In [ ]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

In [ ]:

```
## DAY DATASET ##
```

In [ ]:

```
df = pd.read_csv('/content/day.csv')
df.columns = ["instant","dteday","season","yr","mnth","holiday","weekday","workingday","weathersit","temp","atemp",
df
```

Out[ ]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.226642	0.652917	0.350133	247
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255046	0.590000	0.155471	644
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.242400	0.752917	0.124383	159
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.231700	0.483333	0.350754	364

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.223487	0.577500	0.154846	439

In [ ]:
```
df.corr()['cnt']
```

Out[ ]:
```
instant       0.628830
season        0.406100
yr            0.566710
mnth          0.279977
holiday      -0.068348
weekday       0.067443
workingday    0.061156
weathersit   -0.297391
temp          0.627494
atemp         0.631066
hum          -0.100659
windspeed    -0.234545
casual        0.672804
registered    0.945517
cnt           1.000000
Name: cnt, dtype: float64
```

In [ ]:
```
x = df.drop('cnt',axis=1)
x = df.drop('dteday',axis=1)
x_train, x_test, y_train, y_test = train_test_split(x, df['cnt'], test_size=0.15, random_state=1)
print('Test Samples:- ',x_test.shape[0])
print('Train Samples:- ',x_train.shape[0])
```

```
Test Samples:-  110
Train Samples:-  621
```

In [ ]:
```
lr = LinearRegression()
Z = x_train[['instant', 'season', 'yr', 'temp','atemp','casual','registered']]
lr.fit(Z, y_train)
print(lr.intercept_)
print(lr.coef_)
```

```
9.094947017729282e-13
[-4.37592630e-15 -6.08402217e-14  1.35078336e-12  1.49990859e-12
 -1.67432452e-12  1.00000000e+00  1.00000000e+00]
```

In [ ]:
```
y = lr.predict(x_test[['instant', 'season', 'yr', 'temp','atemp','casual','registered']])
y
```

Out[ ]:
```
array([3830., 2114., 3915., 4322., 6591., 5204., 5323., 3351., 3068.,
       4972., 3351., 1977.,  822., 2485., 7736., 2368., 5515., 6572.,
       4511., 4359., 1851., 7335., 4758., 2689., 4966., 4118., 6824.,
       4375., 3389., 2475.,  754., 3523., 1683., 1749., 6241., 4570.,
       1416., 3368., 6591., 4023., 4553., 7424., 4840., 5728., 5202.,
       4717., 7338., 4035., 2277., 4866., 2034., 7693., 3403., 1812.,
       4304., 2947., 1969., 7006., 4656., 1927., 2376., 2134., 5084.,
       2121., 4154., 8294., 4592., 5319., 1685., 3204., 7261., 6825.,
       5035., 1562., 7444., 7415., 7765., 4507., 1872., 3310., 4826.,
       3376., 7570., 2402., 5267., 3659., 3126., 4150., 6864., 3574.,
       1005., 5058., 5138., 5342., 3005., 1746., 1000., 6891., 4665.,
       1341., 3577.,  985., 4367., 5047., 6192., 2192., 2594., 2169.,
       2133., 2302.])
```

In [ ]:
```
inp=[('scale',StandardScaler()), ('polynomial', PolynomialFeatures(7)), ('model',LinearRegression())]
pipe=Pipeline(inp)
pipe.fit(x_train[['instant', 'season', 'yr', 'temp','atemp','casual','registered']],y_train)
```

Out[ ]:
```
Pipeline(steps=[('scale', StandardScaler()),
                ('polynomial', PolynomialFeatures(degree=7)),
                ('model', LinearRegression())])
```

In [ ]:
```
yPoly = pipe.predict(x_test[['instant', 'season', 'yr', 'temp','atemp','casual','registered']])
yPoly
```

Out[ ]:
```
array([3829.42847728, 2028.99811303, 3914.26137998, 4322.48772538,
       7056.52338316, 5204.44983582, 5318.12258821, 3344.31213494,
       3037.89759813, 4925.432554  , 3350.68186584, 1977.83032558,
        836.92224486, 2579.68354915, 7734.04542893, 2417.63845855,
       5520.19269826, 6574.97991822, 4489.51925555, 4408.27129326,
       1849.06936732, 7336.92264597, 4758.49981841, 2690.01687249,
       4955.66629281, 4066.68438968, 6854.13286931, 4373.99227992,
       3384.54644522, 2498.18195886,  656.61514575, 3521.03040124,
       1679.79488492, 1504.81695341, 6230.19898626, 4567.78918838,
       1396.6924309 , 3366.47518228, 6594.61545036, 4016.84094112,
       4549.26939784, 7412.48267103, 4941.98850267, 5726.60033323,
       5197.54178741, 4748.5135965 , 7331.52287635, 4034.55114496,
       2297.92596694, 4866.20466324, 2039.27881723, 7686.48616185,
```

```
3403.29641993, 1814.44871518, 4304.95029322, 2946.01164577,
1967.18770682, 7005.89620843, 4657.57765694, 1929.710691  ,
2374.45846503, 2135.04136446, 5086.79107655, 2115.73189069,
4153.95853664, 8095.43651603, 4591.97276598, 5320.47223671,
1683.67487035, 3204.18367606, 7262.96095655, 6834.38702465,
5039.3807564 , 1562.49771119, 7445.61616503, 7426.56391564,
7765.40374956, 4509.69569714, 1875.49580058, 3307.9533895 ,
4825.17993142, 3376.59424398, 7584.62374247, 2395.39399507,
5267.13636652, 3649.86422837, 3128.24333571, 4149.89220708,
6865.67694864, 3595.28238584, 1001.12113665, 5058.35318099,
5134.26928377, 5347.92063677, 3012.84940518, 1749.65178423,
1000.44009962, 6920.44860228, 4661.32522137, 1018.21534091,
3575.51313581, 1016.64353826, 4393.97122967, 5040.85859264,
6191.40804062, 2192.92213622, 2591.38451263, 2155.67789681,
```

In [ ]:

```
## HOUR DATASET ##
```

In [ ]:

```
dfh = pd.read_csv('/content/hour.csv')
dfh.columns = ["instant","dteday","season","yr","mnth","hr","holiday","weekday","workingday","weathersit","temp","a
dfh
```

Out[ ]:

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	r
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0000	3	
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0000	8	
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0000	5	
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0000	3	
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0000	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
17374	17375	2012-12-31	1	1	12	19	0	1	1	2	0.26	0.2576	0.60	0.1642	11	
17375	17376	2012-12-31	1	1	12	20	0	1	1	2	0.26	0.2576	0.60	0.1642	8	
17376	17377	2012-12-31	1	1	12	21	0	1	1	1	0.26	0.2576	0.60	0.1642	7	
17377	17378	2012-12-31	1	1	12	22	0	1	1	1	0.26	0.2727	0.56	0.1343	13	
17378	17379	2012-12-31	1	1	12	23	0	1	1	1	0.26	0.2727	0.65	0.1343	12	

17379 rows × 17 columns

In [ ]:
```
dfh.corr()['cnt']
```

Out[ ]:
```
instant       0.278379
season        0.178056
yr            0.250495
mnth          0.120638
hr            0.394071
holiday      -0.030927
weekday       0.026900
workingday    0.030284
weathersit   -0.142426
temp          0.404772
atemp         0.400929
hum          -0.322911
windspeed     0.093234
casual        0.694564
registered    0.972151
cnt           1.000000
Name: cnt, dtype: float64
```

In [ ]:
```
x1_data = dfh.drop('cnt',axis=1)
x1_train, x1_test, y1_train, y1_test = train_test_split(x1_data, dfh['cnt'], test_size=0.15, random_state=1)
print('Test Samples:- ',x1_test.shape[0])
print('Train Samples:- ',x1_train.shape[0])
```

```
Test Samples:-  2607
Train Samples:-  14772
```

In [ ]:
```
lr1 = LinearRegression()
Z1 = x1_train[['hr', 'temp','atemp','hum','casual','registered']]
lr1.fit(Z1, y1_train)
print(lr1.intercept_)
print(lr1.coef_)
```

```
0.0
[ 3.12152063e-15  9.61453139e-14 -2.33018195e-13  6.78465530e-14
  1.00000000e+00  1.00000000e+00]
```

In [ ]:
```
y2=lr1.predict(x1_test[['hr', 'temp','atemp','hum','casual','registered']])
print(y2)
```

```
[389. 146. 152. ...   7. 285.  33.]
```

In [ ]:
```
inp1=[('scale',StandardScaler()), ('polynomial', PolynomialFeatures(7)), ('model',LinearRegression())]
pipe1=Pipeline(inp1)
pipe1.fit(x1_train[['hr', 'temp','atemp','hum','casual','registered']],y1_train)
```

Out[ ]:
```
Pipeline(steps=[('scale', StandardScaler()),
                ('polynomial', PolynomialFeatures(degree=7)),
                ('model', LinearRegression())])
```

In [ ]:
```
yPoly2=pipe1.predict(x1_test[['hr', 'temp','atemp','hum','casual','registered']])
print(yPoly2)
```

```
[389. 146. 152. ...   7. 285.  33.]
```

In [ ]:
```
print("Mutli-linear:-")
print('R2: ', r2_score(y2,y1_test))
print('MSE: ', mean_squared_error(y2,y1_test))
print()
print("Multi-polynomial:-")
print('R2: ',r2_score(yPoly2,y1_test))
print('MSE:',mean_squared_error(yPoly2,y1_test))
```

```
Mutli-linear:-
R2:  1.0
MSE:  2.156654875400162e-27

Multi-polynomial:-
R2:  1.0
MSE: 1.3107261237333296e-15
```

In [ ]:
```
print("Mutli-linear:-")
print('R2: ', r2_score(y,y_test))
print('MSE: ', mean_squared_error(y,y_test))
print()
print("Multi-polynomial:- ")
print('R2: ',r2_score(yPoly,y_test))
print('MSE:',mean_squared_error(yPoly,y_test))
```

```
Mutli-linear:-
R2:  1.0
MSE:  5.303825689011885e-25

Multi-polynomial:-
R2:  0.9988511903187917
MSE: 4327.126389286028
```

In [ ]:
```
'''
Conclusion:-
    - The value of R2 is same in both.
'''
```

Out[ ]:
```
'\nConclusion:-\n    - The value of R2 is same in both.\n'
```