In [ ]:
```
NAME : SHINDE SHUBHAM DNYANDEV,     ROLL NO. : EN23107121,    BATCH : C
```

In [75]:
```
import pandas as pd
import numpy as np
```

In [171…
```
df = pd.read_csv("/home/admin1/Downloads/Acadamics.csv")
df
```

Out[171…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	NaN	75	300.0	NaN
4	5	Manish	12	NaN	68	70.0	54.0
5	6	Sham	64	NaN	15	70.0	81.0
6	7	Virat	76	NaN	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	NaN	36	76.0	NaN
11	12	Abhishek	10	41.0	423	NaN	56.0
12	13	Omkar	48	23.0	59	NaN	45.0
13	14	Adesh	64	46.0	71	NaN	NaN
14	15	Anup	31	79.0	48	62.0	65.0

In [173…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 7 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Roll No  15 non-null     int64  
 1   Name     15 non-null     object 
 2   DS       15 non-null     int64  
 3   NoT      10 non-null     float64
 4   DM       15 non-null     int64  
 5   DT       12 non-null     float64
 6   Mdm      12 non-null     float64
dtypes: float64(3), int64(3), object(1)
memory usage: 968.0+ bytes
```

In [175…
```
df.describe()
```

Out[175…

	Roll No	DS	NoT	DM	DT	Mdm
count	15.000000	15.000000	10.000000	15.000000	12.000000	12.000000
mean	8.000000	90.133333	49.300000	75.400000	77.416667	110.833333
std	4.472136	128.185617	15.542415	99.271489	74.604970	201.261853
min	1.000000	10.000000	23.000000	3.000000	5.000000	6.000000
25%	4.500000	46.500000	41.250000	36.000000	40.000000	45.000000
50%	8.000000	64.000000	47.500000	59.000000	70.000000	55.000000
75%	11.500000	78.000000	54.750000	72.500000	76.500000	68.250000
max	15.000000	545.000000	79.000000	423.000000	300.000000	747.000000

In [177…

```
df.isnull()
```

Out[177…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	True	False	False	True
4	False	False	False	True	False	False	False
5	False	False	False	True	False	False	False
6	False	False	False	True	False	False	False
7	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False
10	False	False	False	True	False	False	True
11	False	False	False	False	False	True	False
12	False	False	False	False	False	True	False
13	False	False	False	False	False	True	True
14	False	False	False	False	False	False	False

In [179…

```
df.notnull()
```

Out[179…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True
3	True	True	True	False	True	True	False
4	True	True	True	False	True	True	True
5	True	True	True	False	True	True	True
6	True	True	True	False	True	True	True
7	True	True	True	True	True	True	True
8	True	True	True	True	True	True	True
9	True	True	True	True	True	True	True
10	True	True	True	False	True	True	False
11	True	True	True	True	True	False	True
12	True	True	True	True	True	False	True
13	True	True	True	True	True	False	False
14	True	True	True	True	True	True	True

In [181…

```
df.fillna(25, inplace = True)
df
```

Out[181…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	25.0	75	300.0	25.0
4	5	Manish	12	25.0	68	70.0	54.0
5	6	Sham	64	25.0	15	70.0	81.0
6	7	Virat	76	25.0	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	25.0	36	76.0	25.0
11	12	Abhishek	10	41.0	423	25.0	56.0
12	13	Omkar	48	23.0	59	25.0	45.0
13	14	Adesh	64	46.0	71	25.0	25.0
14	15	Anup	31	79.0	48	62.0	65.0

In [183…

```
df = pd.read_csv("/home/admin1/Downloads/Acadamics.csv")
df
```

Out[183…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	NaN	75	300.0	NaN
4	5	Manish	12	NaN	68	70.0	54.0
5	6	Sham	64	NaN	15	70.0	81.0
6	7	Virat	76	NaN	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	NaN	36	76.0	NaN
11	12	Abhishek	10	41.0	423	NaN	56.0
12	13	Omkar	48	23.0	59	NaN	45.0
13	14	Adesh	64	46.0	71	NaN	NaN
14	15	Anup	31	79.0	48	62.0	65.0

In [185…

```
df.fillna(method = 'ffill')
```

Out[185…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	39.0	75	300.0	78.0
4	5	Manish	12	39.0	68	70.0	54.0
5	6	Sham	64	39.0	15	70.0	81.0
6	7	Virat	76	39.0	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	67.0	36	76.0	45.0
11	12	Abhishek	10	41.0	423	76.0	56.0
12	13	Omkar	48	23.0	59	76.0	45.0
13	14	Adesh	64	46.0	71	76.0	45.0
14	15	Anup	31	79.0	48	62.0	65.0

In [187…

```
df.fillna(method = 'bfill')
```

Out[187…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	49.0	75	300.0	54.0
4	5	Manish	12	49.0	68	70.0	54.0
5	6	Sham	64	49.0	15	70.0	81.0
6	7	Virat	76	49.0	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	41.0	36	76.0	56.0
11	12	Abhishek	10	41.0	423	62.0	56.0
12	13	Omkar	48	23.0	59	62.0	45.0
13	14	Adesh	64	46.0	71	62.0	65.0
14	15	Anup	31	79.0	48	62.0	65.0

In [189…

```
df.fillna(df['DM'].mean())
```

Out[189…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	75.4	75	300.0	75.4
4	5	Manish	12	75.4	68	70.0	54.0
5	6	Sham	64	75.4	15	70.0	81.0
6	7	Virat	76	75.4	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	75.4	36	76.0	75.4
11	12	Abhishek	10	41.0	423	75.4	56.0
12	13	Omkar	48	23.0	59	75.4	45.0
13	14	Adesh	64	46.0	71	75.4	75.4
14	15	Anup	31	79.0	48	62.0	65.0

In [195…

```
df.fillna(method = 'pad')
```

Out[195…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	39.0	75	300.0	78.0
4	5	Manish	12	39.0	68	70.0	54.0
5	6	Sham	64	39.0	15	70.0	81.0
6	7	Virat	76	39.0	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	67.0	36	76.0	45.0
11	12	Abhishek	10	41.0	423	76.0	56.0
12	13	Omkar	48	23.0	59	76.0	45.0
13	14	Adesh	64	46.0	71	76.0	45.0
14	15	Anup	31	79.0	48	62.0	65.0

In [197…
```
df['NoT'].fillna(50)
```

Out[197…
```
0     56.0
1     42.0
2     39.0
3     50.0
4     50.0
5     50.0
6     50.0
7     49.0
8     51.0
9     67.0
10    50.0
11    41.0
12    23.0
13    46.0
14    79.0
Name: NoT, dtype: float64
```

In [199…
```
df
```

Out[199…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	NaN	75	300.0	NaN
4	5	Manish	12	NaN	68	70.0	54.0
5	6	Sham	64	NaN	15	70.0	81.0
6	7	Virat	76	NaN	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	NaN	36	76.0	NaN
11	12	Abhishek	10	41.0	423	NaN	56.0
12	13	Omkar	48	23.0	59	NaN	45.0
13	14	Adesh	64	46.0	71	NaN	NaN
14	15	Anup	31	79.0	48	62.0	65.0

In [201…

```
df.replace(to_replace = np.nan, value = 80)
```

Out[201…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	80.0	75	300.0	80.0
4	5	Manish	12	80.0	68	70.0	54.0
5	6	Sham	64	80.0	15	70.0	81.0
6	7	Virat	76	80.0	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	80.0	36	76.0	80.0
11	12	Abhishek	10	41.0	423	80.0	56.0
12	13	Omkar	48	23.0	59	80.0	45.0
13	14	Adesh	64	46.0	71	80.0	80.0
14	15	Anup	31	79.0	48	62.0	65.0

In [203…

```
df.interpolate(method = 'linear')
```

Out[203…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	41.0	75	300.0	66.0
4	5	Manish	12	43.0	68	70.0	54.0
5	6	Sham	64	45.0	15	70.0	81.0
6	7	Virat	76	47.0	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	54.0	36	76.0	50.5
11	12	Abhishek	10	41.0	423	72.5	56.0
12	13	Omkar	48	23.0	59	69.0	45.0
13	14	Adesh	64	46.0	71	65.5	55.0
14	15	Anup	31	79.0	48	62.0	65.0

In [205…

```
df.dropna(how = 'all')
```

Out[205…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
3	4	Sujal	65	NaN	75	300.0	NaN
4	5	Manish	12	NaN	68	70.0	54.0
5	6	Sham	64	NaN	15	70.0	81.0
6	7	Virat	76	NaN	3	72.0	6.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
10	11	Ram	75	NaN	36	76.0	NaN
11	12	Abhishek	10	41.0	423	NaN	56.0
12	13	Omkar	48	23.0	59	NaN	45.0
13	14	Adesh	64	46.0	71	NaN	NaN
14	15	Anup	31	79.0	48	62.0	65.0

In [207…

```
df.dropna(how = 'any')
```

Out[207…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
14	15	Anup	31	79.0	48	62.0	65.0

In [209…

```
df.dropna(axis = 0)
```

Out[209…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
14	15	Anup	31	79.0	48	62.0	65.0

In [211…

```
df.dropna(axis = 1)
```

Out[211…

	Roll No	Name	DS	DM
0	1	Rakesh	80	74
1	2	Yogesh	90	69
2	3	Yuvraj	545	24
3	4	Sujal	65	75
4	5	Manish	12	68
5	6	Sham	64	15
6	7	Virat	76	3
7	8	Roshan	84	89
8	9	Om	63	41
9	10	Vinod	45	36
10	11	Ram	75	36
11	12	Abhishek	10	423
12	13	Omkar	48	59
13	14	Adesh	64	71
14	15	Anup	31	48

In [213…

```
df.dropna(axis = 0, how = 'any')
```

Out[213…

	Roll No	Name	DS	NoT	DM	DT	Mdm
0	1	Rakesh	80	56.0	74	42.0	48.0
1	2	Yogesh	90	42.0	69	26.0	61.0
2	3	Yuvraj	545	39.0	24	34.0	78.0
7	8	Roshan	84	49.0	89	5.0	44.0
8	9	Om	63	51.0	41	78.0	747.0
9	10	Vinod	45	67.0	36	94.0	45.0
14	15	Anup	31	79.0	48	62.0	65.0

In [215…

```
import seaborn as sns
from scipy import stats
```

In [423…

```
df1 = pd.read_csv("/home/admin1/Downloads/Iris.csv")
df1
```

Out[423…

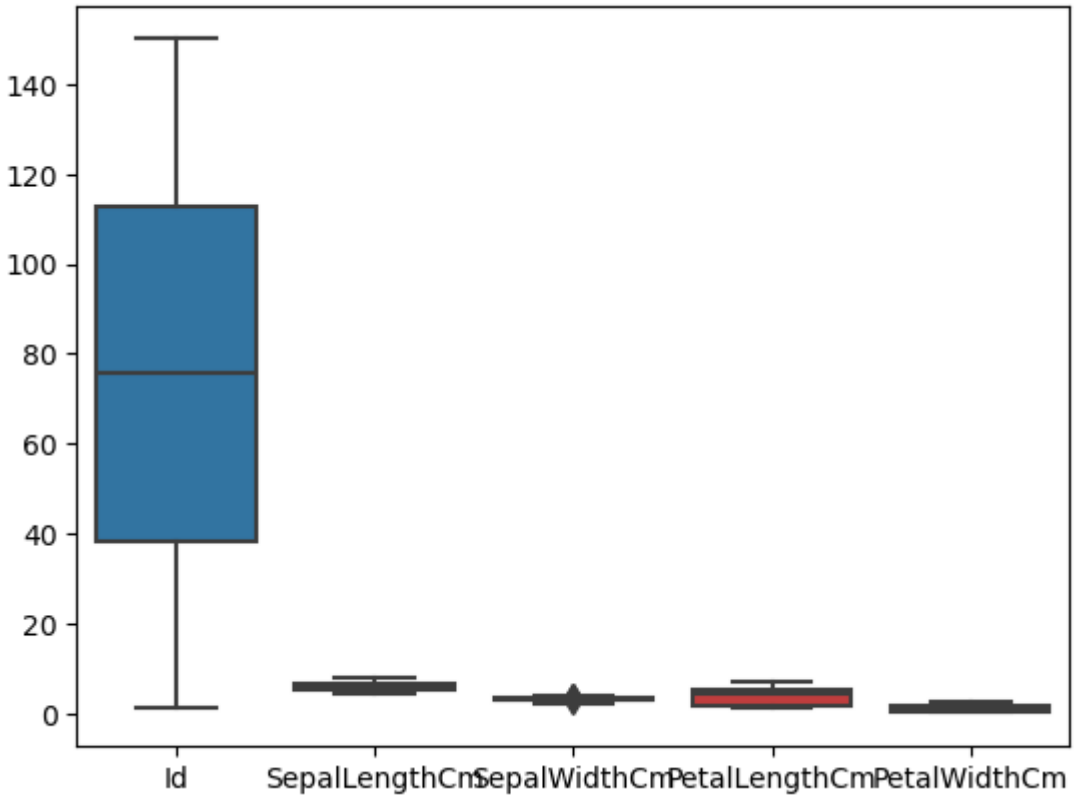
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
…	…	…	…	…	…	…
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [425…

```
sns.boxplot(df1)
```

Out[425… <Axes: >



In [427…
```
Z = np.abs(stats.zscore(df1['SepalLengthCm']))
Z
```

Out[427…
```
0      0.900681
1      1.143017
2      1.385353
3      1.506521
4      1.021849
         ...
145    1.038005
146    0.553333
147    0.795669
148    0.432165
149    0.068662
Name: SepalLengthCm, Length: 150, dtype: float64
```

In [429…
```
threshold = 3
```

In [431…
```
print(np.where(Z > threshold))
```

```
(array([], dtype=int64),)
```

In [433…
```
outlier_position = np.where(Z >= threshold) | (Z <= threshold)
```

In [435…
```
filtered = df1[outlier_position]
filtered
```

Out[435…

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [437…
```
df1.describe()
```

Out[437…

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [439…
```
Q1 = np.percentile(df1['SepalLengthCm'], 25, interpolation = 'midpoint')
Q1
```

Out[439… 5.1

In [441…
```
Q3 = np.percentile(df1['SepalLengthCm'], 75, interpolation = 'midpoint')
Q3
```

Out[441… 6.4

In [443…
```
IQR = Q3 - Q1
IQR
```

Out[443… 1.3000000000000007

In [445…
```
UB = Q3 + (1.5 * IQR)
LB = Q1 - (1.5 * IQR)
```

In [447…
```
UB
```

Out[447…
```
8.350000000000001
```

In [449…
```
LB
```

Out[449…
```
3.1499999999999986
```

In [451…
```
df1.shape
```

Out[451…
```
(150, 6)
```

In [453…
```
U = np.where(df1['SepalLengthCm'] > UB)
U
```

Out[453…
```
(array([], dtype=int64),)
```

In [455…
```
L = np.where(df1['SepalLengthCm'] < LB)
L
```

Out[455…
```
(array([], dtype=int64),)
```

In [457…
```
from sklearn.preprocessing import StandardScaler
```

In [459…
```
df1
```

Out[459…

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [461…
```
df1.drop(['Species'], axis = 1, inplace = True)
```

In [463…
```
df1
```

Out[463…

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2
...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5.0	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

150 rows × 5 columns

In [465…
```
std_scaler = StandardScaler()
```

In [467…
```
std_scaler
```

Out[467…

▾ StandardScaler

StandardScaler()

In [493…
```
df1_scaled_Data = std_scaler.fit_transform(df1.to_numpy())
df1_scaled_Data
```

Out[493…
```
array([[-1.72054204e+00, -9.00681170e-01,  1.03205722e+00,
        -1.34127240e+00, -1.31297673e+00],
       [-1.69744751e+00, -1.14301691e+00, -1.24957601e-01,
        -1.34127240e+00, -1.31297673e+00],
       [-1.67435299e+00, -1.38535265e+00,  3.37848329e-01,
        -1.39813811e+00, -1.31297673e+00],
       [-1.65125846e+00, -1.50652052e+00,  1.06445364e-01,
        -1.28440670e+00, -1.31297673e+00],
       [-1.62816394e+00, -1.02184904e+00,  1.26346019e+00,
        -1.34127240e+00, -1.31297673e+00],
       [-1.60506942e+00, -5.37177559e-01,  1.95766909e+00,
        -1.17067529e+00, -1.05003079e+00],
       [-1.58197489e+00, -1.50652052e+00,  8.00654259e-01,
        -1.34127240e+00, -1.18150376e+00],
       [-1.55888037e+00, -1.02184904e+00,  8.00654259e-01,
        -1.28440670e+00, -1.31297673e+00],
       [-1.53578584e+00, -1.74885626e+00, -3.56360566e-01,
        -1.34127240e+00, -1.31297673e+00],
       [-1.51269132e+00, -1.14301691e+00,  1.06445364e-01,
        -1.28440670e+00, -1.44444970e+00],
       [-1.48959680e+00, -5.37177559e-01,  1.49486315e+00,
        -1.28440670e+00, -1.31297673e+00],
       [-1.46650227e+00, -1.26418478e+00,  8.00654259e-01,
        -1.22754100e+00, -1.31297673e+00],
       [-1.44340775e+00, -1.26418478e+00, -1.24957601e-01,
        -1.34127240e+00, -1.44444970e+00],
       [-1.42031323e+00, -1.87002413e+00, -1.24957601e-01,
        -1.51186952e+00, -1.44444970e+00],
       [-1.39721870e+00, -5.25060772e-02,  2.18907205e+00,
        -1.45500381e+00, -1.31297673e+00],
       [-1.37412418e+00, -1.73673948e-01,  3.11468391e+00,
        -1.28440670e+00, -1.05003079e+00],
       [-1.35102965e+00, -5.37177559e-01,  1.95766909e+00,
        -1.39813811e+00, -1.05003079e+00],
       [-1.32793513e+00, -9.00681170e-01,  1.03205722e+00,
        -1.34127240e+00, -1.18150376e+00],
       [-1.30484061e+00, -1.73673948e-01,  1.72626612e+00,
        -1.17067529e+00, -1.18150376e+00],
       [-1.28174608e+00, -9.00681170e-01,  1.72626612e+00,
        -1.28440670e+00, -1.18150376e+00],
       [-1.25865156e+00, -5.37177559e-01,  8.00654259e-01,
        -1.17067529e+00, -1.31297673e+00],
       [-1.23555703e+00, -9.00681170e-01,  1.49486315e+00,
        -1.28440670e+00, -1.05003079e+00],
       [-1.21246251e+00, -1.50652052e+00,  1.26346019e+00,
        -1.56873522e+00, -1.31297673e+00],
       [-1.18936799e+00, -9.00681170e-01,  5.69251294e-01,
        -1.17067529e+00, -9.18557817e-01],
       [-1.16627346e+00, -1.26418478e+00,  8.00654259e-01,
        -1.05694388e+00, -1.31297673e+00],
       [-1.14317894e+00, -1.02184904e+00, -1.24957601e-01,
        -1.22754100e+00, -1.31297673e+00],
       [-1.12008441e+00, -1.02184904e+00,  8.00654259e-01,
        -1.22754100e+00, -1.05003079e+00],
       [-1.09698989e+00, -7.79513300e-01,  1.03205722e+00,
        -1.28440670e+00, -1.31297673e+00],
       [-1.07389537e+00, -7.79513300e-01,  8.00654259e-01,
        -1.34127240e+00, -1.31297673e+00],
       [-1.05080084e+00, -1.38535265e+00,  3.37848329e-01,
        -1.22754100e+00, -1.31297673e+00],
       [-1.02770632e+00, -1.26418478e+00,  1.06445364e-01,
        -1.22754100e+00, -1.31297673e+00],
       [-1.00461179e+00, -5.37177559e-01,  8.00654259e-01,
        -1.28440670e+00, -1.05003079e+00],
       [-9.81517269e-01, -7.79513300e-01,  2.42047502e+00,
```

```
 -1.28440670e+00, -1.44444970e+00],
[-9.58422745e-01, -4.16009689e-01,  2.65187798e+00,
 -1.34127240e+00, -1.31297673e+00],
[-9.35328221e-01, -1.14301691e+00,  1.06445364e-01,
 -1.28440670e+00, -1.44444970e+00],
[-9.12233697e-01, -1.02184904e+00,  3.37848329e-01,
 -1.45500381e+00, -1.31297673e+00],
[-8.89139173e-01, -4.16009689e-01,  1.03205722e+00,
 -1.39813811e+00, -1.31297673e+00],
[-8.66044649e-01, -1.14301691e+00,  1.06445364e-01,
 -1.28440670e+00, -1.44444970e+00],
[-8.42950125e-01, -1.74885626e+00, -1.24957601e-01,
 -1.39813811e+00, -1.31297673e+00],
[-8.19855601e-01, -9.00681170e-01,  8.00654259e-01,
 -1.28440670e+00, -1.31297673e+00],
[-7.96761077e-01, -1.02184904e+00,  1.03205722e+00,
 -1.39813811e+00, -1.18150376e+00],
[-7.73666553e-01, -1.62768839e+00, -1.74477836e+00,
 -1.39813811e+00, -1.18150376e+00],
[-7.50572030e-01, -1.74885626e+00,  3.37848329e-01,
 -1.39813811e+00, -1.31297673e+00],
[-7.27477506e-01, -1.02184904e+00,  1.03205722e+00,
 -1.22754100e+00, -7.87084847e-01],
[-7.04382982e-01, -9.00681170e-01,  1.72626612e+00,
 -1.05694388e+00, -1.05003079e+00],
[-6.81288458e-01, -1.26418478e+00, -1.24957601e-01,
 -1.34127240e+00, -1.18150376e+00],
[-6.58193934e-01, -9.00681170e-01,  1.72626612e+00,
 -1.22754100e+00, -1.31297673e+00],
[-6.35099410e-01, -1.50652052e+00,  3.37848329e-01,
 -1.34127240e+00, -1.31297673e+00],
[-6.12004886e-01, -6.58345429e-01,  1.49486315e+00,
 -1.28440670e+00, -1.31297673e+00],
[-5.88910362e-01, -1.02184904e+00,  5.69251294e-01,
 -1.34127240e+00, -1.31297673e+00],
[-5.65815838e-01,  1.40150837e+00,  3.37848329e-01,
  5.35295827e-01,  2.64698913e-01],
[-5.42721314e-01,  6.74501145e-01,  3.37848329e-01,
  4.21564419e-01,  3.96171883e-01],
[-5.19626790e-01,  1.28034050e+00,  1.06445364e-01,
  6.49027235e-01,  3.96171883e-01],
[-4.96532266e-01, -4.16009689e-01, -1.74477836e+00,
  1.37235899e-01,  1.33225943e-01],
[-4.73437742e-01,  7.95669016e-01, -5.87763531e-01,
  4.78430123e-01,  3.96171883e-01],
[-4.50343218e-01, -1.73673948e-01, -5.87763531e-01,
  4.21564419e-01,  1.33225943e-01],
[-4.27248694e-01,  5.53333275e-01,  5.69251294e-01,
  5.35295827e-01,  5.27644853e-01],
[-4.04154170e-01, -1.14301691e+00, -1.51337539e+00,
 -2.60824029e-01, -2.61192967e-01],
[-3.81059646e-01,  9.16836886e-01, -3.56360566e-01,
  4.78430123e-01,  1.33225943e-01],
[-3.57965122e-01, -7.79513300e-01, -8.19166497e-01,
  8.03701950e-02,  2.64698913e-01],
[-3.34870598e-01, -1.02184904e+00, -2.43898725e+00,
 -1.47092621e-01, -2.61192967e-01],
[-3.11776074e-01,  6.86617933e-02, -1.24957601e-01,
  2.50967307e-01,  3.96171883e-01],
[-2.88681550e-01,  1.89829664e-01, -1.97618132e+00,
  1.37235899e-01, -2.61192967e-01],
[-2.65587026e-01,  3.10997534e-01, -3.56360566e-01,
  5.35295827e-01,  2.64698913e-01],
[-2.42492502e-01, -2.94841818e-01, -3.56360566e-01,
 -9.02269170e-02,  1.33225943e-01],
[-2.19397978e-01,  1.03800476e+00,  1.06445364e-01,
```

```
  3.64698715e-01,  2.64698913e-01],
[-1.96303454e-01, -2.94841818e-01, -1.24957601e-01,
  4.21564419e-01,  3.96171883e-01],
[-1.73208930e-01, -5.25060772e-02, -8.19166497e-01,
  1.94101603e-01, -2.61192967e-01],
[-1.50114406e-01,  4.32165405e-01, -1.97618132e+00,
  4.21564419e-01,  3.96171883e-01],
[-1.27019882e-01, -2.94841818e-01, -1.28197243e+00,
  8.03701950e-02, -1.29719997e-01],
[-1.03925358e-01,  6.86617933e-02,  3.37848329e-01,
  5.92161531e-01,  7.90590793e-01],
[-8.08308339e-02,  3.10997534e-01, -5.87763531e-01,
  1.37235899e-01,  1.33225943e-01],
[-5.77363100e-02,  5.53333275e-01, -1.28197243e+00,
  6.49027235e-01,  3.96171883e-01],
[-3.46417860e-02,  3.10997534e-01, -5.87763531e-01,
  5.35295827e-01,  1.75297293e-03],
[-1.15472620e-02,  6.74501145e-01, -3.56360566e-01,
  3.07833011e-01,  1.33225943e-01],
[ 1.15472620e-02,  9.16836886e-01, -1.24957601e-01,
  3.64698715e-01,  2.64698913e-01],
[ 3.46417860e-02,  1.15917263e+00, -5.87763531e-01,
  5.92161531e-01,  2.64698913e-01],
[ 5.77363100e-02,  1.03800476e+00, -1.24957601e-01,
  7.05892939e-01,  6.59117823e-01],
[ 8.08308339e-02,  1.89829664e-01, -3.56360566e-01,
  4.21564419e-01,  3.96171883e-01],
[ 1.03925358e-01, -1.73673948e-01, -1.05056946e+00,
 -1.47092621e-01, -2.61192967e-01],
[ 1.27019882e-01, -4.16009689e-01, -1.51337539e+00,
  2.35044910e-02, -1.29719997e-01],
[ 1.50114406e-01, -4.16009689e-01, -1.51337539e+00,
 -3.33612130e-02, -2.61192967e-01],
[ 1.73208930e-01, -5.25060772e-02, -8.19166497e-01,
  8.03701950e-02,  1.75297293e-03],
[ 1.96303454e-01,  1.89829664e-01, -8.19166497e-01,
  7.62758643e-01,  5.27644853e-01],
[ 2.19397978e-01, -5.37177559e-01, -1.24957601e-01,
  4.21564419e-01,  3.96171883e-01],
[ 2.42492502e-01,  1.89829664e-01,  8.00654259e-01,
  4.21564419e-01,  5.27644853e-01],
[ 2.65587026e-01,  1.03800476e+00,  1.06445364e-01,
  5.35295827e-01,  3.96171883e-01],
[ 2.88681550e-01,  5.53333275e-01, -1.74477836e+00,
  3.64698715e-01,  1.33225943e-01],
[ 3.11776074e-01, -2.94841818e-01, -1.24957601e-01,
  1.94101603e-01,  1.33225943e-01],
[ 3.34870598e-01, -4.16009689e-01, -1.28197243e+00,
  1.37235899e-01,  1.33225943e-01],
[ 3.57965122e-01, -4.16009689e-01, -1.05056946e+00,
  3.64698715e-01,  1.75297293e-03],
[ 3.81059646e-01,  3.10997534e-01, -1.24957601e-01,
  4.78430123e-01,  2.64698913e-01],
[ 4.04154170e-01, -5.25060772e-02, -1.05056946e+00,
  1.37235899e-01,  1.75297293e-03],
[ 4.27248694e-01, -1.02184904e+00, -1.74477836e+00,
 -2.60824029e-01, -2.61192967e-01],
[ 4.50343218e-01, -2.94841818e-01, -8.19166497e-01,
  2.50967307e-01,  1.33225943e-01],
[ 4.73437742e-01, -1.73673948e-01, -1.24957601e-01,
  2.50967307e-01,  1.75297293e-03],
[ 4.96532266e-01, -1.73673948e-01, -3.56360566e-01,
  2.50967307e-01,  1.33225943e-01],
[ 5.19626790e-01,  4.32165405e-01, -3.56360566e-01,
  3.07833011e-01,  1.33225943e-01],
[ 5.42721314e-01, -9.00681170e-01, -1.28197243e+00,
```

```
 -4.31421141e-01, -1.29719997e-01],
[ 5.65815838e-01, -1.73673948e-01, -5.87763531e-01,
  1.94101603e-01,  1.33225943e-01],
[ 5.88910362e-01,  5.53333275e-01,  5.69251294e-01,
  1.27454998e+00,  1.71090158e+00],
[ 6.12004886e-01, -5.25060772e-02, -8.19166497e-01,
  7.62758643e-01,  9.22063763e-01],
[ 6.35099410e-01,  1.52267624e+00, -1.24957601e-01,
  1.21768427e+00,  1.18500970e+00],
[ 6.58193934e-01,  5.53333275e-01, -3.56360566e-01,
  1.04708716e+00,  7.90590793e-01],
[ 6.81288458e-01,  7.95669016e-01, -1.24957601e-01,
  1.16081857e+00,  1.31648267e+00],
[ 7.04382982e-01,  2.12851559e+00, -1.24957601e-01,
  1.61574420e+00,  1.18500970e+00],
[ 7.27477506e-01, -1.14301691e+00, -1.28197243e+00,
  4.21564419e-01,  6.59117823e-01],
[ 7.50572030e-01,  1.76501198e+00, -3.56360566e-01,
  1.44514709e+00,  7.90590793e-01],
[ 7.73666553e-01,  1.03800476e+00, -1.28197243e+00,
  1.16081857e+00,  7.90590793e-01],
[ 7.96761077e-01,  1.64384411e+00,  1.26346019e+00,
  1.33141568e+00,  1.71090158e+00],
[ 8.19855601e-01,  7.95669016e-01,  3.37848329e-01,
  7.62758643e-01,  1.05353673e+00],
[ 8.42950125e-01,  6.74501145e-01, -8.19166497e-01,
  8.76490051e-01,  9.22063763e-01],
[ 8.66044649e-01,  1.15917263e+00, -1.24957601e-01,
  9.90221459e-01,  1.18500970e+00],
[ 8.89139173e-01, -1.73673948e-01, -1.28197243e+00,
  7.05892939e-01,  1.05353673e+00],
[ 9.12233697e-01, -5.25060772e-02, -5.87763531e-01,
  7.62758643e-01,  1.57942861e+00],
[ 9.35328221e-01,  6.74501145e-01,  3.37848329e-01,
  8.76490051e-01,  1.44795564e+00],
[ 9.58422745e-01,  7.95669016e-01, -1.24957601e-01,
  9.90221459e-01,  7.90590793e-01],
[ 9.81517269e-01,  2.24968346e+00,  1.72626612e+00,
  1.67260991e+00,  1.31648267e+00],
[ 1.00461179e+00,  2.24968346e+00, -1.05056946e+00,
  1.78634131e+00,  1.44795564e+00],
[ 1.02770632e+00,  1.89829664e-01, -1.97618132e+00,
  7.05892939e-01,  3.96171883e-01],
[ 1.05080084e+00,  1.28034050e+00,  3.37848329e-01,
  1.10395287e+00,  1.44795564e+00],
[ 1.07389537e+00, -2.94841818e-01, -5.87763531e-01,
  6.49027235e-01,  1.05353673e+00],
[ 1.09698989e+00,  2.24968346e+00, -5.87763531e-01,
  1.67260991e+00,  1.05353673e+00],
[ 1.12008441e+00,  5.53333275e-01, -8.19166497e-01,
  6.49027235e-01,  7.90590793e-01],
[ 1.14317894e+00,  1.03800476e+00,  5.69251294e-01,
  1.10395287e+00,  1.18500970e+00],
[ 1.16627346e+00,  1.64384411e+00,  3.37848329e-01,
  1.27454998e+00,  7.90590793e-01],
[ 1.18936799e+00,  4.32165405e-01, -5.87763531e-01,
  5.92161531e-01,  7.90590793e-01],
[ 1.21246251e+00,  3.10997534e-01, -1.24957601e-01,
  6.49027235e-01,  7.90590793e-01],
[ 1.23555703e+00,  6.74501145e-01, -5.87763531e-01,
  1.04708716e+00,  1.18500970e+00],
[ 1.25865156e+00,  1.64384411e+00, -1.24957601e-01,
  1.16081857e+00,  5.27644853e-01],
[ 1.28174608e+00,  1.88617985e+00, -5.87763531e-01,
  1.33141568e+00,  9.22063763e-01],
[ 1.30484061e+00,  2.49201920e+00,  1.72626612e+00,
```

```
        1.50201279e+00,  1.05353673e+00],
       [ 1.32793513e+00,  6.74501145e-01, -5.87763531e-01,
        1.04708716e+00,  1.31648267e+00],
       [ 1.35102965e+00,  5.53333275e-01, -5.87763531e-01,
        7.62758643e-01,  3.96171883e-01],
       [ 1.37412418e+00,  3.10997534e-01, -1.05056946e+00,
        1.04708716e+00,  2.64698913e-01],
       [ 1.39721870e+00,  2.24968346e+00, -1.24957601e-01,
        1.33141568e+00,  1.44795564e+00],
       [ 1.42031323e+00,  5.53333275e-01,  8.00654259e-01,
        1.04708716e+00,  1.57942861e+00],
       [ 1.44340775e+00,  6.74501145e-01,  1.06445364e-01,
        9.90221459e-01,  7.90590793e-01],
       [ 1.46650227e+00,  1.89829664e-01, -1.24957601e-01,
        5.92161531e-01,  7.90590793e-01],
       [ 1.48959680e+00,  1.28034050e+00,  1.06445364e-01,
        9.33355755e-01,  1.18500970e+00],
       [ 1.51269132e+00,  1.03800476e+00,  1.06445364e-01,
        1.04708716e+00,  1.57942861e+00],
       [ 1.53578584e+00,  1.28034050e+00,  1.06445364e-01,
        7.62758643e-01,  1.44795564e+00],
       [ 1.55888037e+00, -5.25060772e-02, -8.19166497e-01,
        7.62758643e-01,  9.22063763e-01],
       [ 1.58197489e+00,  1.15917263e+00,  3.37848329e-01,
        1.21768427e+00,  1.44795564e+00],
       [ 1.60506942e+00,  1.03800476e+00,  5.69251294e-01,
        1.10395287e+00,  1.71090158e+00],
       [ 1.62816394e+00,  1.03800476e+00, -1.24957601e-01,
        8.19624347e-01,  1.44795564e+00],
       [ 1.65125846e+00,  5.53333275e-01, -1.28197243e+00,
        7.05892939e-01,  9.22063763e-01],
       [ 1.67435299e+00,  7.95669016e-01, -1.24957601e-01,
        8.19624347e-01,  1.05353673e+00],
       [ 1.69744751e+00,  4.32165405e-01,  8.00654259e-01,
        9.33355755e-01,  1.44795564e+00],
       [ 1.72054204e+00,  6.86617933e-02, -1.24957601e-01,
        7.62758643e-01,  7.90590793e-01]])
```

In [495…
```
df1_scaled_Data = pd.DataFrame(df1_scaled_Data, columns = ['Id', 'SepalLengthCm', 'Se
```

In [497…
```
df1_scaled_Data
```

Out[497…

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	-1.720542	-0.900681	1.032057	-1.341272	-1.312977
1	-1.697448	-1.143017	-0.124958	-1.341272	-1.312977
2	-1.674353	-1.385353	0.337848	-1.398138	-1.312977
3	-1.651258	-1.506521	0.106445	-1.284407	-1.312977
4	-1.628164	-1.021849	1.263460	-1.341272	-1.312977
...	...	...	...	...	...
145	1.628164	1.038005	-0.124958	0.819624	1.447956
146	1.651258	0.553333	-1.281972	0.705893	0.922064
147	1.674353	0.795669	-0.124958	0.819624	1.053537
148	1.697448	0.432165	0.800654	0.933356	1.447956
149	1.720542	0.068662	-0.124958	0.762759	0.790591

150 rows × 5 columns