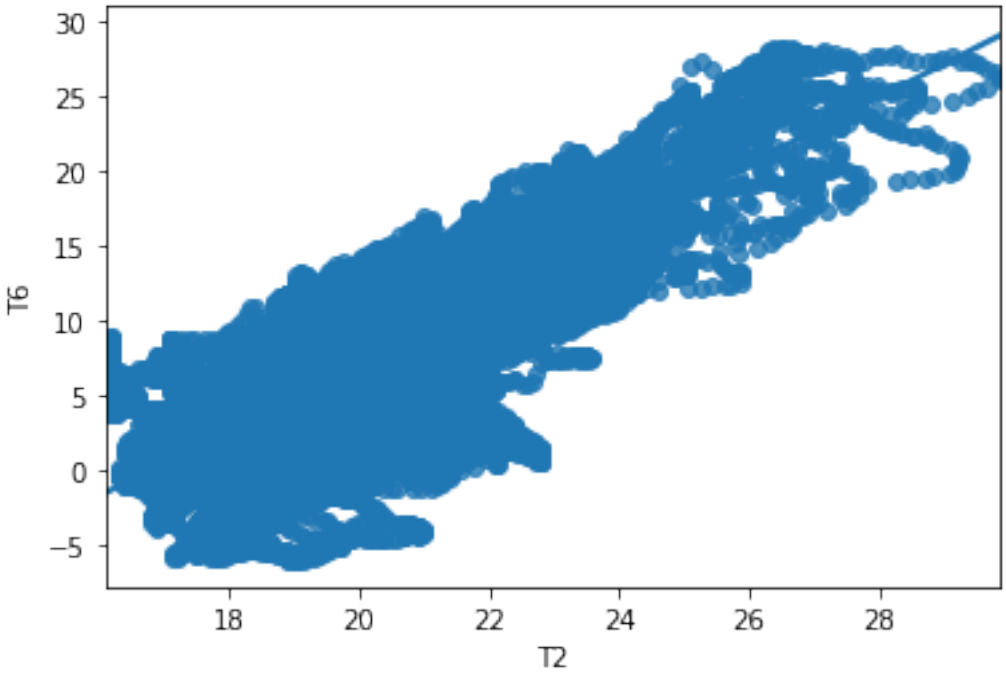
```
#importing the python libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#importing the dataset
dataset = pd.read_csv('energydata_complete.csv')

dataset.shape

(19735, 29)

dataset.head()

                  date  Appliances  lights     T1       RH_1    T2
RH_2  \
0  2016-01-11 17:00:00          60      30  19.89  47.596667  19.2
44.790000
1  2016-01-11 17:10:00          60      30  19.89  46.693333  19.2
44.722500
2  2016-01-11 17:20:00          50      30  19.89  46.300000  19.2
44.626667
3  2016-01-11 17:30:00          50      40  19.89  46.066667  19.2
44.590000
4  2016-01-11 17:40:00          60      40  19.89  46.333333  19.2
44.530000

      T3       RH_3         T4  ...          T9   RH_9     T_out
Press_mm_hg  \
0  19.79  44.730000  19.000000  ...  17.033333  45.53  6.600000
733.5
1  19.79  44.790000  19.000000  ...  17.066667  45.56  6.483333
733.6
2  19.79  44.933333  18.926667  ...  17.000000  45.50  6.366667
733.7
3  19.79  45.000000  18.890000  ...  17.000000  45.40  6.250000
733.8
4  19.79  45.000000  18.890000  ...  17.000000  45.40  6.133333
733.9

   RH_out  Windspeed  Visibility  Tdewpoint        rv1        rv2
0    92.0   7.000000   63.000000        5.3  13.275433  13.275433
1    92.0   6.666667   59.166667        5.2  18.606195  18.606195
2    92.0   6.333333   55.333333        5.1  28.642668  28.642668
3    92.0   6.000000   51.500000        5.0  45.410389  45.410389
4    92.0   5.666667   47.666667        4.9  10.084097  10.084097

[5 rows x 29 columns]
```

```
#Normalising the dataset
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
featured_dataset = dataset.drop(columns = ['date', 'lights'])
normalised_dataset =
pd.DataFrame(scaler.fit_transform(featured_dataset))
heating_target = featured_dataset['Appliances']

#splitting the dataset
from sklearn.model_selection import train_test_split
x_test, x_train, y_test, y_train = train_test_split(featured_dataset,
heating_target, test_size = 0.3, random_state = 42)

#select a sample of data
sns.regplot(x='T2', y='T6', data=dataset)
```

```
<AxesSubplot:xlabel='T2', ylabel='T6'>
```




```
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(x_train, y_train)
```

```
LinearRegression()
```

```
predicted_values = lin_reg.predict(x_test)

# R2 value
from sklearn.metrics import r2_score
r2_score = r2_score(y_test, predicted_values)
r2_score
```

```
1.0
```

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y_test, predicted_values)
mae
```

```
1.1489019281812012e-13
```

```
# residual sum of square
rss = np.sum(np.square(y_test - predicted_values))
rss
```

```
4.925673808599796e-22
```

```
from sklearn.metrics import mean_squared_error
mse = np.sqrt(mean_squared_error(y_test, predicted_values))
mse
```

```
1.8883091532838084e-13
```

```
from sklearn.linear_model import Ridge
ridge_reg = Ridge(alpha =0.4)
ridge_reg.fit(x_train, y_train)
```

```
Ridge(alpha=0.4)
```