ML 6 - Standardization

December 1, 2023

1 Standardization

In Standardization we scale down the values of dataset within the range or decrease the values of datapoints within the range and the output we get before or after the scale down is SAME , the only region for scale down the datapoints is just for increase the accuracy of ML model or enhance the model

Case 1 -Implementing STANDARDIZATION MANUALLY

[2]:
```
# Here we import seaborn for importing the dataset . seaborn provides lot of␣
↪built in dataset

import seaborn as sns
```

[3]:
```
# This is how we import the dataset
# Tips dataset is a restrount dataset that contain lot of features like -␣
↪total_bill , sex , smoker etc

df=sns.load_dataset("tips")
```

[4]:
```
# .head (  )  function gives the initial 5 datapoints from the entire dataset

df.head()
```

[4]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

[5]:
```
total_bill=list(df["total_bill"])
```

[12]:
```
total_bill
```

[12]:
```
[16.99,
 10.34,
 21.01,
```

23.68,
24.59,
25.29,
8.77,
26.88,
15.04,
14.78,
10.27,
35.26,
15.42,
18.43,
14.83,
21.58,
10.33,
16.29,
16.97,
20.65,
17.92,
20.29,
15.77,
39.42,
19.82,
17.81,
13.37,
12.69,
21.7,
19.65,
9.55,
18.35,
15.06,
20.69,
17.78,
24.06,
16.31,
16.93,
18.69,
31.27,
16.04,
17.46,
13.94,
9.68,
30.4,
18.29,
22.23,
32.4,
28.55,
18.04,

```
12.54,
10.29,
34.81,
9.94,
25.56,
19.49,
38.01,
26.41,
11.24,
48.27,
20.29,
13.81,
11.02,
18.29,
17.59,
20.08,
16.45,
3.07,
20.23,
15.01,
12.02,
17.07,
26.86,
25.28,
14.73,
10.51,
17.92,
27.2,
22.76,
17.29,
19.44,
16.66,
10.07,
32.68,
15.98,
34.83,
13.03,
18.28,
24.71,
21.16,
28.97,
22.49,
5.75,
16.32,
22.75,
40.17,
27.28,
```

12.03,
21.01,
12.46,
11.35,
15.38,
44.3,
22.42,
20.92,
15.36,
20.49,
25.21,
18.24,
14.31,
14.0,
7.25,
38.07,
23.95,
25.71,
17.31,
29.93,
10.65,
12.43,
24.08,
11.69,
13.42,
14.26,
15.95,
12.48,
29.8,
8.52,
14.52,
11.38,
22.82,
19.08,
20.27,
11.17,
12.26,
18.26,
8.51,
10.33,
14.15,
16.0,
13.16,
17.47,
34.3,
41.19,
27.05,

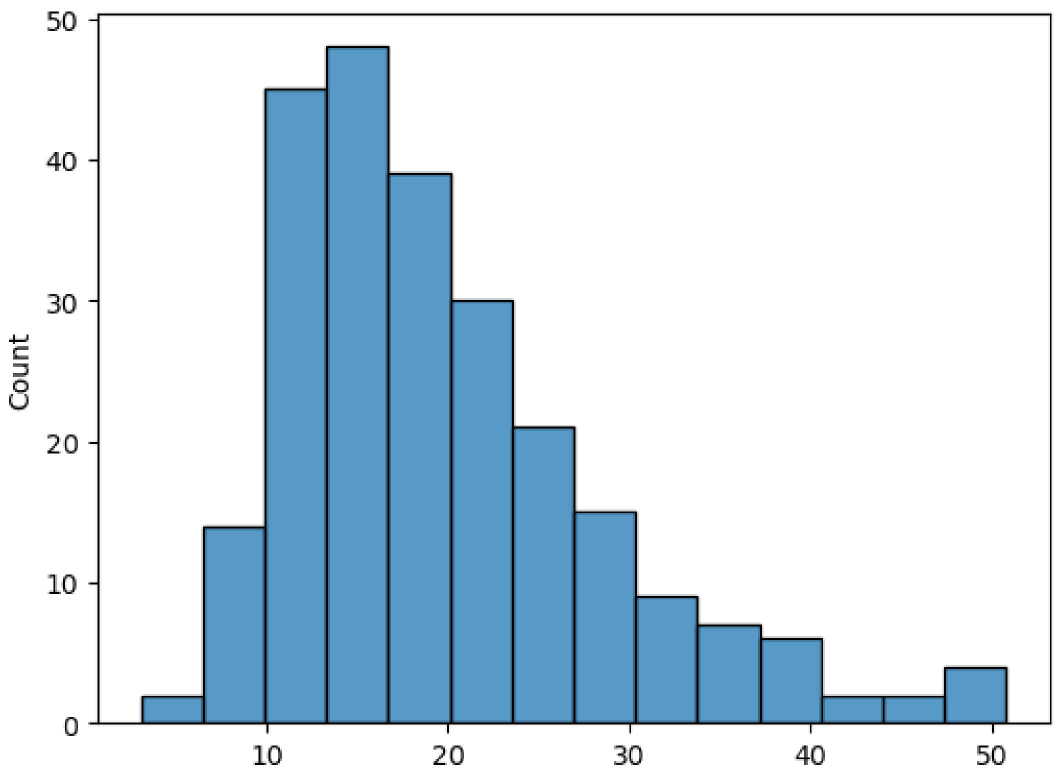
```
16.43,
8.35,
18.64,
11.87,
9.78,
7.51,
14.07,
13.13,
17.26,
24.55,
19.77,
29.85,
48.17,
25.0,
13.39,
16.49,
21.5,
12.66,
16.21,
13.81,
17.51,
24.52,
20.76,
31.71,
10.59,
10.63,
50.81,
15.81,
7.25,
31.85,
16.82,
32.9,
17.89,
14.48,
9.6,
34.63,
34.65,
23.33,
45.35,
23.17,
40.55,
20.69,
20.9,
30.46,
18.15,
23.1,
15.69,
```

```
19.81,
28.44,
15.48,
16.58,
7.56,
10.34,
43.11,
13.0,
13.51,
18.71,
12.74,
13.0,
16.4,
20.53,
16.47,
26.59,
38.73,
24.27,
12.76,
30.06,
25.89,
48.33,
13.27,
28.17,
12.9,
28.15,
11.59,
7.74,
30.14,
12.16,
13.42,
8.58,
15.98,
13.42,
16.27,
10.09,
20.45,
13.28,
22.12,
24.01,
15.69,
11.61,
10.77,
15.53,
10.07,
12.6,
32.83,
```

```
35.83,
29.03,
27.18,
22.67,
17.82,
18.78]
```

[13]:
```
sns.histplot(total_bill)
```

[13]: <AxesSubplot: ylabel='Count'>



[6]:
```
# import numpy as np for mathematical calucaltion
# np.mean ( ) function is used for getting the mean
# np.std is used for getting the standard deviation
# we find MEAND and  STANDARD DEVIATION becouse they use in z- score when we␣
↪find the STANDARDIZATION

import numpy as np
mean=np.mean(total_bill)
std=np.std(total_bill)
```

[7]:
```
mean,std
```

[7]: (19.78594262295082, 8.884150577771132)

[8]:
```
# This is how we implemnt the z - score
# With the help of the z - score we scale the data within the same range
# Here we get the values of "total_bill" , the only differece is that here we␣
↪reduce the range of value otherwise evrything is ame including the plot
# We simply scale down the "total_bill" within the same range

normalised_data=[]
for i in total_bill:
    z_score=(i-mean)/std # z - score
    normalised_data.append(z_score)
print(normalised_data)
```

```
[-0.3147113050904943, -1.0632353132988692, 0.13777989987156145,
0.43831510316725475, 0.540744704290506, 0.6195367051545455, -1.2399545152367863,
0.7985071071171495, -0.5342033074974614, -0.5634689078183903,
-1.0711145133852733, 1.7417599174609364, -0.49143050702841123,
-0.15262490331304146, -0.557840907756673, 0.2019391005751361,
-1.0643609133112126, -0.3935033059545337, -0.31696250511518104,
0.09725829942719795, -0.2100305039425557, 0.05673669898283484,
-0.45203450659639155, 2.2100095225958003, 0.003833498402694168,
-0.222412104078337, -0.7221785095588127, -0.7987193103981653,
0.21544630072325727, -0.015301701807144186, -1.1521577142739994,
-0.16162970341178864, -0.5319521074727743, 0.10176069947657193,
-0.22578890411536368, 0.4810879036363046, -0.3912521059298469,
-0.32146490516455467, -0.12335930299211233, 1.2926455125359115,
-0.4216433062631192, -0.2618081045103532, -0.6580193088552376,
-1.137524914113535, 1.1947183114620337, -0.16838330348584943,
0.2751031013774587, 1.419838313930718, 0.986482309178501, -0.19652330379443494,
-0.8156033105833167, -1.0688633133605865, 1.691107916905483,
-1.1082593137926062, 0.6499279054878179, -0.03331130200463894,
2.051299920855377, 0.7456039065370088, -0.9619313121879614, 3.206165533519728,
0.05673669898283484, -0.672652109015702, -0.9866945124595167,
-0.16838330348584943, -0.24717530434988882, 0.0330990987236229,
-0.37549370575703894, -1.8815465222725365, 0.049983098908774455,
-0.5375801075344916, -0.8741345112251745, -0.3057065049917467,
0.7962559070924626, 0.6184111051422023, -0.5690969078801073, -1.044100113089031,
-0.2100305039425557, 0.834526307512139, 0.3347599020316602, -0.2809433047201916,
-0.03893930206635573, -0.351856105497827, -1.0936265136321417,
1.451355114276334, -0.4283969063371796, 1.6933591169301694, -0.760448909978489,
-0.16950890349819261, 0.5542519044386273, 0.1546639000567126,
1.0337575096969245, 0.3043687016983874, -1.5798857189644997,
-0.3901265059175033, 0.3336343020193166, 2.294429523521557, 0.8435311076108866,
-0.8730089112128311, 0.13777989987156145, -0.8246081106820639,
-0.9495497120521837, -0.4959329070777848, 2.759302328619389,
0.29648950161198384, 0.12764949976047066, -0.49818410710247185,
```

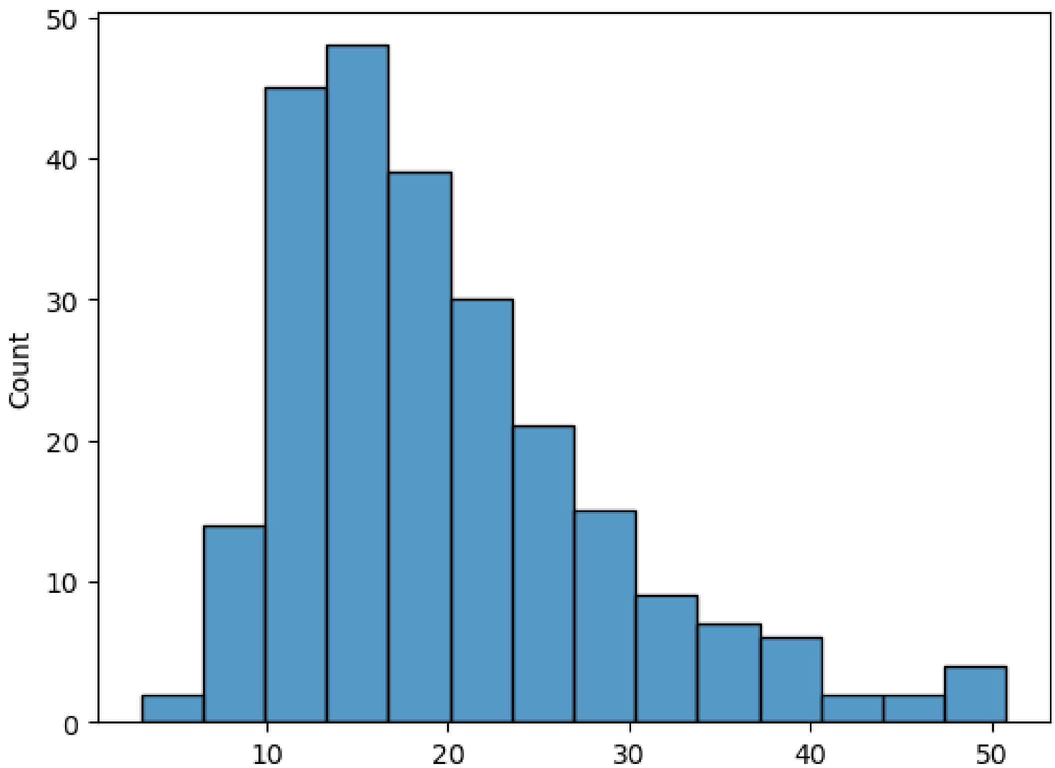
```
0.07924869922970319, 0.6105319050557984, -0.1740113035475666,
-0.616372108398531, -0.6512657087811771, -1.4110457171129864, 2.058053520929438,
0.46870630350052706, 0.6668119056729693, -0.27869210469550476,
1.141815110881893, -1.0283417129162231, -0.8279849107190942, 0.4833391036609914,
-0.9112793116325074, -0.7165505094970955, -0.6220001084602481,
-0.43177370637421, -0.822356910657377, 1.1271823107214287, -1.268094515545372,
-0.5927345081393192, -0.9461729120151534, 0.34151350210572057,
-0.07946090251071923, 0.05448549895814805, -0.9698105122743653,
-0.8471201109289324, -0.1717601035228794, -1.2692201155577154,
-1.0643609133112126, -0.6343817085960257, -0.4261457063124928,
-0.7458161098180245, -0.26068250449801, 1.6337023162759678, 2.4092407247805854,
0.8176423073269878, -0.3777449057817257, -1.2872297157552102,
-0.12898730305382952, -0.8910185114103258, -1.126268913990101,
-1.3817801167920576, -0.6433865086947731, -0.7491929098550546,
-0.2843201047572215, 0.5362423042411325, -0.0017945016590230187,
1.1328103107831458, 3.194909533396294, 0.5868943047965863, -0.7199273095341256,
-0.3709913057076653, 0.1929343004763889, -0.8020961104351955,
-0.40250810605328086, -0.672652109015702, -0.25618010444863604,
0.5328655042041021, 0.10963989956297591, 1.3421719130790222,
-1.0350953129902838, -1.03059291294091, 3.492067936654957, -0.44753210654701775,
-1.4110457171129864, 1.3579303132518301, -0.3338465053003322, 1.476118314547889,
-0.2134073039795861, -0.5972369081886928, -1.1465297142122826,
1.6708471166833014, 1.6730983167079878, 0.39891910273523484, 2.877490329915449,
0.3809095025377405, 2.3372023239906063, 0.10176069947657193,
0.12539829973578348, 1.2014719115360946, -0.18414170365865737,
0.3730303024513365, -0.46103930669513893, 0.0027078983903505707,
0.9741007090427235, -0.4846769069543507, -0.3608609055965746,
-1.3761521167303405, -1.0632353132988692, 2.6253559271505225,
-0.7638257100155192, -0.7064201093860047, -0.12110810296742554,
-0.7930913103364481, -0.7638257100155192, -0.3811217058187561,
0.08375109927907717, -0.3732425057323521, 0.7658647067591904,
2.1323431217441033, 0.5047255038955165, -0.7908401103117614, 1.1564479110423573,
0.6870727058951509, 3.212919133593788, -0.7334345096822469, 0.9437095087094511,
-0.7750817101389533, 0.9414583086847639, -0.9225353117559416,
-1.3558913165081588, 1.165452711141105, -0.8583761110523666,
-0.7165505094970955, -1.2613409154713113, -0.4283969063371796,
-0.7165505094970955, -0.39575450597922046, -1.0913753136074549,
0.0747462991803296, -0.7323089096699035, 0.26272150124168114,
0.47545990357458784, -0.46103930669513893, -0.9202841117312548,
-1.0148345127681022, -0.4790489068926337, -1.0936265136321417,
-0.8088497105092561, 1.468239114461485, 1.8059191181645116, 1.0405111097709854,
0.8322751074874521, 0.3246295019205694, -0.2212865040659901,
-0.11322890288102155]
```

[10]:

```
# Range of x axis decreases after applying the z - score

sns.histplot(total_bill)
```

[10]: <AxesSubplot: ylabel='Count'>



Case 2 - Implement Standardization using " StandardScaler "

[14]:
```
# for implementing "StandardScaler" we have to use "from sklearn.preprocessing"

from sklearn.preprocessing import StandardScaler
```

[15]:
```
scaler=StandardScaler()
```

[17]:
```
scaler
```

[17]: StandardScaler()

[19]:
```
scaler.fit(df[["total_bill","tip"]])
```

[19]: StandardScaler()

[21]:
```
# Here by using the pandas we store the features in proper manner

import pandas as pd
pd.DataFrame(scaler.transform(df[["total_bill","tip"]]))
```

```
[21]:             0         1
      0   -0.314711 -1.439947
      1   -1.063235 -0.969205
      2    0.137780  0.363356
      3    0.438315  0.225754
      4    0.540745  0.443020
      ..        …         …
      239  1.040511  2.115963
      240  0.832275 -0.722971
      241  0.324630 -0.722971
      242 -0.221287 -0.904026
      243 -0.113229  0.001247

      [244 rows x 2 columns]
```

THANK YOU SO MUCH !!

YOURS VIRAT TIWARI :)