In [1]:
```
# Aim: To perform and find the accuracy of K means algorithm
```

In [2]:
```
# Name : Shriya Mechineni
# class : 3rd year
# Section : A
# Roll No. :49
```

In [3]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # for data visualization
import seaborn as sns # for statistical data visualization
%matplotlib inline
from sklearn.datasets import make_blobs
import warnings
warnings.filterwarnings('ignore')
import os
```

In [4]:
```
os.getcwd()
```

Out[4]:
```
'C:\\Users\\admin'
```

In [5]:
```
os.chdir ("C:\\Users\\admin\\Desktop")
```

In [6]:
```
df=pd.read_csv('CHD_preprocessed.csv')
```

In [7]:
```
df.head()
```

Out[7]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol
0	1	39	1	0	0.0	0.0	0	0	0	195.0
1	0	46	0	0	0.0	0.0	0	0	0	250.0
2	1	48	0	1	20.0	0.0	0	0	0	245.0
3	0	61	1	1	30.0	0.0	0	1	0	225.0
4	0	46	1	1	23.0	0.0	0	0	0	285.0

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4133 entries, 0 to 4132
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4133 non-null   int64  
 1   age              4133 non-null   int64  
 2   education        4133 non-null   int64  
 3   currentSmoker    4133 non-null   int64  
 4   cigsPerDay       4133 non-null   float64
 5   BPMeds           4133 non-null   float64
 6   prevalentStroke  4133 non-null   int64  
 7   prevalentHyp     4133 non-null   int64  
 8   diabetes         4133 non-null   int64  
 9   totChol          4133 non-null   float64
 10  sysBP            4133 non-null   float64
 11  diaBP            4133 non-null   float64
 12  BMI              4133 non-null   float64
 13  heartRate        4133 non-null   float64
 14  glucose          4133 non-null   float64
 15  TenYearCHD       4133 non-null   int64  
dtypes: float64(8), int64(8)
memory usage: 516.8 KB
```

In [9]:

```
df.size
```

Out[9]:

```
66128
```

In [10]:

```
df.shape
```

Out[10]:

```
(4133, 16)
```

In [11]:

```
df.describe()
```

Out[11]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	preva
count	4133.000000	4133.000000	4133.000000	4133.000000	4133.000000	4133.000000	4133.000000	4133
mean	0.427293	49.557222	0.280668	0.494798	9.101621	0.034358	0.006049	0
std	0.494745	8.561628	0.449380	0.500033	11.918440	0.182168	0.077548	0
min	0.000000	32.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
25%	0.000000	42.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
50%	0.000000	49.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
75%	1.000000	56.000000	1.000000	1.000000	20.000000	0.000000	0.000000	1
max	1.000000	70.000000	1.000000	1.000000	70.000000	1.000000	1.000000	1

In [12]:

```
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score
```

In [13]:

```
X, y = make_blobs(n_samples=1000, centers=3, n_features=2, random_state=42)
```

In [14]:

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2, random_state=0)
kmeans.fit(X)
```

```
Out[14]:  ▼ KMeans
          KMeans(n_clusters=2, random_state=0)
```

```
In [15]:  kmeans.cluster_centers_

Out[15]:  array([[-6.86656253, -6.78519427],
                 [ 1.09680929,  5.53496642]])
```

```
In [16]:  kmeans.inertia_

Out[16]:  18773.018255690862
```

```
In [ ]:
```

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js