Welcome to 30 Days ML | Day 18

Import Library

In [1]:

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

Import Dataset

In [2]:

```
df = pd.read_csv('placement.csv')
```

In [3]:

```
df
```

Out[3]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...	...	...	...
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
998	8.62	46.0	1
999	4.90	10.0	1

1000 rows × 3 columns

In [4]:

```
df.head()
```

Out[4]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0

In [5]:
```
df.sample(5)
```

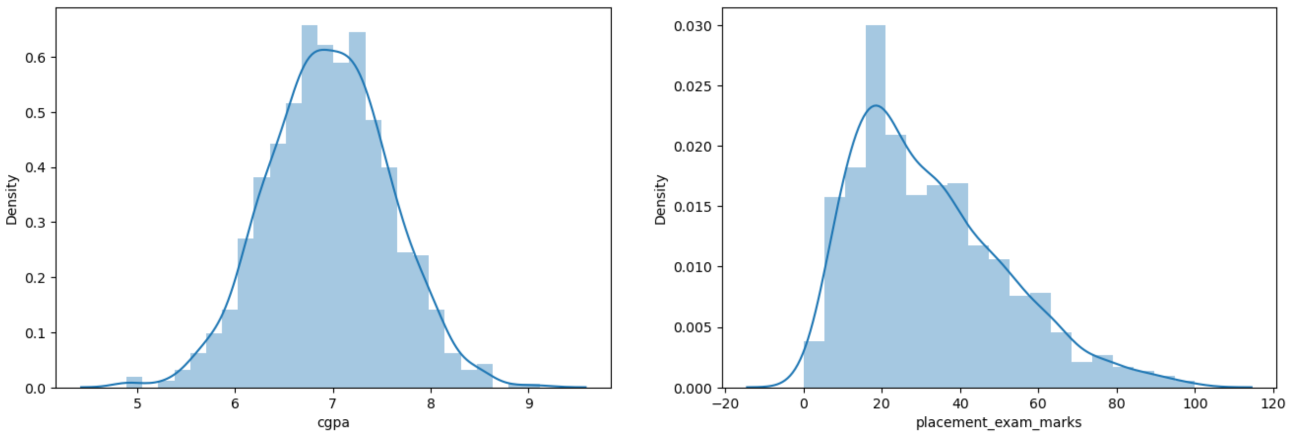
Out[5]:

	cgpa	placement_exam_marks	placed
655	7.36	34.0	0
339	7.32	18.0	1
274	7.13	4.0	1
138	7.53	8.0	1
83	7.38	20.0	1

Plot Show in CGPA and Placement Marks

In [6]:
```
plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(df['cgpa'])
plt.subplot(1,2,2)
sns.distplot(df['placement_exam_marks'])
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_11704\4260214945.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.githu
b.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df['cgpa'])
C:\Users\ASUS\AppData\Local\Temp\ipykernel_11704\4260214945.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.githu
b.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df['placement_exam_marks'])
```



Print Mean | Std | Min & Max Value

In [7]:

```
print("Mean value of cgpa",df['cgpa'].mean())
print("Std value of cgpa",df['cgpa'].std())
print("Min value of cgpa",df['cgpa'].min())
print("Max value of cgpa",df['cgpa'].max())
```

```
Mean value of cgpa 6.96124
Std value of cgpa 0.6158978751323894
Min value of cgpa 4.89
Max value of cgpa 9.12
```

Approach 1

Step 1:Finding the boundary values (Highest & Lowest)

In [8]:

```
print("Highest allowed",df['cgpa'].mean() + 3*df['cgpa'].std())
print("Lowest allowed",df['cgpa'].mean() - 3*df['cgpa'].std())
```

```
Highest allowed 8.808933625397168
Lowest allowed 5.113546374602832
```

Step 2: Finding the outliers

In [9]:

```
df[(df['cgpa'] > 8.80) | (df['cgpa'] < 5.11)]
```

Out[9]:

	cgpa	placement_exam_marks	placed
485	4.92	44.0	1
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
999	4.90	10.0	1

Step 3: Treat Outliers with Trimming

In [10]:

```
new_df = df[(df['cgpa'] < 8.80) & (df['cgpa'] > 5.11)]
new_df
```

Out[10]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...	...	...	...
991	7.04	57.0	0
992	6.26	12.0	0
993	6.73	21.0	1
994	6.48	63.0	0
998	8.62	46.0	1

995 rows × 3 columns

Approach 2 : Calculating the Zscore

In [12]:

```
df['cgpa_zscore'] = (df['cgpa'] - df['cgpa'].mean())/df['cgpa'].std()
```

In [13]:

```
df
```

Out[13]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26.0	1	0.371425
1	7.46	38.0	1	0.809810
2	7.54	40.0	1	0.939701
3	6.42	8.0	1	-0.878782
4	7.23	17.0	0	0.436371
...	...	...	...	...
995	8.87	44.0	1	3.099150
996	9.12	65.0	1	3.505062
997	4.89	34.0	0	-3.362960
998	8.62	46.0	1	2.693239
999	4.90	10.0	1	-3.346724

1000 rows × 4 columns

CGPA Score More then 3

In [14]:

```
df[df['cgpa_zscore'] > 3]
```

Out[14]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
995	8.87	44.0	1	3.099150
996	9.12	65.0	1	3.505062

CGPA Score Less then 3

In [15]:

```
df[df['cgpa_zscore'] < -3]
```

Out[15]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
485	4.92	44.0	1	-3.314251
997	4.89	34.0	0	-3.362960
999	4.90	10.0	1	-3.346724

Show or Merge Both CGPA

In [16]:

```
df[(df['cgpa_zscore'] > 3) | (df['cgpa_zscore'] < -3)]
```

Out[16]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
485	4.92	44.0	1	-3.314251
995	8.87	44.0	1	3.099150
996	9.12	65.0	1	3.505062
997	4.89	34.0	0	-3.362960
999	4.90	10.0	1	-3.346724

Apply Trimming

In [17]:

```
new_df = df[(df['cgpa_zscore'] < 3) & (df['cgpa_zscore'] > -3)]
```

In [18]:
```
new_df
```

Out[18]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26.0	1	0.371425
1	7.46	38.0	1	0.809810
2	7.54	40.0	1	0.939701
3	6.42	8.0	1	-0.878782
4	7.23	17.0	0	0.436371
...	...	...	...	...
991	7.04	57.0	0	0.127878
992	6.26	12.0	0	-1.138565
993	6.73	21.0	1	-0.375452
994	6.48	63.0	0	-0.781363
998	8.62	46.0	1	2.693239

995 rows × 4 columns

In [ ]:
```

```