Machine Learning Assignment No.3

Que.1 Using Make_blob generate data of 1000 data points with three cluster apply kmeans on it

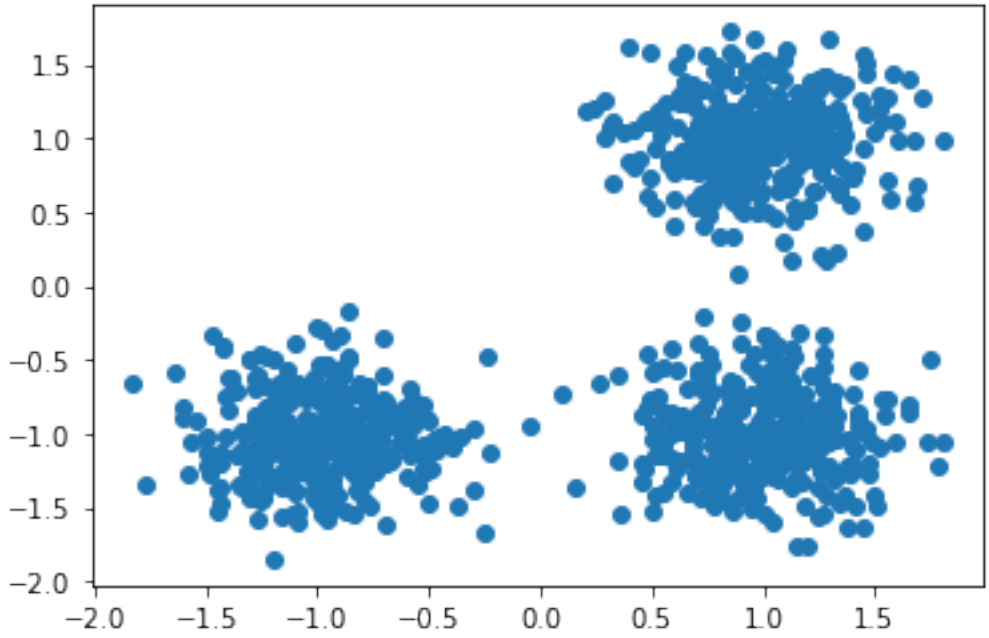
with k = 3 and use the metrics and get the accuracy (For Accuracy take reference of DBSCAN # evaluation) ● Apply DBscan on Cust Segmentation Data

```
import numpy as np
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
%matplotlib inline

#
#############################################################################
#######
# Generate sample data
centers = [[1, 1], [-1, -1], [1, -1]]
X, labels_true = make_blobs(
    n_samples=1000, centers=centers, cluster_std=0.3, random_state=0
)
#return independent variable , dependendent

X.shape

(1000, 2)

set(labels_true)

{0, 1, 2}

labels_true.shape  #dependent var

(1000,)

X.shape #independent

(1000, 2)

X

array([[ 0.68543411,  0.57399462],
       [ 1.5292157 ,  1.12004716],
       [ 0.85126135, -0.63502669],
       ...,
       [-0.8178665 , -1.31445112],
```

```
       [-0.51052077, -0.88667225],
       [ 0.70827913, -0.59613368]])
```

```
plt.scatter(X[:,0],X[:,1])
```

```
<matplotlib.collections.PathCollection at 0x1cd560b1fd0>
```




```
print(set(labels_true))
```

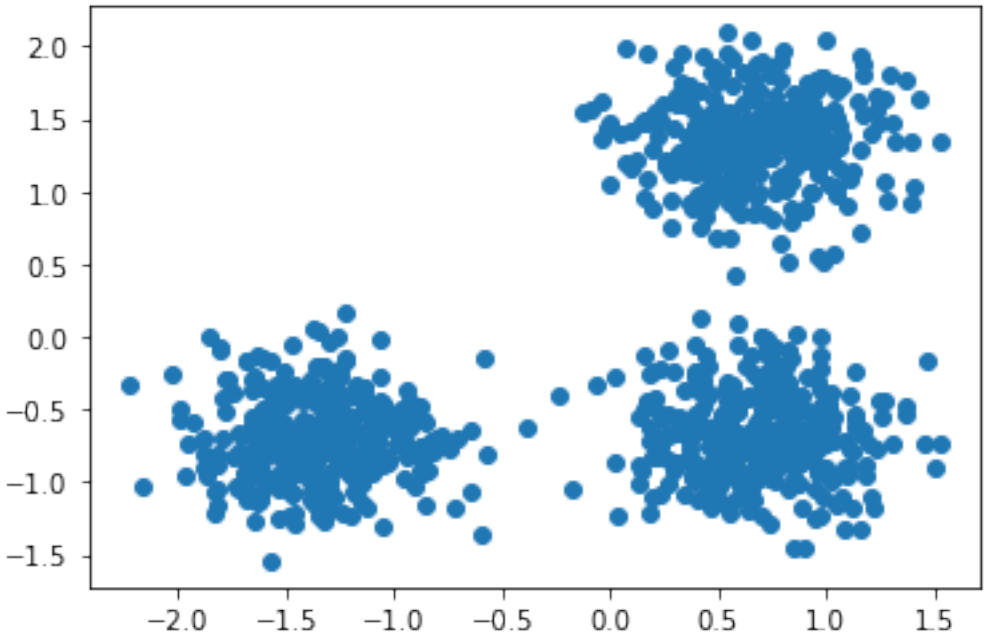
```
{0, 1, 2}
```

```
X = StandardScaler().fit_transform(X)
```

```
X
```

```
array([[ 0.36960469,  0.92478301],
       [ 1.24084714,  1.48033771],
       [ 0.54082878, -0.30527713],
       ...,
       [-1.1826208 , -0.9965246 ],
       [-0.86527251, -0.56130172],
       [ 0.39319321, -0.26570733]])
```

```
plt.scatter(X[:,0],X[:,1])
plt.show()
```



Modelling

```
from sklearn.cluster import KMeans
k_means = KMeans(init = "k-means++", n_clusters = 3, n_init = 100)
k_means.fit(X)#training
```

```
KMeans(n_clusters=3, n_init=100)
```

```
X.shape
```

```
(1000, 2)
```

```
labels = k_means.labels_
print(labels[0:5])
print(labels.shape)
print("number of classes =",set(labels))
```

```
[1 1 2 0 2]
(1000,)
number of classes = {0, 1, 2}
```

```
k_means.labels_
```

```
array([1, 1, 2, 0, 2, 0, 1, 1, 2, 0, 0, 1, 2, 1, 2, 1, 0, 2, 2, 2, 2,
2,
       1, 2, 1, 0, 0, 1, 1, 0, 0, 2, 2, 2, 0, 0, 2, 0, 1, 2, 2, 2, 1,
1,
       2, 2, 2, 1, 1, 0, 0, 1, 1, 1, 0, 2, 0, 2, 1, 0, 1, 0, 1, 1, 0,
0,
       1, 2, 1, 1, 1, 0, 1, 1, 0, 1, 1, 2, 1, 1, 1, 0, 2, 1, 2, 0, 0,
1,
```

0, 1, 2, 0, 0, 1, 1, 1, 2, 0, 0, 1, 2, 1, 1, 0, 2, 1, 0, 2, 1,
1,
2, 1, 2, 1, 2, 2, 0, 0, 1, 1, 2, 2, 1, 1, 0, 1, 1, 1, 0, 1, 1,
0,
2, 2, 1, 2, 0, 1, 2, 1, 0, 2, 2, 1, 2, 2, 2, 1, 2, 1, 1, 0, 1,
1,
0, 0, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 0, 0, 1, 0, 2, 2, 1, 2, 0,
1,
0, 1, 1, 1, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 1, 2, 0, 1, 0, 0, 0,
1,
2, 0, 2, 1, 2, 0, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 0, 2, 0,
1,
0, 1, 1, 1, 2, 2, 2, 0, 1, 2, 0, 0, 2, 2, 1, 0, 2, 2, 1, 1, 2,
2,
1, 2, 2, 0, 2, 2, 2, 1, 0, 1, 2, 2, 2, 1, 2, 0, 0, 1, 2, 2, 2,
0,
1, 2, 2, 2, 2, 2, 2, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 2, 0, 2, 0, 1,
2,
2, 0, 0, 1, 1, 0, 1, 2, 2, 0, 0, 2, 1, 1, 0, 1, 1, 1, 2, 1, 2,
1,
2, 0, 2, 0, 1, 0, 1, 0, 0, 1, 2, 0, 2, 2, 1, 0, 0, 0, 2, 0, 1,
0,
1, 0, 1, 2, 1, 2, 2, 0, 0, 1, 0, 1, 2, 1, 1, 2, 0, 2, 2, 1, 1,
1,
1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 0, 2, 2, 0, 1, 2, 0, 2, 1, 2, 2,
1,
2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 2, 0, 1, 2, 2, 2, 0, 2, 0, 0, 1,
2,
1, 0, 0, 1, 2, 0, 2, 1, 1, 0, 1, 0, 0, 0, 0, 0, 2, 1, 2, 0, 1,
1,
0, 0, 2, 2, 1, 2, 1, 2, 2, 0, 2, 2, 2, 0, 1, 0, 1, 1, 1, 0, 0,
1,
0, 0, 2, 0, 1, 0, 1, 1, 1, 2, 0, 2, 2, 2, 0, 1, 0, 1, 1, 1, 1,
1,
1, 0, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 0, 0, 2, 0, 0, 1, 0, 1,
2,
1, 0, 0, 0, 2, 1, 2, 1, 2, 2, 0, 0, 0, 0, 1, 2, 1, 0, 2, 0, 0,
1,
0, 2, 0, 2, 0, 1, 1, 2, 1, 1, 2, 0, 1, 0, 1, 0, 0, 0, 0, 0, 2,
2,
0, 1, 2, 0, 0, 1, 1, 1, 0, 1, 2, 2, 0, 1, 2, 1, 2, 2, 2, 1, 2,
0,
1, 1, 1, 2, 0, 0, 1, 1, 0, 2, 0, 2, 2, 2, 1, 0, 2, 0, 0, 0, 1,
0,
1, 1, 0, 2, 2, 0, 1, 1, 1, 2, 2, 1, 2, 1, 2, 1, 0, 0, 2, 0, 2,
2,
0, 0, 1, 0, 1, 2, 2, 1, 0, 1, 2, 2, 1, 0, 0, 0, 1, 1, 1, 0, 2,
2,
0, 2, 2, 2, 0, 1, 2, 2, 2, 0, 2, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1,
1,

```
       1, 1, 2, 2, 2, 1, 0, 1, 2, 2, 1, 0, 2, 1, 2, 2, 0, 1, 0, 1, 0,
0,
       2, 1, 2, 0, 0, 1, 0, 2, 0, 2, 1, 0, 2, 1, 1, 1, 0, 0, 1, 2, 2,
1,
       2, 2, 0, 0, 1, 1, 0, 2, 0, 1, 1, 2, 0, 2, 1, 0, 0, 2, 1, 1, 1,
1,
       0, 0, 2, 1, 0, 1, 2, 0, 0, 2, 0, 2, 0, 1, 2, 1, 0, 0, 1, 1, 2,
1,
       1, 1, 0, 0, 1, 1, 2, 0, 0, 2, 0, 2, 0, 1, 1, 0, 0, 0, 1, 0, 0,
0,
       1, 0, 1, 2, 2, 0, 0, 2, 2, 1, 0, 0, 0, 2, 1, 0, 1, 0, 1, 2, 2,
0,
       0, 1, 2, 2, 1, 0, 2, 2, 1, 0, 2, 1, 2, 2, 1, 2, 1, 2, 2, 0, 0,
0,
       0, 2, 2, 1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 1, 0, 2, 1, 2, 1, 2, 0,
2,
       0, 1, 0, 1, 1, 1, 0, 2, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0,
2,
       0, 2, 2, 0, 1, 0, 2, 1, 2, 1, 2, 1, 2, 0, 2, 1, 2, 1, 2, 2, 2,
0,
       2, 0, 0, 0, 2, 2, 0, 0, 1, 1, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 0,
2,
       2, 1, 1, 1, 1, 0, 2, 1, 1, 0, 0, 1, 0, 2, 2, 1, 2, 1, 0, 2, 0,
1,
       1, 2, 0, 1, 2, 0, 1, 0, 1, 1, 1, 1, 0, 1, 2, 0, 1, 0, 2, 0, 0,
0,
       0, 0, 1, 1, 2, 0, 0, 0, 1, 2, 0, 1, 2, 2, 2, 0, 1, 0, 2, 2, 2,
2,
       1, 2, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 2, 2, 2,
2,
       2, 2, 0, 2, 1, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 1, 1, 2, 1,
2,
       1, 1, 1, 2, 2, 0, 0, 0, 0, 2])
```

```
X
```

```
array([[ 0.36960469,  0.92478301],
       [ 1.24084714,  1.48033771],
       [ 0.54082878, -0.30527713],
       ...,
       [-1.1826208 , -0.9965246 ],
       [-0.86527251, -0.56130172],
       [ 0.39319321, -0.26570733]])
```

```
import pandas as pd
Data = pd.DataFrame({'X-axis':X[:,0],'Y-
axis':X[:,1],'True_labels':labels_true,'K_means.labels_':k_means.label
s_})
```

```
Data
```

```
       X-axis    Y-axis  True_labels  K_means.labels_
0    0.369605  0.924783            0                1
1    1.240847  1.480338            0                1
2    0.540829 -0.305277            2                2
3   -0.714977 -1.168275            1                0
4    0.468010 -1.177683            2                2
..        ...       ...          ...              ...
995 -1.347578 -1.005354            1                0
996 -0.643776 -1.067028            1                0
997 -1.182621 -0.996525            1                0
998 -0.865273 -0.561302            1                0
999  0.393193 -0.265707            2                2

[1000 rows x 4 columns]
```

```
X_axis = Data["X-axis"].values
```

```
X_axis
```

```
array([ 3.69604694e-01,  1.24084714e+00,  5.40828779e-01, -
7.14977248e-01,
        4.68010168e-01, -1.08526314e+00,  1.39425955e+00,
6.60547640e-04,
       -2.33213616e-01, -1.41040651e+00, -1.01315775e+00,
6.66411077e-01,
        1.14536428e+00,  9.55114763e-01,  5.71149482e-01,
1.07550726e+00,
       -1.60786841e+00,  3.48417928e-01,  4.34320644e-01,
4.55031606e-01,
        8.04592087e-01,  2.71530502e-01,  2.86788048e-01,
1.72606175e-01,
        6.52295869e-01, -1.27876472e+00, -1.25777884e+00,
8.98104676e-02,
        9.42393873e-01, -1.87634697e+00, -8.27051188e-01,
1.00458392e+00,
        7.19998005e-01,  7.01178735e-01, -1.53224817e+00, -
1.36120250e+00,
        6.37750837e-01, -1.46020509e+00,  1.02539333e+00,
2.68426176e-01,
        7.54108530e-01,  6.91292995e-01,  1.73628097e-01, -
3.96033929e-02,
        5.13477266e-01,  6.98666903e-01,  5.44715057e-01,
4.97918296e-01,
        6.75318514e-01, -1.31137734e+00, -1.14404060e+00,
8.56500129e-01,
       -1.45228139e-02,  3.26142927e-01, -1.13195545e+00,
3.93574906e-01,
       -1.47621588e+00,  1.70173545e-01,  8.36098779e-01, -
1.11275703e+00,
        5.55582855e-01, -1.47903158e+00,  2.93928251e-01,
9.25859791e-01,
```

```
       -9.29893490e-01, -3.88429607e-01,  1.02839429e+00,
9.68606908e-01,
        6.81344947e-01,  7.06374458e-01,  1.23304740e+00, -
1.34977519e+00,
        6.73269365e-01,  7.63746687e-01, -1.73546829e+00,
9.30149827e-01,
        6.45568638e-01,  9.56156912e-01,  6.82239740e-01,
6.85678298e-01,
        6.57805216e-01, -1.60544255e+00,  8.67658529e-01,
5.79577556e-01,
        1.19011267e+00, -1.52525896e+00, -9.60443836e-01,
7.51199914e-01,
       -1.41901233e+00,  8.26447516e-01,  3.52977449e-01, -
2.23100630e+00,
       -1.35489805e+00,  1.27290971e+00,  9.48513868e-01,
8.88082001e-01,
        5.99635185e-01, -1.59369312e+00, -1.40218393e+00,
1.06241638e+00,
        4.83895009e-01,  5.71980613e-01,  3.70626790e-01, -
1.37528420e+00,
        6.29889002e-01,  5.80049068e-01, -1.46375919e+00,
9.04290556e-01,
        1.14870067e+00,  2.17125458e-01,  4.36627271e-01,
7.86791422e-01,
        6.75733246e-01,  8.84627494e-01,  8.23827491e-01,
7.06460620e-01,
       -1.57751720e+00, -1.83606944e+00,  5.08966989e-01,
9.62178899e-01,
        8.22678507e-01,  6.79432724e-01,  4.68576703e-01,
6.54105739e-01,
       -1.36043183e+00,  1.06035608e+00,  8.92697500e-01,
1.15541432e+00,
       -5.88105178e-01,  9.33480660e-01,  3.38964226e-01, -
1.07921990e+00,
        5.83862352e-01,  1.27532088e-01,  9.83060730e-01,
1.50616623e+00,
       -1.39704753e+00,  5.75186409e-01,  1.01658887e+00,
1.43261937e+00,
       -1.21413140e+00,  8.35707681e-01,  7.08398856e-01,
4.80036345e-01,
        3.18705952e-01,  1.00135561e+00,  4.95934685e-01,
7.81118545e-01,
        5.81828367e-01,  4.27631847e-01,  5.18056033e-01, -
1.30080486e+00,
        2.81103559e-01,  5.59613439e-01, -1.34020794e+00, -
1.30647692e+00,
        5.55396486e-01,  9.48905383e-01,  5.26249456e-01,
9.09313718e-01,
        5.84753733e-01,  2.53286042e-01,  2.32778312e-01,
1.13785555e+00,
```

```
        8.47351301e-01,  6.73568596e-01, -1.27173351e+00, -
1.13386843e+00,
        7.45187210e-01, -1.40623503e+00,  3.97915488e-01,
8.81547630e-01,
        5.45828263e-01,  3.21534906e-01, -1.72567092e+00,
9.08621582e-01,
       -1.63346056e+00,  4.99849527e-01,  6.30316918e-01,
5.70673397e-01,
       -1.99151469e+00, -1.42441461e+00, -1.39333163e+00,
8.38875489e-01,
       -1.21355772e+00,  4.09533913e-01, -1.63715973e+00, -
1.30340751e+00,
        2.66801072e-01, -1.79189226e+00,  3.41338934e-01,
9.78529872e-01,
       -1.61239165e+00,  5.08346340e-01, -8.20957394e-01, -
1.47793727e+00,
       -1.93576994e+00,  1.05135837e+00,  5.49595805e-01, -
1.42093118e+00,
        8.69903179e-01,  1.05955100e+00,  5.07360674e-01, -
1.07313627e+00,
        7.03007818e-01,  6.97660646e-01,  4.01877241e-01,
1.10178976e+00,
        2.46324482e-01,  4.05479041e-01,  1.10302857e+00,
9.37983819e-01,
        6.97989634e-02,  7.09739861e-01,  6.44864616e-01,
7.70847229e-01,
       -1.22295612e+00,  1.04050175e+00, -1.49642776e+00,
9.00829274e-01,
       -7.46799520e-01,  1.65867909e-01,  1.15721928e+00,
2.58892140e-01,
        5.19237547e-01,  7.00178203e-01, -6.06603265e-02, -
1.68503785e+00,
        3.35901716e-01,  7.03870339e-01, -1.48269208e+00, -
1.72350498e+00,
        7.61178590e-01,  6.96767746e-01,  3.29148044e-01, -
1.81141175e+00,
        7.02929624e-01,  1.96791754e-01,  5.69518624e-01,
8.81852234e-01,
        6.62436581e-01,  8.23843946e-01,  7.69407428e-01,
9.64928248e-01,
        2.87735223e-01, -1.49498924e+00,  6.57110542e-01,
9.34482036e-01,
        2.44660795e-01,  1.40807741e+00, -1.66120288e+00,
5.86637641e-01,
        8.87531291e-01,  7.42878755e-01,  6.63738963e-01,
3.36265015e-01,
        3.83305612e-01, -1.30003923e+00, -1.30177086e+00,
8.56690436e-01,
        7.55688067e-01,  9.83782905e-01,  9.89981220e-01, -
1.63016390e+00,
```

```
        1.50497323e-01,  6.82244333e-01,  3.94531625e-01,
5.29615863e-01,
        6.43837700e-01,  4.89981966e-01, -1.13132918e+00, -
8.79762286e-01,
        9.82334873e-01, -1.66154404e+00, -1.29237253e+00,
8.77131357e-01,
        1.11709027e+00, -1.78837098e+00, -1.61303055e+00,
5.98106937e-01,
        7.99178027e-01, -1.64301370e+00,  8.53291037e-01, -
1.57115501e+00,
        1.20494016e+00,  7.25564510e-01,  7.35269569e-01, -
1.81114069e+00,
       -9.45999485e-01,  6.25453010e-01,  4.92351379e-01, -
1.77296756e+00,
       -3.79266158e-02,  6.37487306e-01,  9.63594563e-01, -
1.58904073e+00,
       -1.63676012e+00,  5.69515105e-01,  3.79055916e-01,
9.08088607e-01,
       -9.78083967e-01,  6.62434691e-01,  7.33720856e-01,
6.89135820e-01,
        9.50078308e-01,  7.30076730e-01,  5.00797456e-01,
1.16920717e+00,
        3.73450989e-02, -1.35359176e+00,  1.08773489e+00, -
7.80562661e-01,
        4.85524483e-01, -1.61831370e+00,  4.84380489e-01, -
1.35836355e+00,
       -1.37413433e+00,  9.76348321e-01,  5.30645253e-01, -
1.23978186e+00,
        1.78908087e-01,  1.08401136e+00,  6.31558089e-01, -
1.26829485e+00,
       -1.64834444e+00, -1.32091486e+00,  9.65986071e-01, -
1.28112285e+00,
        8.11011010e-01, -1.18229717e+00,  7.24679750e-01, -
9.60240847e-01,
        7.46037487e-01,  8.18587871e-01,  1.03523604e+00,
4.58256351e-01,
        5.93251454e-01, -1.81873027e+00, -9.26681551e-01,
6.19439215e-01,
       -1.05416841e+00,  6.13359916e-01,  5.42709190e-01,
5.72860337e-01,
        1.27773946e+00,  1.94546851e-02, -1.05724998e+00,
7.86567209e-01,
        1.05012521e+00,  8.70133941e-01,  1.27290941e+00,
7.83106944e-01,
        1.04400536e+00,  6.85159787e-01,  5.91011869e-01,
1.08488633e+00,
        4.43896956e-01,  3.72068471e-01,  8.20375643e-01,
8.55814999e-01,
        7.29330222e-01,  2.98007135e-01, -1.42326675e+00,
8.19520365e-01,
```

```
        5.89496746e-01, -1.13775623e+00,  3.96878931e-01,
9.99102707e-01,
       -1.23184729e+00,  1.12895821e+00,  6.53246571e-01,
5.30583550e-01,
        1.11030341e+00,  5.75259013e-01,  6.10397990e-01,
8.62489135e-01,
        1.17803554e+00,  2.03596058e-01,  7.79799556e-01,
1.36136603e+00,
       -5.97140313e-01,  4.41865487e-01, -1.31093962e+00, -
1.38596062e+00,
        7.64831599e-01, -1.30268578e+00,  8.38809463e-01,
6.99833821e-01,
        1.37712135e-01,  8.36065957e-01, -1.69380526e+00,
8.38847756e-01,
       -1.21507888e+00, -1.20683441e+00,  3.74037448e-01,
5.09122950e-01,
        4.19404509e-01, -1.18578199e+00, -1.02458143e+00,
7.79443269e-01,
        6.75282302e-01, -8.60035468e-01,  5.45646671e-01,
6.04007562e-01,
        4.92315762e-01, -1.07329409e+00,  5.22703237e-01, -
9.92341357e-01,
       -1.68054765e+00, -1.30567812e+00, -1.04697174e+00, -
1.45643149e+00,
        2.01435671e-01,  8.93683459e-01,  4.34476468e-01, -
1.32135185e+00,
        9.10764649e-01,  5.81439377e-01, -9.39330450e-01, -
5.67023806e-01,
        2.91608147e-01,  4.54559775e-01,  5.57826389e-01,
1.45370960e+00,
        4.76117704e-01,  2.71655847e-01,  7.86938710e-01, -
1.38848379e+00,
        1.16863508e+00,  9.07488665e-01,  2.69089667e-01, -
1.16082878e+00,
        4.51978182e-01, -1.65538962e+00,  8.04797423e-01,
4.63710268e-01,
       -9.64149191e-02, -1.19846489e+00, -1.05148967e+00,
7.42888135e-01,
       -1.25682984e+00, -1.21182382e+00,  7.56047804e-01, -
8.40272182e-01,
        8.34973193e-01, -1.51133935e+00,  9.84576740e-01,
3.93595380e-01,
        5.58363243e-01,  5.82548457e-01, -1.63478180e+00,
1.34311145e-01,
        2.14815456e-01,  1.83540913e-01, -1.28668320e+00,
3.45358625e-01,
       -1.49091783e+00,  9.87832201e-01,  7.56144048e-01,
1.16318192e+00,
        6.52992261e-01,  6.11390463e-01,  3.17135236e-01, -
1.29660326e+00,
```

```
        6.09015455e-01,  1.11198039e+00,  5.53300784e-01,
6.53812375e-01,
        9.65709138e-01,  7.70810965e-01,  6.32345929e-01,
9.60527238e-01,
        7.07133897e-01,  9.57871478e-01,  7.02690698e-01, -
1.15257394e+00,
       -1.46732064e+00,  7.83004362e-01, -1.62270417e+00, -
1.87832450e+00,
        9.86350339e-01, -1.86088084e+00,  6.11474048e-01,
9.77284741e-01,
        5.98687187e-01, -1.15681212e+00, -1.06546644e+00, -
1.74084511e+00,
        8.32938101e-01,  7.91385068e-01,  5.49611963e-01,
1.01313478e+00,
        6.97563225e-01,  1.06917552e+00, -1.69019744e+00, -
1.53972348e+00,
       -1.49506970e+00, -1.44543947e+00,  3.89132875e-01,
5.84147345e-01,
        5.86780595e-01, -1.53562777e+00,  6.89520298e-01, -
1.56445656e+00,
       -1.38782803e+00,  8.17076546e-01, -1.60675708e+00,
4.29485852e-01,
       -1.29637976e+00,  9.35180150e-01, -1.20446382e+00,
4.44854442e-01,
       -1.29306357e-01,  8.13457483e-01,  3.84577924e-01,
5.36536483e-01,
        5.32519707e-01, -1.68256155e+00,  6.35350631e-01, -
1.29646784e+00,
        9.02660495e-01, -1.43478574e+00, -9.00225631e-01, -
1.36486082e+00,
       -1.24029671e+00, -8.69467686e-01,  7.57936472e-01,
5.04639035e-01,
       -7.04822112e-01,  4.55916231e-01,  7.53171067e-01, -
1.10319496e+00,
       -1.40300363e+00,  9.71727072e-01, -4.28333059e-03,
5.50803015e-01,
       -1.15403929e+00,  5.36179340e-01,  5.45644505e-01,
7.15657659e-01,
       -1.33818542e+00,  8.48884164e-01,  3.87534436e-01,
4.17025274e-01,
        5.87788627e-01,  1.12514478e+00,  8.36770748e-01,
7.64293448e-01,
        4.95244895e-01, -1.49496969e+00,  1.54689782e-01,
8.12964657e-01,
        8.37868991e-01,  5.79744779e-01, -1.26283171e+00, -
1.33995178e+00,
        6.26402327e-01,  4.12906446e-01, -1.22559318e+00,
1.12854891e+00,
       -1.30409334e+00,  7.00269500e-01,  8.31585682e-01,
4.35695918e-01,
```

```
        9.31892388e-01, -1.50000107e+00,  8.48895528e-01, -
1.54089642e+00,
       -1.97338961e+00, -1.36415758e+00,  5.34099617e-01, -
1.52722339e+00,
        8.04299105e-01,  4.79256357e-01, -1.08247524e+00,
8.27845294e-01,
        5.88568722e-01, -1.95686646e+00,  6.87152500e-01,
1.89432189e-01,
        7.49368265e-01,  6.85570921e-01,  4.64914665e-01,
8.31900748e-01,
        5.60870216e-01,  6.37085252e-01,  5.49032790e-01,
1.36449965e+00,
       -1.18398500e+00, -1.46102187e+00,  7.40077918e-01, -
1.47133482e+00,
        7.06218595e-01,  9.67350369e-01, -9.41578373e-01, -
1.52789225e+00,
        4.65318506e-01, -1.99388890e+00,  9.63581288e-01,
1.02245610e+00,
        9.69010839e-01,  8.62706312e-01, -1.86687216e+00,
7.08582441e-01,
        8.47321040e-01,  5.66701384e-01,  1.39749524e+00, -
1.03571101e+00,
       -1.34356777e+00, -1.20316991e+00,  1.03504252e+00,
3.75371773e-01,
        8.34572725e-01, -1.77886255e+00,  7.61081762e-01,
7.38334643e-01,
       -1.07309409e+00,  4.16917104e-01,  3.04171996e-01,
8.61162981e-01,
       -1.61498277e+00,  5.40135852e-01,  8.66056617e-01,
1.09084716e+00,
        3.86701034e-01, -1.13087513e+00,  1.03629222e+00,
9.97585420e-01,
        3.52310483e-01, -1.62391135e+00, -1.72616980e+00, -
1.28531080e+00,
       -2.02584897e+00, -1.38990362e+00, -1.14669169e+00,
4.01853152e-01,
        9.86529373e-01,  4.95038639e-01,  5.09691804e-01,
2.32470963e-01,
        3.91801596e-01,  6.48325800e-01,  9.45062638e-01,
4.82526518e-01,
       -1.71285172e+00,  6.77020163e-01,  6.50208679e-01,
3.82774632e-01,
        1.04339880e+00, -1.30121668e+00,  6.75710026e-01,
2.89693802e-01,
        1.23359158e+00,  3.85001231e-01, -1.44996817e+00,
4.45541139e-01,
       -1.48391621e+00,  4.42525091e-01, -1.38977505e+00, -
1.08452338e+00,
        1.02095395e+00,  4.96953404e-01,  1.20880372e+00, -
1.61846587e+00,
```

```
       -1.16296358e+00,  5.49796416e-01, -1.49653970e+00,
8.93602945e-01,
       -1.82347325e+00,  8.14935230e-01,  7.39027563e-01, -
9.09333724e-01,
        7.06215688e-01,  3.67722727e-01,  4.06279596e-01,
1.07382831e+00,
       -1.62276565e+00, -1.46385369e+00,  7.81278725e-01,
3.40688751e-01,
        1.86884298e-01,  6.80250705e-01,  6.94863785e-01,
3.66841393e-01,
       -1.57448331e+00, -1.79529323e+00,  2.81737588e-01,
7.60536449e-01,
       -9.36596359e-01,  7.29274337e-01, -1.32992502e+00,
7.58924108e-01,
        5.84927870e-01,  6.80433669e-01, -1.35670096e+00,
3.89874001e-01,
        6.63911125e-01, -1.53569371e+00, -1.25735523e+00,
9.47751860e-01,
        9.61608876e-01,  1.03401635e+00,  7.54833594e-01,
1.23114189e+00,
       -1.64866387e+00, -9.69181918e-01,  8.80070873e-01,
9.03950962e-01,
       -1.28734799e+00,  3.14812937e-01,  5.61066738e-01, -
9.59792437e-01,
       -9.78806899e-01,  9.66811023e-01, -1.26828122e+00,
3.80997449e-01,
       -1.35470579e+00,  9.88710830e-01,  8.32879887e-01,
7.05871145e-01,
       -8.77299596e-01, -1.69454717e+00,  8.19038875e-01,
9.88504028e-01,
        1.37621935e-01,  6.97080731e-01,  9.20253762e-01,
1.23492462e-01,
       -9.33552241e-01, -1.23599420e+00,  8.03376759e-01,
8.18018113e-01,
        2.11797057e-01, -1.08070319e+00, -1.22164993e+00,
8.15328931e-01,
       -1.36747599e+00,  1.18623868e+00, -1.20680293e+00,
2.14141714e-01,
        5.09381411e-01, -1.66717117e+00, -1.21640338e+00, -
1.20859444e+00,
        5.53836198e-01, -1.10088966e+00, -1.49052029e+00, -
1.38176147e+00,
        1.31742202e+00, -1.08060513e+00,  7.42405179e-01,
1.01074627e+00,
        3.60067014e-01, -1.55387678e+00, -1.57751485e+00,
7.27005197e-01,
        9.94489770e-01,  5.66284063e-01, -1.89363386e+00, -
1.25932049e+00,
       -1.30024385e+00,  7.42318881e-01,  9.46193276e-02, -
1.30935879e+00,
```

```
        5.38653866e-01, -1.53648898e+00,  6.53547642e-01,
3.82999206e-01,
        7.21228431e-01, -1.27517206e+00, -1.81827155e+00,
3.00858253e-01,
        9.81422059e-01,  6.44420493e-01,  6.86872778e-01, -
1.65962582e+00,
        7.06820797e-01,  8.43226093e-01,  4.89946772e-01, -
1.66417920e+00,
        7.99436302e-01,  7.83280659e-01,  1.46523649e+00,
7.82625854e-01,
        4.97446221e-01,  1.13488424e+00,  6.53762098e-02,
5.70394676e-01,
        1.60614921e-01, -1.57403031e+00, -9.59383223e-01, -
1.61555700e+00,
       -1.29794269e+00,  5.48589283e-01,  5.52647774e-01,
5.74428415e-01,
        7.92652906e-01,  1.90632959e-01,  9.79656569e-01,
5.78201242e-01,
        4.90412456e-01, -1.24596314e+00,  6.60182539e-01, -
1.51187988e+00,
        2.35064995e-01,  6.24940892e-01, -1.55320346e+00,
7.02271657e-01,
        1.88677550e-01,  9.00700432e-01,  8.73612963e-01,
1.00683474e+00,
       -8.53608868e-01,  1.25450135e+00, -1.26844900e+00,
4.10043759e-01,
       -1.50149954e+00,  1.09275902e+00,  7.99388249e-01,
4.55765600e-02,
       -1.18112912e+00,  6.83336329e-01,  9.73212069e-01, -
1.28261157e+00,
       -1.60469660e+00,  3.62728051e-01, -1.38100293e+00, -
1.63600953e+00,
       -1.69055696e+00, -1.19597257e+00, -6.51782764e-01, -
1.54614422e+00,
       -1.81877885e+00,  1.23998407e+00, -1.20520581e+00,
9.47787724e-01,
       -1.42278845e+00,  1.52475111e+00,  1.30363980e+00, -
1.47846771e+00,
        1.46579476e-01, -1.11894508e+00,  4.92044372e-01,
5.48809096e-01,
        9.51930330e-01,  4.57704232e-01,  6.20282775e-01,
4.71500954e-01,
        4.70161439e-01, -1.26139212e+00,  9.33185272e-01,
1.68396934e-01,
        2.59172673e-02,  6.67789909e-01,  4.49973076e-01,
7.54258628e-01,
        6.61889244e-01, -1.38882980e+00,  2.26770071e-01, -
1.24478915e+00,
       -1.14444551e+00, -1.41515752e+00, -1.73612003e-01,
5.36141845e-01,
```

```
       -1.31370842e+00, -1.58637840e+00,  1.16524704e+00,
3.31017320e-01,
       -1.87961250e+00, -8.06319044e-01,  7.60259829e-01,
4.69644545e-01,
        1.52959990e+00, -1.04880769e+00,  1.81642972e-01, -
1.43959708e+00,
        9.33933666e-01,  1.21660503e+00, -1.13658108e+00,
3.50732481e-01,
        1.15438386e+00,  8.09276171e-01,  9.46716628e-01,
8.73014908e-01,
        9.60161634e-01, -1.30665665e+00,  1.58431767e-01,
4.42457569e-01,
        3.06337857e-01, -1.43213856e+00, -9.35604904e-01,
9.59822966e-01,
       -1.27004908e+00,  1.11427479e+00,  3.06400435e-01,
1.15734488e+00,
        7.45786587e-01,  6.89550377e-01, -1.14765715e+00,
7.69124306e-01,
       -1.45631063e+00,  8.38653347e-01,  4.81341142e-01,
8.87241826e-01,
       -8.72420387e-01,  1.04735286e+00,  5.91534962e-01, -
1.50222702e+00,
        7.15012730e-01, -1.23406593e+00,  5.49171115e-01,
4.11649122e-01,
        5.98655753e-01,  4.81507973e-01, -7.97745984e-01,
4.24666957e-01,
        6.98509320e-01, -1.63451644e+00,  9.33361524e-01, -
1.13712280e+00,
        7.15322381e-01, -1.01592142e+00, -1.75064444e+00, -
1.61946571e+00,
       -1.35212864e+00, -1.61173467e+00,  4.13360213e-01,
1.94481489e-01,
        5.61611874e-01, -9.65304025e-01, -1.29293178e+00, -
1.13668782e+00,
        4.86104426e-01,  3.43262450e-01, -1.52435622e+00,
1.01483054e+00,
        7.41126097e-01,  1.03462246e+00,  9.10974980e-01, -
1.14380361e+00,
        1.29006536e+00, -1.11526410e+00,  1.92789519e-01,
4.25266226e-01,
        9.17335049e-01,  8.37145383e-01,  6.60166663e-01,
9.32545203e-01,
       -1.05606580e+00,  3.98291261e-01,  6.46383690e-01, -
1.23576553e+00,
       -1.60415196e+00, -1.07121239e+00, -1.36689281e+00, -
1.82951775e+00,
       -1.60290853e+00,  7.82467587e-01, -1.32063667e+00, -
1.06942575e+00,
       -8.81607031e-01, -1.48376261e+00,  7.04093668e-01, -
1.22495881e+00,
```

```
        4.08551150e-01,  6.06994143e-01,  7.24971698e-01,
7.21102563e-01,
        8.70495943e-01,  8.61718518e-01, -1.08061974e+00,
4.47596221e-01,
        6.27205067e-01, -1.61610331e+00, -1.60501591e+00,
1.36695576e+00,
        9.42854471e-01,  7.72571189e-01,  7.93259353e-01, -
1.51047903e+00,
        4.93974563e-01, -2.17073793e+00,  4.08213910e-01, -
1.39381124e+00,
        4.35490428e-01,  2.04071596e-01,  3.33487239e-01,
1.97318711e-01,
        2.76870036e-01,  7.15358738e-01,  5.97900523e-01,
1.30027835e+00,
        7.38380031e-01,  1.27263841e+00,  4.28723163e-01, -
1.34757768e+00,
       -6.43776248e-01, -1.18262080e+00, -8.65272510e-01,
3.93193208e-01])
```

```
Y_axis = Data["Y-axis"].values
```

```
Y_axis
```

```
array([ 9.24783008e-01,  1.48033771e+00, -3.05277133e-01, -
1.16827460e+00,
       -1.17768297e+00, -5.35951538e-01,  1.34530379e+00,
1.48074727e+00,
       -4.07721376e-01, -7.59551789e-01, -6.23949668e-01,
1.77562020e+00,
       -6.59664183e-01,  1.28206168e+00, -4.37742458e-01,
1.72519247e+00,
       -5.06963992e-01, -6.87374833e-01, -1.09421795e+00, -
4.07379823e-01,
       -8.22279082e-01, -8.23071971e-02,  1.21731647e+00, -
4.30421780e-01,
        1.70520405e+00, -4.23272786e-01, -2.27523978e-01,
1.15678908e+00,
        1.38208815e+00, -6.97092264e-01, -7.17192334e-01, -
4.94320069e-01,
       -9.27350692e-01, -6.63617750e-01, -1.03819214e+00, -
6.97857581e-01,
       -4.03048187e-01, -1.28662721e+00,  1.21981891e+00, -
5.23688465e-01,
       -4.64034212e-01, -2.98736496e-01,  1.08686929e+00,
1.62192798e+00,
       -5.74320454e-01, -1.23964209e+00, -6.75816769e-01,
1.24748551e+00,
        1.32544977e+00, -4.07851510e-01, -6.96589938e-01,
1.30584209e+00,
        1.43495973e+00,  1.20351082e+00, -7.16414422e-01, -
2.41260454e-01,
```

```
       -1.06214777e+00, -4.53990117e-01,  1.06378221e+00, -
5.76774246e-01,
        1.16001032e+00, -5.89965010e-02,  1.43971093e+00,
9.95311210e-01,
       -5.55707815e-01, -6.18762379e-01,  5.77086011e-01, -
2.26778367e-01,
        1.27063770e+00,  8.52538189e-01,  1.66171106e+00, -
6.71614828e-01,
        1.88114880e+00,  1.45880922e+00, -1.02461279e+00,
1.39533924e+00,
        2.04699973e+00, -8.82806607e-01,  1.00167550e+00,
1.48893719e+00,
        1.15059723e+00, -6.91888237e-01, -1.00126321e+00,
1.06958025e+00,
       -7.34237785e-01, -6.70084716e-01, -7.29160752e-01,
1.86820755e+00,
       -3.56396838e-01,  1.56480772e+00, -6.84094533e-01, -
3.25068769e-01,
       -1.22423703e+00,  1.63474491e+00,  1.77328566e+00,
8.69220760e-01,
       -8.10432923e-01, -9.49338739e-01, -6.70717083e-01,
1.45493891e+00,
       -8.48842920e-01,  1.00432267e+00,  1.72786787e+00, -
4.25891646e-01,
       -1.96071228e-01,  1.28513812e+00, -7.84963580e-01, -
2.77622288e-01,
        1.61841784e+00,  1.37751039e+00, -6.24000114e-01,
1.76304166e+00,
       -5.91725520e-01,  1.63967789e+00, -1.05514095e+00, -
4.85361315e-01,
       -5.76805871e-01, -1.21918188e+00,  1.01760695e+00,
1.13167759e+00,
       -8.31419883e-01, -8.62206493e-01,  1.41819488e+00,
1.38687799e+00,
       -9.37309041e-01,  1.16533158e+00,  8.64673467e-01,
1.93686571e+00,
       -1.36385321e-01,  1.60955186e+00,  1.22456990e+00, -
5.52924969e-01,
       -5.10055765e-01, -8.88984663e-01,  1.46196581e+00, -
9.04745655e-01,
       -8.48837769e-01,  1.38699958e+00, -6.26348886e-01,
1.64647601e+00,
       -5.66951352e-01, -7.85741984e-01, -1.09090997e-01,
1.82713526e+00,
       -5.89989271e-01, -7.33588988e-01, -1.02281666e+00,
1.32824403e+00,
       -4.38819172e-01,  1.94119245e+00,  1.44058187e+00, -
2.52605970e-01,
        9.47155799e-01,  1.92263467e+00, -5.60185500e-01, -
5.82675267e-01,
```

```
       -1.20896334e+00,  1.35981696e+00,  1.48518853e+00,
1.30949646e+00,
        9.38538269e-01,  1.20763022e+00, -2.19949114e-01, -
2.30303860e-01,
        1.32276424e+00,  9.55800142e-01, -7.18654582e-01, -
7.58514397e-01,
        1.38760079e+00, -5.82364349e-01, -3.67116496e-01, -
1.17398381e+00,
        1.54754759e+00, -5.49311335e-01, -6.62236011e-01,
1.57025319e+00,
       -7.76002532e-01,  1.21138222e+00,  1.62685015e+00,
1.47115017e+00,
       -5.61571120e-01, -8.32272998e-01, -1.07258837e+00, -
7.05412071e-01,
       -7.12077229e-01,  1.30076518e-01, -5.76397273e-01, -
6.52729893e-01,
       -8.40173637e-01, -6.89237921e-01,  1.59778664e+00, -
6.94012197e-01,
       -5.93852373e-01,  1.83197792e+00, -7.96599946e-01, -
9.19140812e-01,
       -5.83935422e-01,  1.68772323e+00, -9.39852150e-01, -
4.41755711e-01,
       -6.31946435e-01,  1.30328480e+00, -1.02746897e+00, -
6.07670411e-01,
       -8.06485737e-01,  1.90328576e+00,  1.43087549e+00, -
7.75327386e-01,
        1.60246885e+00,  9.72687234e-01,  1.08886926e+00,
1.21584089e+00,
        1.19354895e+00,  1.50893063e+00, -9.30531997e-01, -
2.10848073e-01,
       -6.31291696e-01, -1.15393724e+00, -4.55509456e-01,
1.21758043e+00,
       -7.70232738e-01,  1.95361811e+00,  1.29558304e+00,
1.17810387e+00,
       -5.31307027e-01, -3.54799989e-01, -3.36303756e-01, -
8.60445553e-01,
        1.26288355e+00, -6.87859154e-01, -3.51668378e-01, -
1.04726170e+00,
       -5.03234077e-01, -4.05912296e-01,  1.75542402e+00, -
8.98686945e-02,
        4.65552944e-03, -1.09568558e+00,  1.73131679e+00,
1.63154355e+00,
       -9.97507410e-01, -6.88816080e-01,  1.08694593e+00, -
3.10417515e-01,
       -2.44753007e-01, -7.73251222e-01, -1.12811905e+00, -
5.44490647e-01,
       -5.59945434e-01,  1.03466302e+00, -5.44408136e-01,
1.40592244e+00,
       -4.04750323e-01, -7.42634155e-01, -9.97282596e-01,
1.59663919e+00,
```

```
       -9.81270087e-01, -6.15105743e-01, -7.03575564e-01,
1.38518977e+00,
       -6.46766001e-01, -1.10131743e+00, -5.10517433e-01, -
8.57042727e-01,
        1.49583596e+00, -9.66235258e-01, -5.30990602e-01, -
9.68471184e-01,
       -7.02989467e-01, -6.11584348e-01, -1.18375535e+00, -
4.78828517e-01,
        1.53590840e+00, -3.66185928e-01, -8.01135984e-01,
1.24714439e+00,
        1.14776754e+00, -2.90502749e-01, -4.85802540e-01,
1.38793006e+00,
       -4.77664568e-01, -7.47138486e-01, -1.45223963e+00, -
5.32462052e-01,
        1.40832721e+00, -5.12038492e-01, -1.28111069e+00, -
6.67844666e-02,
       -8.32216136e-01,  1.07951108e+00,  1.19912390e+00, -
2.87117238e-01,
        1.36440795e+00, -4.24911191e-01, -6.20385655e-01, -
4.69059309e-01,
       -9.50977087e-01, -3.86823553e-01,  1.33443868e+00,
1.75555165e+00,
       -5.24144464e-01,  1.48352460e+00,  1.48089692e+00,
1.47392641e+00,
       -7.67655882e-01,  1.24247795e+00, -7.95757385e-01,
1.80667990e+00,
       -1.23785516e+00, -5.85347509e-01, -8.93263568e-01, -
6.81111856e-01,
        1.36791688e+00, -6.03108271e-01,  1.75433063e+00, -
1.15488040e+00,
       -6.73091847e-01,  1.45502303e+00, -7.09566527e-01, -
5.09499272e-01,
       -7.15233096e-01, -1.31348524e+00,  1.32466307e+00, -
7.20942283e-01,
       -1.27566752e+00, -9.67368280e-01, -1.23421228e+00, -
5.50584923e-01,
        1.02264192e+00, -3.36292230e-01,  1.53613093e+00, -
6.13929846e-01,
        1.55202613e+00, -6.79549132e-01,  1.75438441e+00, -
1.11823734e+00,
       -6.79107370e-01, -8.14668107e-01, -5.48897911e-01,
1.82160592e+00,
       -8.45248266e-01,  1.30257670e+00, -7.61848605e-01,
4.28456347e-01,
        9.46837895e-01, -8.68783741e-01, -8.77421669e-01, -
6.63165047e-01,
       -9.54611720e-01,  1.29023654e+00,  1.05991633e+00,
1.54403314e+00,
        1.50959118e+00, -7.12669688e-01, -6.82745360e-01, -
7.07243698e-01,
```

```
       -1.01360278e+00, -7.72930077e-01, -3.09818770e-01,
1.18245927e+00,
       -4.75989997e-01,  1.17951596e+00, -7.17541616e-01, -
8.54772719e-02,
       -5.55503806e-02, -5.04765564e-01,  1.11612589e+00, -
7.02732982e-01,
       -1.58892221e-01, -5.74094865e-01,  1.26730967e+00, -
6.23581134e-01,
       -9.02445110e-01,  1.22273368e+00, -1.07863024e+00, -
4.58082346e-01,
       -9.63591263e-01, -4.18856812e-01, -8.92815160e-01, -
5.33665305e-01,
       -1.36191966e+00, -1.26834178e-01, -1.02974690e+00, -
6.03643600e-01,
       -1.83899552e-01, -5.56630695e-01,  1.24519597e+00, -
5.77824340e-01,
       -9.21465630e-01, -3.82875153e-01, -9.38199348e-01, -
1.53199849e-01,
       -9.39783079e-01, -6.01472204e-01,  1.56623813e+00, -
1.14054864e+00,
        7.53621925e-01, -9.77941648e-01, -8.34980507e-01,
1.08627603e+00,
       -8.12985114e-01, -1.14969759e+00, -8.36308441e-01,
1.12577853e+00,
        1.23887457e+00, -8.03725589e-01,  1.21455330e+00, -
7.35460818e-01,
       -1.65666854e-01, -1.05526556e+00, -4.93762425e-01, -
3.33562648e-01,
       -6.75618060e-01,  8.78815136e-01, -3.74396239e-01, -
1.27439076e+00,
        1.35935249e+00,  1.40603076e+00, -3.59909169e-01, -
7.99920470e-01,
       -9.21953138e-01, -7.58824819e-01,  1.27263131e+00, -
7.41065934e-01,
        1.64610401e+00, -6.83276690e-01, -3.98653853e-01, -
7.96008996e-01,
       -6.48384605e-01, -6.61384088e-01, -1.00249826e+00, -
6.60784903e-01,
        1.32450847e+00, -6.17769127e-01,  1.57386213e+00,
1.10595559e+00,
        1.55769931e+00, -4.04200253e-01, -1.30148661e+00,
1.42906794e+00,
       -6.18514564e-01, -7.76200618e-01, -6.95998694e-02, -
9.18371259e-01,
        7.99727324e-01, -6.50367245e-01,  1.78927286e+00,
1.45429019e+00,
        1.41354974e+00, -4.90733778e-01, -6.69593759e-01, -
5.51053918e-01,
       -5.14232001e-01, -2.49731777e-01, -9.21234796e-01,
1.13518374e+00,
```

```
       -6.94892799e-01,  1.31088929e+00,  1.34275630e+00,
1.52702876e+00,
        1.68715100e+00,  1.60312765e+00,  1.26512040e+00, -
2.49935002e-01,
        1.14157804e+00, -3.99553299e-01,  1.36353654e+00,
1.70404070e+00,
        1.32306838e+00, -4.91029791e-01, -1.19976349e+00, -
9.23716144e-01,
       -4.95463603e-01, -9.32719420e-01, -6.20608775e-01, -
1.13311354e+00,
       -9.33075370e-01, -3.77922783e-01, -8.35817209e-01, -
9.00423621e-01,
        1.27652615e+00,  8.19166266e-03,  1.01390940e+00, -
1.66621396e-03,
        8.46650882e-01, -4.33440916e-01, -4.42521970e-01, -
8.29336816e-01,
       -8.64654321e-01,  1.09751404e+00, -9.39137960e-01,
1.33613807e+00,
       -5.63652009e-03, -1.10577350e+00, -1.11306414e+00, -
1.25181571e+00,
       -8.19590021e-01, -9.44643209e-01,  9.08967377e-01,
9.87291250e-02,
        1.18078651e+00, -4.79717477e-01, -8.26687637e-01, -
3.73781084e-01,
       -7.57160626e-01,  1.02457680e+00, -9.47570210e-01, -
2.92872382e-01,
       -6.28084106e-01, -7.65433356e-01, -5.53080212e-01,
1.01686986e+00,
        1.54326267e+00, -3.68416826e-01,  8.86705790e-01,
1.22449230e+00,
       -3.58263442e-01, -1.63349922e-01,  1.23768539e+00, -
7.89472429e-01,
        1.48256723e+00, -8.83570156e-01, -1.03406368e+00, -
8.57834574e-01,
       -6.01333513e-01, -9.28228080e-01, -3.22077661e-01, -
9.27745588e-01,
       -7.04145816e-01,  1.52279143e+00, -1.56942296e-01, -
9.22228855e-01,
       -9.78159956e-01,  1.77786871e+00,  1.04611153e+00,
1.50515954e+00,
       -1.00050830e+00,  9.97848438e-01, -3.90882366e-01, -
7.59705784e-01,
       -6.35728582e-01,  1.67806409e+00, -9.82636516e-01,
1.47629210e+00,
       -7.53734014e-01, -1.17757473e+00, -5.91122156e-01,
1.49546288e+00,
       -6.30783307e-01, -9.20803090e-01,  9.60425183e-01,
1.34775009e+00,
        1.44338728e+00, -9.69385921e-01, -4.74161605e-01, -
1.95482844e-01,
```

```
        1.02733138e+00,  1.22869112e+00, -1.46775755e-01, -
6.28244294e-01,
       -6.46928247e-01, -7.01640048e-01, -6.09942290e-01, -
9.65566093e-01,
        1.46694940e+00, -1.01157998e+00, -9.01833426e-01, -
1.01190781e+00,
       -9.55201747e-01, -6.46246011e-01,  1.05944264e+00, -
9.25597818e-01,
        1.54648794e+00,  1.31253002e+00, -7.30111700e-01, -
7.22351527e-01,
       -6.69187462e-01, -7.30873756e-01,  1.68759385e+00,
1.49945215e+00,
        1.23556978e+00, -7.75715868e-01, -8.62264452e-01,
1.46004577e+00,
       -5.01786109e-01,  1.11169069e+00, -4.09321306e-01,
1.76613722e+00,
       -8.58100824e-01, -1.20799733e+00, -9.74943828e-01, -
8.93706802e-01,
       -5.20109666e-01, -3.61312106e-01, -7.69113335e-01, -
9.97656878e-01,
        1.82916117e+00, -4.96611939e-01,  1.44270851e+00, -
9.88709123e-01,
       -7.59661727e-01,  1.49220643e+00, -9.47387324e-01,
1.30106920e+00,
       -5.27954503e-01, -7.99745545e-01,  9.14299375e-01, -
8.69310418e-01,
       -3.90076790e-01, -1.23151604e+00,  1.55820144e+00,
1.25139166e+00,
        1.38808145e+00, -5.11474767e-01, -9.83764896e-01, -
1.16999556e-01,
       -2.70944369e-01, -5.09240580e-01, -6.67865983e-01, -
7.20078802e-01,
       -3.42495515e-01,  1.94713421e+00, -8.81137683e-01, -
9.47016646e-01,
       -5.33193393e-02, -9.12982271e-01, -5.54351385e-01,
2.04216800e+00,
        1.37412339e+00, -1.10432142e+00, -7.85905290e-01, -
2.38595108e-01,
       -2.47655256e-01, -2.92277972e-01, -6.69089422e-01,
1.23304598e+00,
        5.21996007e-01,  6.79573400e-01,  1.34215693e+00,
1.49231288e+00,
       -8.56553403e-01, -6.47299098e-01, -1.04228401e+00,
1.86473040e+00,
       -6.86043925e-01,  1.35777880e+00, -1.05130130e+00, -
9.27355364e-01,
        1.02858434e+00, -1.18897159e+00, -8.93625686e-01,
1.86429695e+00,
       -7.66179190e-01, -9.02506440e-01, -7.82832423e-01,
1.14773684e+00,
```

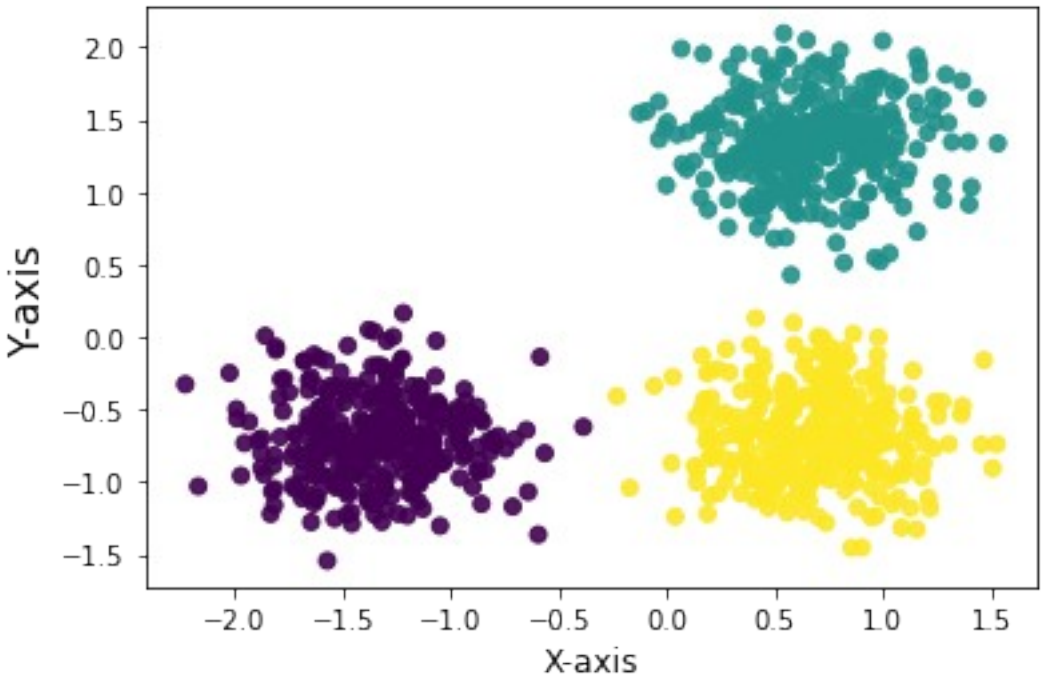
```
       -3.90276507e-01,  8.31304765e-01, -1.11390174e+00, -
7.17064017e-01,
       -3.86387645e-01,  1.23694605e+00, -1.10897792e+00, -
7.43773177e-01,
       -8.92546763e-01,  9.25732800e-01, -9.40325758e-01, -
4.97063261e-01,
       -8.85201348e-01, -3.19175850e-01,  1.80207557e+00, -
5.17488975e-01,
       -8.53299209e-01,  1.60855796e+00,  1.30369249e+00,
1.37799695e+00,
       -1.11890189e+00, -8.78232807e-01,  1.05507989e+00, -
1.07673653e+00,
       -1.21879170e+00,  1.42550499e+00, -9.10615557e-01, -
8.46299555e-01,
       -3.80878739e-01, -4.12386560e-01,  7.57337001e-01,
9.89320130e-01,
       -8.01858467e-01, -8.76687303e-01, -1.10560723e+00,
1.65629176e+00,
        8.64820540e-01, -6.52451977e-01,  3.74998201e-02, -
9.50787794e-01,
        1.15569439e+00, -3.43682407e-01, -2.80478717e-01, -
6.03897973e-01,
        5.47652251e-01,  1.28671408e+00,  8.14432100e-01,
1.56712528e+00,
       -2.76008733e-01, -4.88445549e-01, -7.54834707e-01,
1.24160459e+00,
       -5.16385873e-01,  1.61591840e+00, -3.25787439e-01, -
9.71019165e-01,
       -6.12048330e-01, -4.82656978e-01, -3.86872098e-01, -
6.50667532e-01,
       -4.11890972e-01,  1.31200422e+00, -6.44333640e-01,
1.59258107e+00,
       -7.55523204e-01, -1.00772181e+00,  1.14918349e+00,
1.38492397e+00,
       -1.00118304e+00,  1.51905401e+00,  1.39756976e+00,
1.21231409e+00,
       -4.36758138e-01, -5.11753665e-01,  1.47395198e+00,
5.11949495e-01,
       -8.31358933e-01, -4.69546058e-01,  1.65610099e-01, -
1.91749473e-01,
       -4.56832509e-01, -8.72936929e-01, -5.47606643e-01,
1.48557578e+00,
        1.28558296e+00, -1.08816824e+00, -6.84595953e-01, -
8.40682991e-01,
        1.48128613e+00, -9.44576144e-01, -8.32353332e-01,
5.00506520e-02,
        1.34459010e+00, -6.87313312e-01,  1.47362448e+00, -
4.60293247e-01,
       -7.11465522e-01, -3.33405802e-01, -1.63244614e-01, -
7.04577873e-01,
```

```
       -8.99521477e-01,  1.13006304e+00, -7.10928465e-01, -
7.20726297e-01,
       -9.86930987e-01, -3.58693560e-01,  1.41582063e+00, -
3.46372778e-01,
        2.09453177e+00, -6.68847357e-01,  1.48174313e+00, -
6.45650010e-01,
       -2.43415864e-01, -1.09722082e+00, -1.04608404e+00,
1.65408125e+00,
       -5.06263536e-01, -1.28899051e-01,  1.13293940e+00, -
2.97115028e-01,
       -7.76639054e-01, -9.72111745e-01,  1.41186508e+00, -
6.01992401e-01,
       -8.56569311e-01,  6.49908681e-01, -1.58956940e-01, -
7.71029918e-01,
        1.56466282e+00, -5.37310109e-01,  1.98832701e+00, -
1.12598463e+00,
       -1.27955742e-01, -1.54176558e+00, -7.07514728e-01, -
9.53528109e-01,
       -2.58246952e-02, -5.93338506e-01, -5.63117196e-01,
1.26593255e+00,
        1.51692858e+00, -7.17752117e-01,  1.45548374e+00,
9.83941370e-01,
        1.20124681e+00, -1.21863890e+00,  8.51321974e-01, -
2.41308712e-01,
       -1.07626093e+00,  1.26594874e+00, -8.13482243e-01, -
4.90297489e-01,
        8.79686749e-01, -1.45019943e+00,  1.46492323e+00, -
5.15568029e-01,
       -5.84428260e-01, -5.43330549e-01,  2.80221316e-03,
1.43526868e+00,
       -6.44640682e-01,  8.98234791e-01,  1.97506539e+00,
1.39596400e+00,
       -1.08605380e+00, -5.74042666e-01, -1.26520339e-01, -
5.49812775e-01,
       -1.04323283e+00,  1.68004178e+00, -1.14383410e+00, -
1.14712557e+00,
       -8.09906076e-01, -1.06867319e+00, -6.40863016e-01, -
7.70515220e-01,
       -1.15966959e+00, -4.46436119e-01, -5.79576510e-01, -
7.61731890e-01,
       -5.61255650e-01, -7.37892414e-01, -7.36496168e-01, -
1.08968352e+00,
        1.46670186e+00, -4.97357966e-01, -6.52637459e-01,
1.29205066e+00,
       -5.27662298e-01,  9.19358161e-01, -7.51927990e-01,
1.11005944e+00,
       -1.01833499e+00, -7.23231426e-01, -6.02381631e-01,
1.47640747e+00,
       -2.74152794e-01,  1.41750372e+00, -6.76220492e-01, -
2.80626193e-01,
```

```
       -2.59238512e-01, -7.73522196e-01, -7.61267075e-01, -
7.35563028e-01,
       -3.39461593e-01, -7.93203530e-01, -1.03933300e+00, -
4.40138610e-01,
       -7.94260909e-01, -7.65236684e-01,  1.88305538e+00,
1.95143450e+00,
       -8.00542968e-01, -6.99691618e-01, -2.04415877e-01,
9.35970553e-01,
        1.33563822e+00, -5.30371880e-01, -6.29706113e-01, -
9.38761478e-01,
        9.96856943e-01, -1.17961897e+00, -8.41155859e-01, -
8.43715274e-01,
       -1.32485624e+00,  1.40156171e+00,  1.37618684e+00,
1.29462450e+00,
        1.70648918e+00, -5.14516506e-01, -4.96905316e-01,
9.10618828e-01,
        1.59550751e+00, -6.47852129e-01, -6.31904826e-01,
1.15949521e+00,
       -3.72260734e-01, -7.98124502e-01, -8.55558416e-01,
7.26399752e-01,
       -2.70920429e-01,  1.40732011e+00, -1.01323109e+00, -
3.32378904e-01,
       -7.55829753e-01,  8.89308531e-01,  9.87640273e-01, -
8.69254243e-01,
       -8.48006872e-01,  9.81307431e-01, -8.96797140e-01, -
9.29923725e-01,
        1.45052215e+00, -6.22030305e-01,  6.86276612e-01,
1.69913777e+00,
        1.28694274e+00,  1.32234590e+00, -7.41969592e-01,
1.18152478e+00,
       -7.13821633e-01, -1.20269005e-01,  1.67240734e+00, -
3.51445458e-01,
       -4.39260851e-01, -5.70844352e-01, -3.81317184e-01, -
8.18326911e-01,
       -7.41467928e-01, -1.46502290e-01,  1.37405634e+00,
1.29326885e+00,
       -8.71283536e-01, -6.84475794e-01, -3.54667844e-01, -
4.11851054e-01,
        1.24845856e+00, -3.72370107e-01, -6.99433558e-01,
1.05413706e+00,
       -7.54576458e-01, -7.18672884e-01, -7.67789090e-01, -
7.91127101e-01,
        1.81008495e+00, -5.14475039e-01, -8.32581553e-01, -
4.76257657e-01,
       -7.52424202e-01, -7.98990727e-01,  1.66957913e+00, -
8.05199347e-01,
       -4.62629383e-01,  1.25260082e+00, -6.56438602e-01, -
8.59497057e-01,
       -6.34575794e-01, -5.21528824e-01, -1.97631400e-01, -
1.06501233e+00,
```

```
       -6.66587645e-01,  1.89009971e+00, -1.99418426e-01, -
2.53012920e-02,
       -9.19404592e-01, -1.22335137e+00, -7.04716048e-01, -
5.81541527e-01,
       -1.11957975e+00, -1.03010886e+00, -6.16417853e-01, -
7.89464040e-03,
       -7.78182507e-01,  2.24958734e-02, -8.37037811e-01, -
2.04099880e-01,
        1.49414467e+00, -8.38858547e-01, -5.84573286e-01, -
4.74522925e-01,
       -1.24020089e+00, -7.95881091e-01, -5.82818048e-01, -
5.95508139e-01,
       -6.24334512e-01, -1.02881081e+00, -2.45583951e-01, -
5.97635994e-01,
       -9.64147787e-01,  1.54450182e+00,  1.63315222e+00, -
2.31371955e-01,
        1.12575376e+00, -8.17925260e-01,  1.37534435e+00,
1.47726600e+00,
        1.26073597e+00, -4.45406712e-01, -6.35258011e-01, -
1.00535422e+00,
       -1.06702769e+00, -9.96524603e-01, -5.61301717e-01, -
2.65707327e-01])
```

```
import matplotlib.pyplot as plt
plt.scatter(X_axis,Y_axis, c=Data['K_means.labels_'], alpha=0.9)
plt.xlabel('X-axis', fontsize=12)
plt.ylabel('Y-axis', fontsize=15)

plt.show()
```



Apply DBSCAN on cust_segmentation Data

```
import numpy as np
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

import pandas as pd
cust_df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/cust_segmentation.csv")
cust_df.head()
```

```
   Customer Id  Age  Edu  Years Employed  Income  Card Debt  Other
Debt  \
0            1   41    2               6      19      0.124
1.073
1            2   47    1              26     100      4.582
8.218
2            3   33    2              10      57      6.111
5.802
3            4   29    2               4      19      0.681
0.516
4            5   47    1              31     253      9.308
8.908

   Defaulted Address  DebtIncomeRatio
0        0.0  NBA001              6.3
1        0.0  NBA021             12.8
2        1.0  NBA013             20.9
3        0.0  NBA009              6.3
4        0.0  NBA008              7.2
```


```
   Customer Id  Age  Edu  Years Employed  Income  Card Debt  Other
Debt  \
0            1   41    2               6      19      0.124
1.073
1            2   47    1              26     100      4.582
8.218
2            3   33    2              10      57      6.111
5.802
3            4   29    2               4      19      0.681
0.516
4            5   47    1              31     253      9.308
8.908

   Defaulted Address  DebtIncomeRatio
0        0.0  NBA001              6.3
1        0.0  NBA021             12.8
2        1.0  NBA013             20.9
```

```
3         0.0  NBA009              6.3
4         0.0  NBA008              7.2
```

Data Pre-Processing

```
df = cust_df.drop(["Customer Id","Age","Years
Employed","Address"],axis = 1)
df.head(2)
```

```
   Edu  Income  Card Debt  Other Debt  Defaulted  DebtIncomeRatio
0    2      19      0.124       1.073        0.0              6.3
1    1     100      4.582       8.218        0.0             12.8
```

```
   Edu  Income  Card Debt  Other Debt  Defaulted  DebtIncomeRatio
0    2      19      0.124       1.073        0.0              6.3
1    1     100      4.582       8.218        0.0             12.8
```

```
df.head()
```

```
   Edu  Income  Card Debt  Other Debt  Defaulted  DebtIncomeRatio
0    2      19      0.124       1.073        0.0              6.3
1    1     100      4.582       8.218        0.0             12.8
2    2      57      6.111       5.802        1.0             20.9
3    2      19      0.681       0.516        0.0              6.3
4    1     253      9.308       8.908        0.0              7.2
```

```
   Edu  Income  Card Debt  Other Debt  Defaulted  DebtIncomeRatio
0    2      19      0.124       1.073        0.0              6.3
1    1     100      4.582       8.218        0.0             12.8
2    2      57      6.111       5.802        1.0             20.9
3    2      19      0.681       0.516        0.0              6.3
4    1     253      9.308       8.908        0.0              7.2
```

```
df.tail()
```

```
     Edu  Income  Card Debt  Other Debt  Defaulted  DebtIncomeRatio
845    1      26      0.548       1.220        NaN              6.8
846    2      34      0.359       2.021        0.0              7.0
847    4      18      2.802       3.210        1.0             33.4
848    1      28      0.116       0.696        0.0              2.9
849    1      64      1.866       3.638        0.0              8.6
```

```
     Edu  Income  Card Debt  Other Debt  Defaulted  DebtIncomeRatio
845    1      26      0.548       1.220        NaN              6.8
846    2      34      0.359       2.021        0.0              7.0
847    4      18      2.802       3.210        1.0             33.4
848    1      28      0.116       0.696        0.0              2.9
849    1      64      1.866       3.638        0.0              8.6
```

```
df.describe()
```

```
              Edu      Income   Card Debt  Other Debt   Defaulted  \
count  850.000000  850.000000  850.000000  850.000000  700.000000
mean     1.710588   46.675294    1.576820    3.078773    0.261429
std      0.927784   38.543054    2.125843    3.398799    0.439727
min      1.000000   13.000000    0.012000    0.046000    0.000000
25%      1.000000   24.000000    0.382500    1.045750    0.000000
50%      1.000000   35.000000    0.885000    2.003000    0.000000
75%      2.000000   55.750000    1.898500    3.903250    1.000000
max      5.000000  446.000000   20.561000   35.197000    1.000000

       DebtIncomeRatio
count       850.000000
mean         10.171647
std           6.719441
min           0.100000
25%           5.100000
50%           8.700000
75%          13.800000
max          41.300000
```


```
              Edu      Income   Card Debt  Other Debt   Defaulted  \
count  850.000000  850.000000  850.000000  850.000000  700.000000
mean     1.710588   46.675294    1.576820    3.078773    0.261429
std      0.927784   38.543054    2.125843    3.398799    0.439727
min      1.000000   13.000000    0.012000    0.046000    0.000000
25%      1.000000   24.000000    0.382500    1.045750    0.000000
50%      1.000000   35.000000    0.885000    2.003000    0.000000
75%      2.000000   55.750000    1.898500    3.903250    1.000000
max      5.000000  446.000000   20.561000   35.197000    1.000000

       DebtIncomeRatio
count       850.000000
mean         10.171647
std           6.719441
min           0.100000
25%           5.100000
50%           8.700000
75%          13.800000
max          41.300000
```


```
df.isnull().sum()

Edu                 0
Income              0
Card Debt           0
Other Debt          0
Defaulted         150
DebtIncomeRatio     0
dtype: int64
```

```
Edu                  0
Income               0
Card Debt            0
Other Debt           0
Defaulted          150
DebtIncomeRatio      0
dtype: int64
```

```
df.shape
```

```
(850, 6)
```

(850, 6)

```
df = df.iloc[:,[1,5]].values
```

```
df
```

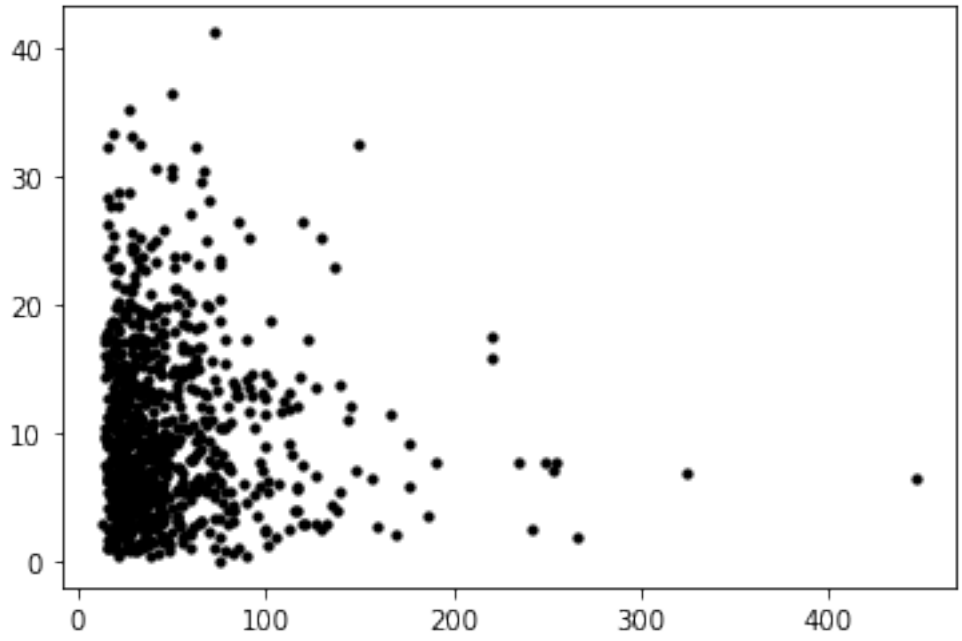
```
array([[ 19. ,   6.3],
       [100. ,  12.8],
       [ 57. ,  20.9],
       ...,
       [ 18. ,  33.4],
       [ 28. ,   2.9],
       [ 64. ,   8.6]])
```

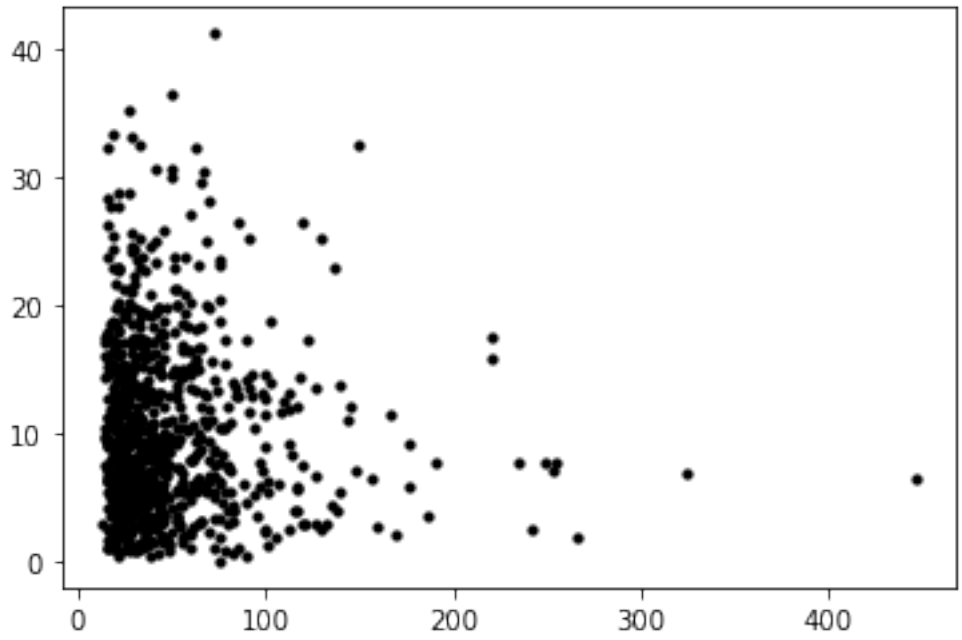
array([[19. ,   6.3],
       [100. , 12.8],
       [57. , 20.9],
       ...,
       [18. , 33.4],
       [28. ,   2.9],
       [64. ,   8.6]])

```
import matplotlib.pyplot as plt
plt.scatter(df[:,0],df[:,1],s = 10, c = "black")
```

```
<matplotlib.collections.PathCollection at 0x22f58540550>
```

<matplotlib.collections.PathCollection at 0x22f58540550>





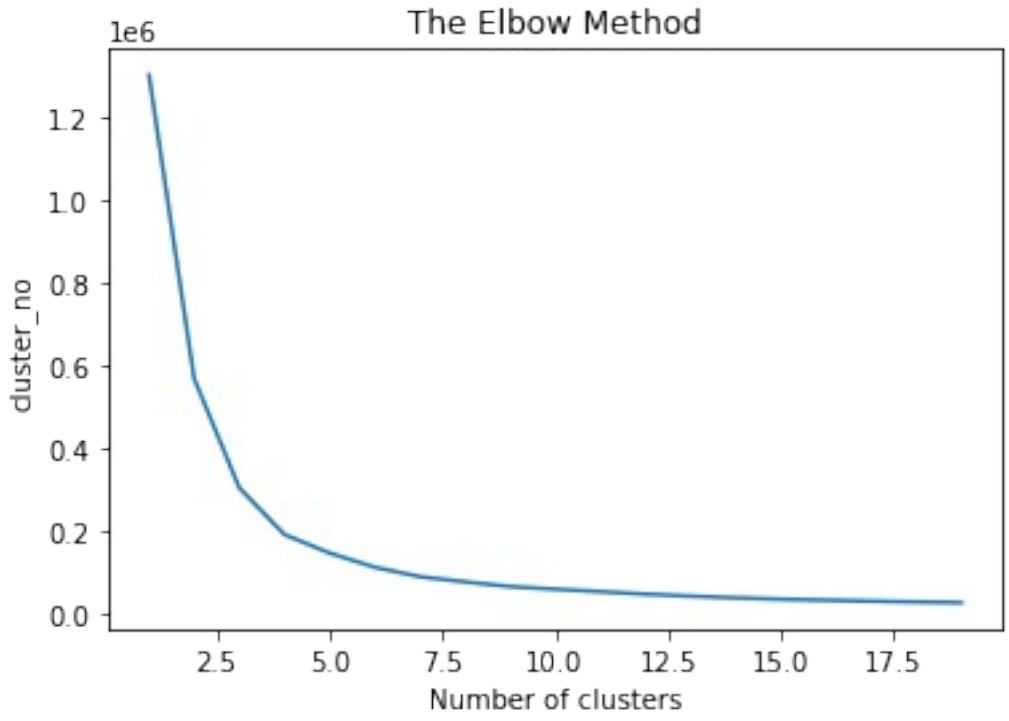

```
from sklearn.cluster import KMeans

cluster_no = []
for i in range(1,20):
    kmeans = KMeans(n_clusters = i,init = "k-means++",max_iter =
300,n_init = 100)
    kmeans.fit(df)
    cluster_no.append(kmeans.inertia_)
plt.plot(range(1,20),cluster_no)
plt.title("The Elbow Method")
```

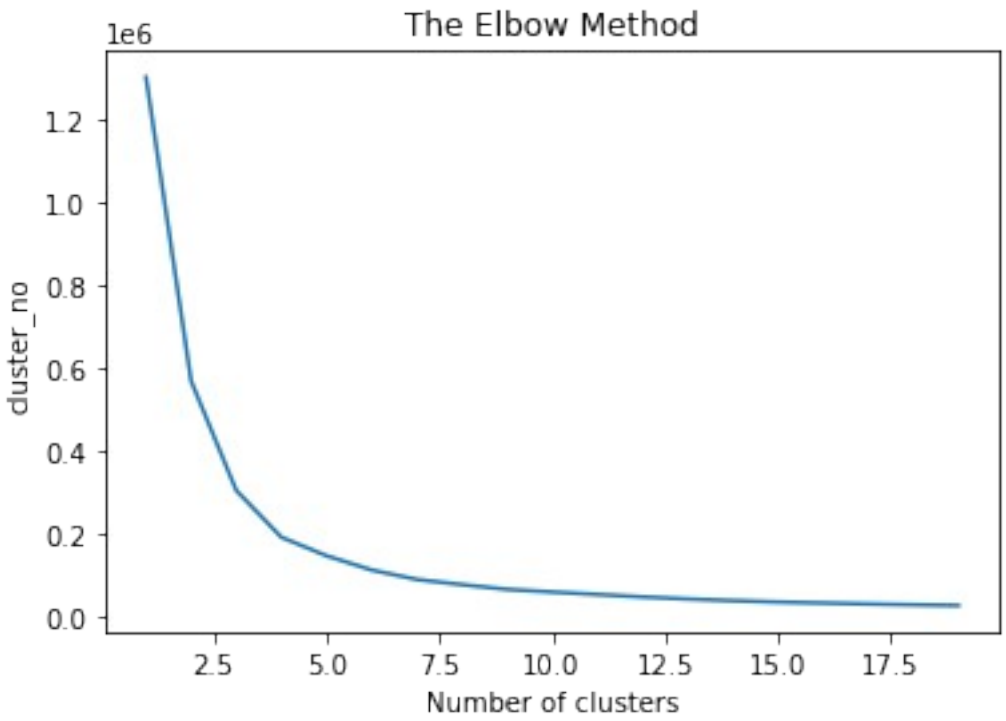
```
plt.xlabel("Number of clusters")
plt.ylabel("cluster_no")
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\cluster\
_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads.
You can avoid it by setting the environment variable
OMP_NUM_THREADS=4.
  warnings.warn(
C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\cluster\
_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads.
You can avoid it by setting the environment variable
OMP_NUM_THREADS=4.
  warnings.warn(
```

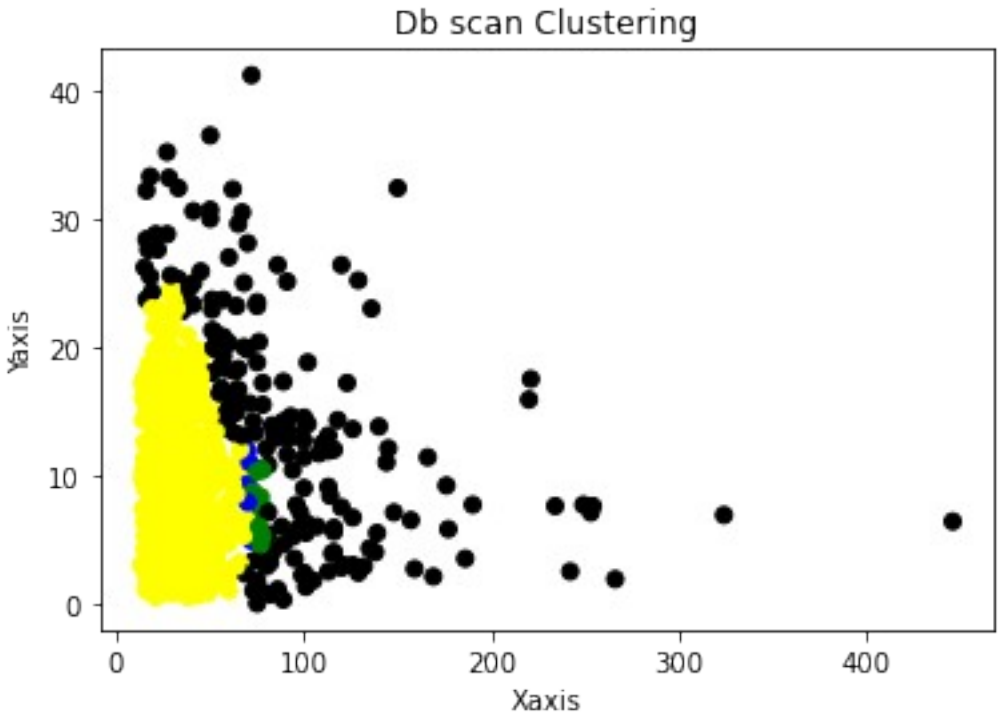
```
Text(0, 0.5, 'cluster_no')
```

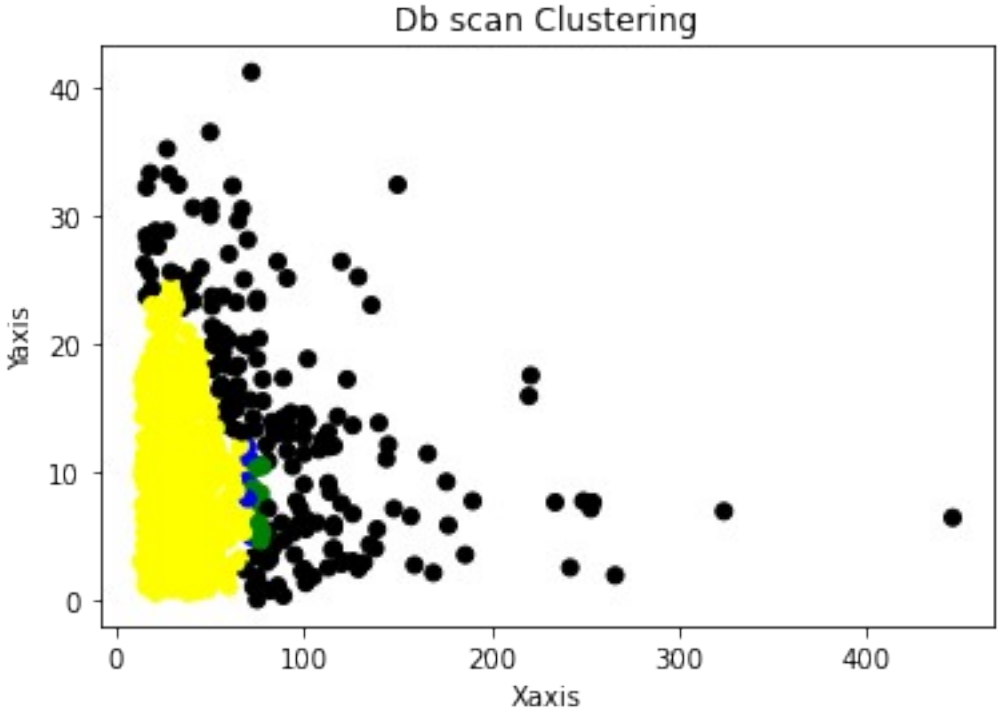
```
Text(0, 0.5, 'cluster_no')
```






```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps = 4,min_samples = 15)

labels = dbscan.fit_predict(df) #dbscan

np.unique(labels)

array([-1,  0,  1,  2], dtype=int64)

array([-1,  0,  1,  2], dtype=int64)

print(set(labels))

{0, 1, 2, -1}
{0, 1, 2, -1}

#-1,0,1,2
colormap = np.array(['yellow', 'green','blue','black'])
plt.scatter(df[:,0], df[:,1], color = colormap[labels])
plt.xlabel('Xaxis')
plt.ylabel('Yaxis')
plt.title("Db scan Clustering")
plt.show()
```


Db scan Clustering
Yaxis
Xaxis
0
10
20
30
40
0
100
200
300
400


Db scan Clustering
Yaxis
Xaxis
0
10
20
30
40
0
100
200
300
400

Que.2 Using dirtydata.csv Demonstrate all the techniques for removing the null values

● Replace by MEAN ● Replace by MEDIAN ● Replace by MODE ● Replace by ARBITUARY VALUE ● Replace by 0

```
import pandas as pd

df = pd.read_csv("C:/Users/Lenovo/Documents/Data Set/dirtydata.csv")

df
```

```
    Duration          Date  Pulse  Maxpulse  Calories
0         60  '2020/12/01'    110       130     409.1
1         60  '2020/12/02'    117       145     479.0
2         60  '2020/12/03'    103       135     340.0
3         45  '2020/12/04'    109       175     282.4
4         45  '2020/12/05'    117       148     406.0
5         60  '2020/12/06'    102       127     300.0
6         60  '2020/12/07'    110       136     374.0
7        450  '2020/12/08'    104       134     253.3
8         30  '2020/12/09'    109       133     195.1
9         60  '2020/12/10'     98       124     269.0
10        60  '2020/12/11'    103       147     329.3
11        60  '2020/12/12'    100       120     250.7
12        60  '2020/12/12'    100       120     250.7
13        60  '2020/12/13'    106       128     345.3
14        60  '2020/12/14'    104       132     379.3
15        60  '2020/12/15'     98       123     275.0
16        60  '2020/12/16'     98       120     215.2
17        60  '2020/12/17'    100       120     300.0
18        45  '2020/12/18'     90       112       NaN
19        60  '2020/12/19'    103       123     323.0
20        45  '2020/12/20'     97       125     243.0
21        60  '2020/12/21'    108       131     364.2
22        45           NaN    100       119     282.0
23        60  '2020/12/23'    130       101     300.0
24        45  '2020/12/24'    105       132     246.0
25        60  '2020/12/25'    102       126     334.5
26        60      20201226    100       120     250.0
27        60  '2020/12/27'     92       118     241.0
28        60  '2020/12/28'    103       132       NaN
29        60  '2020/12/29'    100       132     280.0
30        60  '2020/12/30'    102       129     380.3
31        60  '2020/12/31'     92       115     243.0
```

```
df.head()
```

```
   Duration          Date  Pulse  Maxpulse  Calories
0        60  '2020/12/01'    110       130     409.1
1        60  '2020/12/02'    117       145     479.0
```

```
2        60  '2020/12/03'    103       135     340.0
3        45  '2020/12/04'    109       175     282.4
4        45  '2020/12/05'    117       148     406.0
```

```
df.tail()
```

```
    Duration          Date  Pulse  Maxpulse  Calories
27        60  '2020/12/27'     92       118     241.0
28        60  '2020/12/28'    103       132       NaN
29        60  '2020/12/29'    100       132     280.0
30        60  '2020/12/30'    102       129     380.3
31        60  '2020/12/31'     92       115     243.0
```

```
df.isnull().sum()
```

```
Duration    0
Date        1
Pulse       0
Maxpulse    0
Calories    2
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Duration  32 non-null     int64  
 1   Date      31 non-null     object 
 2   Pulse     32 non-null     int64  
 3   Maxpulse  32 non-null     int64  
 4   Calories  30 non-null     float64
dtypes: float64(1), int64(3), object(1)
memory usage: 1.4+ KB
```

```
df[df['Calories'].isnull()]
```

```
    Duration          Date  Pulse  Maxpulse  Calories
18        45  '2020/12/18'     90       112       NaN
28        60  '2020/12/28'    103       132       NaN
```

```
df.columns
```

```
Index(['Duration', 'Date', 'Pulse', 'Maxpulse', 'Calories'],
dtype='object')
```

```
df.drop('Date', axis = 1)   #chnage is temporary
```

```
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
```

```
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7         450    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    100       120     250.7
13         60    106       128     345.3
14         60    104       132     379.3
15         60     98       123     275.0
16         60     98       120     215.2
17         60    100       120     300.0
18         45     90       112       NaN
19         60    103       123     323.0
20         45     97       125     243.0
21         60    108       131     364.2
22         45    100       119     282.0
23         60    130       101     300.0
24         45    105       132     246.0
25         60    102       126     334.5
26         60    100       120     250.0
27         60     92       118     241.0
28         60    103       132       NaN
29         60    100       132     280.0
30         60    102       129     380.3
31         60     92       115     243.0
```

```
df.drop('Date', axis = 1, inplace = True) # chnage is permenant
```

```
df
```

```
    Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7         450    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    100       120     250.7
13         60    106       128     345.3
14         60    104       132     379.3
```

```
15         60     98        123     275.0
16         60     98        120     215.2
17         60    100        120     300.0
18         45     90        112       NaN
19         60    103        123     323.0
20         45     97        125     243.0
21         60    108        131     364.2
22         45    100        119     282.0
23         60    130        101     300.0
24         45    105        132     246.0
25         60    102        126     334.5
26         60    100        120     250.0
27         60     92        118     241.0
28         60    103        132       NaN
29         60    100        132     280.0
30         60    102        129     380.3
31         60     92        115     243.0
```

```
df.columns
```

```
Index(['Duration', 'Pulse', 'Maxpulse', 'Calories'], dtype='object')
```

```
null_values_present_in_Calories=df['Calories'].isnull().sum()
total_no_rows = len(df)
print(f"%age of null values in Calories column is
{(null_values_present_in_Calories/total_no_rows) * 100}%")
```

```
%age of null values in Calories column is 6.25%
```

```
null_values_present_in_Calories
```

```
2
```

```
total_no_rows
```

```
32
```

Mean/ Median /Mode imputation

```
df=pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/dirtydata.csv",usecols=['Duration','Pulse','Maxpulse',"Calories"])
df.head()
```

```
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
2        60    103       135     340.0
3        45    109       175     282.4
4        45    117       148     406.0
```

```
df
```

```
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7        450    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    100       120     250.7
13        60    106       128     345.3
14        60    104       132     379.3
15        60     98       123     275.0
16        60     98       120     215.2
17        60    100       120     300.0
18        45     90       112       NaN
19        60    103       123     323.0
20        45     97       125     243.0
21        60    108       131     364.2
22        45    100       119     282.0
23        60    130       101     300.0
24        45    105       132     246.0
25        60    102       126     334.5
26        60    100       120     250.0
27        60     92       118     241.0
28        60    103       132       NaN
29        60    100       132     280.0
30        60    102       129     380.3
31        60     92       115     243.0

len(df)

32

df.isnull().sum()

Duration    0
Pulse       0
Maxpulse    0
Calories    2
dtype: int64

## Lets go and see the percentage of missing values
df.isnull().mean()

Duration    0.0000
Pulse       0.0000
```

```
Maxpulse    0.0000
Calories    0.0625
dtype: float64
```

Mean

```
def impute_nan(df,variable,value):
    df[variable+"_mean"]=df[variable].fillna(value)

Calories_mean = df.Calories.mean()

Calories_mean

304.68

impute_nan(df,'Calories',Calories_mean)

df
```

	Duration	Pulse	Maxpulse	Calories	Calories_mean
0	60	110	130	409.1	409.10
1	60	117	145	479.0	479.00
2	60	103	135	340.0	340.00
3	45	109	175	282.4	282.40
4	45	117	148	406.0	406.00
5	60	102	127	300.0	300.00
6	60	110	136	374.0	374.00
7	450	104	134	253.3	253.30
8	30	109	133	195.1	195.10
9	60	98	124	269.0	269.00
10	60	103	147	329.3	329.30
11	60	100	120	250.7	250.70
12	60	100	120	250.7	250.70
13	60	106	128	345.3	345.30
14	60	104	132	379.3	379.30
15	60	98	123	275.0	275.00
16	60	98	120	215.2	215.20
17	60	100	120	300.0	300.00
18	45	90	112	NaN	304.68
19	60	103	123	323.0	323.00
20	45	97	125	243.0	243.00
21	60	108	131	364.2	364.20
22	45	100	119	282.0	282.00
23	60	130	101	300.0	300.00
24	45	105	132	246.0	246.00
25	60	102	126	334.5	334.50
26	60	100	120	250.0	250.00
27	60	92	118	241.0	241.00
28	60	103	132	NaN	304.68
29	60	100	132	280.0	280.00

```
30         60    102       129     380.3         380.30
31         60     92       115     243.0         243.00
```

```
df[df['Calories'].isnull()]
```

```
    Duration  Pulse  Maxpulse  Calories  Calories_mean
18         45     90       112       NaN         304.68
28         60    103       132       NaN         304.68
```

Median

```
def impute_nan_median(df,variable,value):
    df[variable+"_median"]=df[variable].fillna(value)
```

```
median=df.Calories.median()
median
print("the median from Calories column which is available in df
dataframe is",median)
```

```
the median from Calories column which is available in df dataframe is
291.2
```

```
impute_nan_median(df,'Calories',median)
```

```
df
```

```
    Duration  Pulse  Maxpulse  Calories  Calories_mean
Calories_median
0          60    110       130     409.1         409.10
409.1
1          60    117       145     479.0         479.00
479.0
2          60    103       135     340.0         340.00
340.0
3          45    109       175     282.4         282.40
282.4
4          45    117       148     406.0         406.00
406.0
5          60    102       127     300.0         300.00
300.0
6          60    110       136     374.0         374.00
374.0
7         450    104       134     253.3         253.30
253.3
8          30    109       133     195.1         195.10
195.1
9          60     98       124     269.0         269.00
269.0
10         60    103       147     329.3         329.30
329.3
11         60    100       120     250.7         250.70
```

```
250.7
12        60    100       120     250.7         250.70
250.7
13        60    106       128     345.3         345.30
345.3
14        60    104       132     379.3         379.30
379.3
15        60     98       123     275.0         275.00
275.0
16        60     98       120     215.2         215.20
215.2
17        60    100       120     300.0         300.00
300.0
18        45     90       112       NaN         304.68
291.2
19        60    103       123     323.0         323.00
323.0
20        45     97       125     243.0         243.00
243.0
21        60    108       131     364.2         364.20
364.2
22        45    100       119     282.0         282.00
282.0
23        60    130       101     300.0         300.00
300.0
24        45    105       132     246.0         246.00
246.0
25        60    102       126     334.5         334.50
334.5
26        60    100       120     250.0         250.00
250.0
27        60     92       118     241.0         241.00
241.0
28        60    103       132       NaN         304.68
291.2
29        60    100       132     280.0         280.00
280.0
30        60    102       129     380.3         380.30
380.3
31        60     92       115     243.0         243.00
243.0
```

```
df.head()
```

	Duration	Pulse	Maxpulse	Calories	Calories_mean	Calories_median
0	60	110	130	409.1	409.1	409.1
1	60	117	145	479.0	479.0	479.0
2	60	103	135	340.0	340.0	340.0
3	45	109	175	282.4	282.4	282.4
4	45	117	148	406.0	406.0	406.0

```
df.tail()
```

```
    Duration  Pulse  Maxpulse  Calories  Calories_mean
Calories_median
27        60     92       118     241.0         241.00
241.0
28        60    103       132       NaN         304.68
291.2
29        60    100       132     280.0         280.00
280.0
30        60    102       129     380.3         380.30
380.3
31        60     92       115     243.0         243.00
243.0
```

```
df[df['Calories'].isnull()]
```

```
    Duration  Pulse  Maxpulse  Calories  Calories_mean
Calories_median
18        45     90       112       NaN         304.68
291.2
28        60    103       132       NaN         304.68
291.2
```

```
df.iloc[28] # 28 index
```

```
Duration            60.00
Pulse              103.00
Maxpulse           132.00
Calories              NaN
Calories_mean      304.68
Calories_median    291.20
Name: 28, dtype: float64
```

```
print(df['Calories'].std())        # Null values
print(df['Calories_median'].std())
```

```
66.00377941111951
63.925151544599416
```

```
print(df['Calories_mean'].std())
```

```
63.83912947098279
```

```
df['Calories'].isnull().sum()
```

```
2
```

```
df['Calories_median'].isnull().sum()
```

```
0
```

```
df.isnull().sum()
```

```
Duration           0
Pulse              0
Maxpulse           0
Calories           2
Calories_mean      0
Calories_median    0
dtype: int64
```

Mode

```
def impute_nan_mode(df,variable,value):
    print(value)
    df[variable+"_mode"]=df[variable].fillna(value)

mode = df.Calories.mode()
type(mode)

pandas.core.series.Series

mode

0    300.0
Name: Calories, dtype: float64

mode[0]

300.0

print(mode)
impute_nan_mode(df,'Calories',mode[0])
df

0    300.0
Name: Calories, dtype: float64
300.0

    Duration  Pulse  Maxpulse  Calories  Calories_mean
Calories_median  \
0         60    110       130     409.1         409.10
409.1
1         60    117       145     479.0         479.00
479.0
2         60    103       135     340.0         340.00
340.0
3         45    109       175     282.4         282.40
282.4
4         45    117       148     406.0         406.00
406.0
5         60    102       127     300.0         300.00
300.0
6         60    110       136     374.0         374.00
374.0
```

7	450	104	134	253.3	253.30	253.3
8	30	109	133	195.1	195.10	195.1
9	60	98	124	269.0	269.00	269.0
10	60	103	147	329.3	329.30	329.3
11	60	100	120	250.7	250.70	250.7
12	60	100	120	250.7	250.70	250.7
13	60	106	128	345.3	345.30	345.3
14	60	104	132	379.3	379.30	379.3
15	60	98	123	275.0	275.00	275.0
16	60	98	120	215.2	215.20	215.2
17	60	100	120	300.0	300.00	300.0
18	45	90	112	NaN	304.68	291.2
19	60	103	123	323.0	323.00	323.0
20	45	97	125	243.0	243.00	243.0
21	60	108	131	364.2	364.20	364.2
22	45	100	119	282.0	282.00	282.0
23	60	130	101	300.0	300.00	300.0
24	45	105	132	246.0	246.00	246.0
25	60	102	126	334.5	334.50	334.5
26	60	100	120	250.0	250.00	250.0
27	60	92	118	241.0	241.00	241.0
28	60	103	132	NaN	304.68	291.2
29	60	100	132	280.0	280.00	280.0
30	60	102	129	380.3	380.30	380.3
31	60	92	115	243.0	243.00	243.0

```
    Calories_mode
0           409.1
1           479.0
2           340.0
3           282.4
4           406.0
5           300.0
6           374.0
7           253.3
8           195.1
9           269.0
10          329.3
11          250.7
12          250.7
13          345.3
14          379.3
15          275.0
16          215.2
17          300.0
18          300.0
19          323.0
20          243.0
21          364.2
22          282.0
23          300.0
24          246.0
25          334.5
26          250.0
27          241.0
28          300.0
29          280.0
30          380.3
31          243.0
```

```
df.head()
```

```
   Duration  Pulse  Maxpulse  Calories  Calories_mean  Calories_median
\
0        60    110       130     409.1          409.1            409.1

1        60    117       145     479.0          479.0            479.0

2        60    103       135     340.0          340.0            340.0

3        45    109       175     282.4          282.4            282.4

4        45    117       148     406.0          406.0            406.0
```

```
   Calories_mode
0          409.1
1          479.0
2          340.0
3          282.4
4          406.0
```

Replace with 0

```
df["Calories_Replace_with_0"] = df["Calories"]
df.fillna(0)
```

```
    Duration  Pulse  Maxpulse  Calories  Calories_mean
Calories_median  \
0         60    110       130     409.1         409.10
409.1
1         60    117       145     479.0         479.00
479.0
2         60    103       135     340.0         340.00
340.0
3         45    109       175     282.4         282.40
282.4
4         45    117       148     406.0         406.00
406.0
5         60    102       127     300.0         300.00
300.0
6         60    110       136     374.0         374.00
374.0
7        450    104       134     253.3         253.30
253.3
8         30    109       133     195.1         195.10
195.1
9         60     98       124     269.0         269.00
269.0
10        60    103       147     329.3         329.30
329.3
11        60    100       120     250.7         250.70
250.7
12        60    100       120     250.7         250.70
250.7
13        60    106       128     345.3         345.30
345.3
14        60    104       132     379.3         379.30
379.3
15        60     98       123     275.0         275.00
275.0
16        60     98       120     215.2         215.20
215.2
17        60    100       120     300.0         300.00
300.0
```

```
18        45     90       112       0.0         304.68
291.2
19        60    103       123     323.0         323.00
323.0
20        45     97       125     243.0         243.00
243.0
21        60    108       131     364.2         364.20
364.2
22        45    100       119     282.0         282.00
282.0
23        60    130       101     300.0         300.00
300.0
24        45    105       132     246.0         246.00
246.0
25        60    102       126     334.5         334.50
334.5
26        60    100       120     250.0         250.00
250.0
27        60     92       118     241.0         241.00
241.0
28        60    103       132       0.0         304.68
291.2
29        60    100       132     280.0         280.00
280.0
30        60    102       129     380.3         380.30
380.3
31        60     92       115     243.0         243.00
243.0

    Calories_mode  Calories_Replace_with_0
0           409.1                    409.1
1           479.0                    479.0
2           340.0                    340.0
3           282.4                    282.4
4           406.0                    406.0
5           300.0                    300.0
6           374.0                    374.0
7           253.3                    253.3
8           195.1                    195.1
9           269.0                    269.0
10          329.3                    329.3
11          250.7                    250.7
12          250.7                    250.7
13          345.3                    345.3
14          379.3                    379.3
15          275.0                    275.0
16          215.2                    215.2
17          300.0                    300.0
18          300.0                      0.0
19          323.0                    323.0
```

```
20          243.0                     243.0
21          364.2                     364.2
22          282.0                     282.0
23          300.0                     300.0
24          246.0                     246.0
25          334.5                     334.5
26          250.0                     250.0
27          241.0                     241.0
28          300.0                       0.0
29          280.0                     280.0
30          380.3                     380.3
31          243.0                     243.0
```

Replace with arbitary values

```
df["Calories_Replace_with_arbitary_values"] = df["Calories"]
df.fillna(100)
```

```
    Duration  Pulse  Maxpulse  Calories  Calories_mean
Calories_median  \
0         60    110       130     409.1         409.10
409.1
1         60    117       145     479.0         479.00
479.0
2         60    103       135     340.0         340.00
340.0
3         45    109       175     282.4         282.40
282.4
4         45    117       148     406.0         406.00
406.0
5         60    102       127     300.0         300.00
300.0
6         60    110       136     374.0         374.00
374.0
7        450    104       134     253.3         253.30
253.3
8         30    109       133     195.1         195.10
195.1
9         60     98       124     269.0         269.00
269.0
10        60    103       147     329.3         329.30
329.3
11        60    100       120     250.7         250.70
250.7
12        60    100       120     250.7         250.70
250.7
13        60    106       128     345.3         345.30
345.3
14        60    104       132     379.3         379.30
379.3
```

```
15         60     98       123     275.0         275.00
275.0
16         60     98       120     215.2         215.20
215.2
17         60    100       120     300.0         300.00
300.0
18         45     90       112     100.0         304.68
291.2
19         60    103       123     323.0         323.00
323.0
20         45     97       125     243.0         243.00
243.0
21         60    108       131     364.2         364.20
364.2
22         45    100       119     282.0         282.00
282.0
23         60    130       101     300.0         300.00
300.0
24         45    105       132     246.0         246.00
246.0
25         60    102       126     334.5         334.50
334.5
26         60    100       120     250.0         250.00
250.0
27         60     92       118     241.0         241.00
241.0
28         60    103       132     100.0         304.68
291.2
29         60    100       132     280.0         280.00
280.0
30         60    102       129     380.3         380.30
380.3
31         60     92       115     243.0         243.00
243.0

    Calories_mode  Calories_Replace_with_0  \
0           409.1                    409.1
1           479.0                    479.0
2           340.0                    340.0
3           282.4                    282.4
4           406.0                    406.0
5           300.0                    300.0
6           374.0                    374.0
7           253.3                    253.3
8           195.1                    195.1
9           269.0                    269.0
10          329.3                    329.3
11          250.7                    250.7
12          250.7                    250.7
13          345.3                    345.3
```

14	379.3	379.3
15	275.0	275.0
16	215.2	215.2
17	300.0	300.0
18	300.0	100.0
19	323.0	323.0
20	243.0	243.0
21	364.2	364.2
22	282.0	282.0
23	300.0	300.0
24	246.0	246.0
25	334.5	334.5
26	250.0	250.0
27	241.0	241.0
28	300.0	100.0
29	280.0	280.0
30	380.3	380.3
31	243.0	243.0

	Calories_Replace_with_arbitary_values
0	409.1
1	479.0
2	340.0
3	282.4
4	406.0
5	300.0
6	374.0
7	253.3
8	195.1
9	269.0
10	329.3
11	250.7
12	250.7
13	345.3
14	379.3
15	275.0
16	215.2
17	300.0
18	100.0
19	323.0
20	243.0
21	364.2
22	282.0
23	300.0
24	246.0
25	334.5
26	250.0
27	241.0
28	100.0
29	280.0

```
30                                  380.3
31                                  243.0
```

```
print(f"std of original {df['Calories'].std()} mean
{df['Calories_mean'].std()}")
print(f"std of meadian {df['Calories_median'].std()} mode
{df['Calories_mode'].std()}")
print(f"std of Replace with 0 : {df['Calories_Replace_with_0'].std()}
Replace with arbitary values
{df['Calories_Replace_with_arbitary_values'].std()}")
```

```
---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Input In [2], in <cell line: 1>()
----> 1 print(f"std of original {df['Calories'].std()} mean
{df['Calories_mean'].std()}")
      2 print(f"std of meadian {df['Calories_median'].std()} mode
{df['Calories_mode'].std()}")
      3 print(f"std of Replace with 0 :
{df['Calories_Replace_with_0'].std()} Replace with arbitary values
{df['Calories_Replace_with_arbitary_values'].std()}")

NameError: name 'df' is not defined
```

Conclusion : For this example we can use the mean column because mean column having the less standard deviation as compared to the other columns.

So,therefore we can remove the other columns and only use the mean column for building a model

```
df.drop(['Calories','Calories_median','Calories_mode',"Calories_Replac
e_with_0","Calories_Replace_with_arbitary_values"],inplace = True,axis
= 1)
```

```
df
```

	Duration	Pulse	Maxpulse	Calories_mean
0	60	110	130	409.10
1	60	117	145	479.00
2	60	103	135	340.00
3	45	109	175	282.40
4	45	117	148	406.00
5	60	102	127	300.00
6	60	110	136	374.00
7	450	104	134	253.30

8	30	109	133	195.10
9	60	98	124	269.00
10	60	103	147	329.30
11	60	100	120	250.70
12	60	100	120	250.70
13	60	106	128	345.30
14	60	104	132	379.30
15	60	98	123	275.00
16	60	98	120	215.20
17	60	100	120	300.00
18	45	90	112	304.68
19	60	103	123	323.00
20	45	97	125	243.00
21	60	108	131	364.20
22	45	100	119	282.00
23	60	130	101	300.00
24	45	105	132	246.00
25	60	102	126	334.50
26	60	100	120	250.00
27	60	92	118	241.00
28	60	103	132	304.68
29	60	100	132	280.00
30	60	102	129	380.30
31	60	92	115	243.00

These are the columns are used for the further analysis