```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Importing Data

```
data=pd.read_csv("Steel_industry_data.csv")
data.tail()
```

	date	Usage_kWh	Lagging_Current_Reactive.Power_kVarh	Leading_Current_Reactive_Power_kVarh	CO2(tCO2)	Lagging_Current_Powe
35035	31/12/2018 23:00	3.85	4.86	0.00	0.0	
35036	31/12/2018 23:15	3.74	3.74	0.00	0.0	
35037	31/12/2018 23:30	3.78	3.17	0.07	0.0	
35038	31/12/2018 23:45	3.78	3.06	0.11	0.0	
35039	31/12/2018 00:00	3.67	3.02	0.07	0.0	

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35040 entries, 0 to 35039
Data columns (total 11 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   date                                  35040 non-null  object
 1   Usage_kWh                             35040 non-null  float64
 2   Lagging_Current_Reactive.Power_kVarh  35040 non-null  float64
 3   Leading_Current_Reactive_Power_kVarh  35040 non-null  float64
 4   CO2(tCO2)                             35040 non-null  float64
 5   Lagging_Current_Power_Factor          35040 non-null  float64
 6   Leading_Current_Power_Factor          35040 non-null  float64
 7   NSM                                   35040 non-null  int64
 8   WeekStatus                            35040 non-null  object
 9   Day_of_week                           35040 non-null  object
 10  Load_Type                             35040 non-null  object
dtypes: float64(6), int64(1), object(4)
memory usage: 2.9+ MB
```

```
data=data[:-1]
```

```
data.tail()
```

	date	Usage_kWh	Lagging_Current_Reactive.Power_kVarh	Leading_Current_Reactive_Power_kVarh	CO2(tCO2)	Lagging_Current_Powe
35034	31/12/2018 22:45	3.82	4.54	0.00	0.0	
35035	31/12/2018 23:00	3.85	4.86	0.00	0.0	
35036	31/12/2018 23:15	3.74	3.74	0.00	0.0	
35037	31/12/2018 23:30	3.78	3.17	0.07	0.0	
35038	31/12/2018 23:45	3.78	3.06	0.11	0.0	

```
target=data["Usage_kWh"]
data=data.drop(columns="Usage_kWh")
```

Checking for null value

```
data.isnull().sum()
```

```
date                                  0
Lagging_Current_Reactive.Power_kVarh  0
Leading_Current_Reactive_Power_kVarh  0
CO2(tCO2)                             0
Lagging_Current_Power_Factor          0
Leading_Current_Power_Factor          0
NSM                                   0
WeekStatus                            0
Day_of_week                           0
Load_Type                             0
dtype: int64
```

We can extract TIME from data col

```
data["date"]=[x[-5:] for x in data["date"]]
```

```
data.head()
```

	date	Lagging_Current_Reactive.Power_kVarh	Leading_Current_Reactive_Power_kVarh	CC
0	00:15	2.95	0.0	
1	00:30	4.46	0.0	
2	00:45	3.28	0.0	
3	01:00	3.56	0.0	
4	01:15	4.50	0.0	

Start coding or generate with AI.

One Hot Encoding our data

```
data_one_hot=pd.get_dummies(data)
data_one_hot.head()
```

	Lagging_Current_Reactive.Power_kVarh	Leading_Current_Reactive_Power_kVarh	CO2(tCO2)	Lagging_Current_Power_Factor	Leading_Current_
0	2.95	0.0	0.0	73.21	
1	4.46	0.0	0.0	66.77	
2	3.28	0.0	0.0	70.28	
3	3.56	0.0	0.0	68.09	
4	4.50	0.0	0.0	64.72	

5 rows × 114 columns

```
pd.DataFrame(data_one_hot).info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35039 entries, 0 to 35038
Columns: 114 entries, Lagging_Current_Reactive.Power_kVarh to Load_Type_Medium_Load
dtypes: float64(5), int64(1), uint8(108)
memory usage: 5.2 MB
```

Standard Scaler

```
from sklearn.preprocessing import StandardScaler
```

```
ss=StandardScaler()

df=pd.DataFrame(ss.fit_transform(data_one_hot))
```

Train Test Split

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(df,target,random_state=42,test_size=0.3)

x_train.head()
```

	0	1	2	3	4	5	6	7
25270	-0.494647	-0.521394	-0.713566	-1.706972	0.513277	-0.884189	-0.102457	-0.102599
22053	-0.799444	1.863973	-0.713566	1.026453	-2.077967	0.811916	-0.102457	-0.102599
30307	1.823531	-0.521394	1.763107	0.571411	0.513277	0.739742	-0.102457	-0.102599
31403	-0.616075	-0.511966	-0.713566	-0.211833	0.512620	-1.281150	-0.102457	-0.102599
15443	-0.799444	1.699650	-0.713566	1.026453	-2.157753	1.317139	-0.102457	-0.102599

5 rows × 114 columns

```
y_train.head()
```

```
25270     2.74
22053     3.82
30307    96.26
31403     3.56
15443     3.13
Name: Usage_kWh, dtype: float64
```

Start coding or generate with AI.

Linear Regression

```
from sklearn.linear_model import LinearRegression

ll=LinearRegression()
ll.fit(x_train,y_train)
```

```
LinearRegression
LinearRegression()
```

```
y_train_pred=ll.predict(x_train)
y_test_pred=ll.predict(x_test)

from sklearn.metrics import r2_score

r2_score(y_train_pred,y_train),r2_score(y_test_pred,y_test)
```

```
(0.9798182862442881, 0.9839378544022411)
```

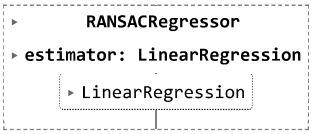
```
from sklearn.metrics import mean_squared_error

mean_squared_error(y_train_pred,y_train),mean_squared_error(y_test_pred,y_test)
```

```
(22.272442050782423, 17.46860990095796)
```

Ransac Regression

```
from sklearn.linear_model import RANSACRegressor
```

```
ransac=RANSACRegressor(LinearRegression(),min_samples=50,residual_threshold=42)
ransac.fit(x_train,y_train)
```




```
y_train_pred_ransac=ransac.predict(x_train)
y_test_pred_ransac=ransac.predict(x_test)
```

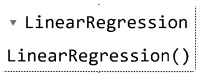
```
r2_score(y_train_pred_ransac,y_train),r2_score(y_test_pred_ransac,y_test)
```

```
(0.9794409874453801, 0.9838274936651664)
```

Polynomial Regression

```
from sklearn.preprocessing import PolynomialFeatures
```

```
pr=LinearRegression()
poly=PolynomialFeatures(degree=2)
x_train_degree_2=poly.fit_transform(x_train)
pr.fit(x_train_degree_2,y_train)
```




```
x_test_degree_2=poly.transform(x_test)
```

```
x_train_poly=pr.predict(x_train_degree_2)
x_test_poly=pr.predict(x_test_degree_2)
```

```
r2_score(x_train_poly,y_train),r2_score(x_test_poly,y_test)
```

```
(0.9976879073938388, 0.9969624103952736)
```

Start coding or generate with AI.