Regression

February 13, 2022

1 Regression

[19]:
```
import pandas as pd
from pandas import DataFrame
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import numpy as np
from sklearn.metrics import mean_squared_error
```

[2]:
```
df_dataset = pd.read_excel("petrol_consumption.xlsx")
```

[3]:
```
df_dataset
```

[3]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	\
0	9.00	3571	1976	0.525	
1	9.00	4092	1250	0.572	
2	9.00	3865	1586	0.580	
3	7.50	4870	2351	0.529	
4	8.00	4399	431	0.544	
5	10.00	5342	1333	0.571	
6	8.00	5319	11868	0.451	
7	8.00	5126	2138	0.553	
8	8.00	4447	8577	0.529	
9	7.00	4512	8507	0.552	
10	8.00	4391	5939	0.530	
11	7.50	5126	14186	0.525	
12	7.00	4817	6930	0.574	
13	7.00	4207	6580	0.545	
14	7.00	4332	8159	0.608	
15	7.00	4318	10340	0.586	
16	7.00	4206	8508	0.572	
17	7.00	3718	4725	0.540	
18	7.00	4716	5915	0.724	
19	8.50	4341	6010	0.677	
20	7.00	4593	7834	0.663	
21	8.00	4983	602	0.602	
22	9.00	4897	2449	0.511	

23	9.00	4258	4686	0.517
24	8.50	4574	2619	0.551
25	9.00	3721	4746	0.544
26	8.00	3448	5399	0.548
27	7.50	3846	9061	0.579
28	8.00	4188	5975	0.563
29	9.00	3601	4650	0.493
30	7.00	3640	6905	0.518
31	7.00	3333	6594	0.513
32	8.00	3063	6524	0.578
33	7.50	3357	4121	0.547
34	8.00	3528	3495	0.487
35	6.58	3802	7834	0.629
36	5.00	4045	17782	0.566
37	7.00	3897	6385	0.586
38	8.50	3635	3274	0.663
39	7.00	4345	3905	0.672
40	7.00	4449	4639	0.626
41	7.00	3656	3985	0.563
42	7.00	4300	3635	0.603
43	7.00	3745	2611	0.508
44	6.00	5215	2302	0.672
45	9.00	4476	3942	0.571
46	7.00	4296	4083	0.623
47	7.00	5002	9794	0.593

	Petrol_Consumption
0	541
1	524
2	561
3	414
4	410
5	457
6	344
7	467
8	464
9	498
10	580
11	471
12	525
13	508
14	566
15	635
16	603
17	714
18	865
19	640

20	649
21	540
22	464
23	547
24	460
25	566
26	577
27	631
28	574
29	534
30	571
31	554
32	577
33	628
34	487
35	644
36	640
37	704
38	648
39	968
40	587
41	699
42	632
43	591
44	782
45	510
46	610
47	524

[4]:
```
y = df_dataset["Petrol_Consumption"]
X = df_dataset.drop(["Petrol_Consumption"], axis = 1)
```

[5]:
```
X
```

[5]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)
0	9.00	3571	1976	0.525
1	9.00	4092	1250	0.572
2	9.00	3865	1586	0.580
3	7.50	4870	2351	0.529
4	8.00	4399	431	0.544
5	10.00	5342	1333	0.571
6	8.00	5319	11868	0.451
7	8.00	5126	2138	0.553
8	8.00	4447	8577	0.529
9	7.00	4512	8507	0.552
10	8.00	4391	5939	0.530
11	7.50	5126	14186	0.525

12	7.00	4817	6930	0.574
13	7.00	4207	6580	0.545
14	7.00	4332	8159	0.608
15	7.00	4318	10340	0.586
16	7.00	4206	8508	0.572
17	7.00	3718	4725	0.540
18	7.00	4716	5915	0.724
19	8.50	4341	6010	0.677
20	7.00	4593	7834	0.663
21	8.00	4983	602	0.602
22	9.00	4897	2449	0.511
23	9.00	4258	4686	0.517
24	8.50	4574	2619	0.551
25	9.00	3721	4746	0.544
26	8.00	3448	5399	0.548
27	7.50	3846	9061	0.579
28	8.00	4188	5975	0.563
29	9.00	3601	4650	0.493
30	7.00	3640	6905	0.518
31	7.00	3333	6594	0.513
32	8.00	3063	6524	0.578
33	7.50	3357	4121	0.547
34	8.00	3528	3495	0.487
35	6.58	3802	7834	0.629
36	5.00	4045	17782	0.566
37	7.00	3897	6385	0.586
38	8.50	3635	3274	0.663
39	7.00	4345	3905	0.672
40	7.00	4449	4639	0.626
41	7.00	3656	3985	0.563
42	7.00	4300	3635	0.603
43	7.00	3745	2611	0.508
44	6.00	5215	2302	0.672
45	9.00	4476	3942	0.571
46	7.00	4296	4083	0.623
47	7.00	5002	9794	0.593

[6]:
```
y
```

[6]:
```
0     541
1     524
2     561
3     414
4     410
5     457
6     344
7     467
```

```
8     464
9     498
10    580
11    471
12    525
13    508
14    566
15    635
16    603
17    714
18    865
19    640
20    649
21    540
22    464
23    547
24    460
25    566
26    577
27    631
28    574
29    534
30    571
31    554
32    577
33    628
34    487
35    644
36    640
37    704
38    648
39    968
40    587
41    699
42    632
43    591
44    782
45    510
46    610
47    524
Name: Petrol_Consumption, dtype: int64
```

[7]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,␣
↪random_state=43)
```

[9]:
```
len(X_train)
```

[9]:
```
38
```

[10]:
```
len(y_train)
```

[10]:
```
38
```

[11]:
```
len(X_test)
```

[11]:
```
10
```

[12]:
```
len(y_test)
```

[12]:
```
10
```

[14]:
```
rf_regressor = RandomForestRegressor().fit(X_train, y_train)
```

[15]:
```
y_pred = rf_regressor.predict(X_test)
```

[21]:
```
y_pred
```

[21]:
```
array([473.19, 594.24, 631.35, 621.46, 625.69, 564.11, 759.01, 613.81,
       446.27, 521.44])
```

[22]:
```
y_test
```

[22]:
```
6     344
26    577
33    628
13    508
14    566
34    487
18    865
31    554
22    464
5     457
Name: Petrol_Consumption, dtype: int64
```

[20]:
```
rmse = float(format(np.sqrt(mean_squared_error(y_test, y_pred)), '.3f'))

print("\nRMSE: ", rmse)
```

```
RMSE:  76.589
```