Practical no. 5

AIM: Perform the data classification algorithm using any Clustering algorithm

CLUSTERING ALGORITHM -- KMeans Clustering

Importing Libraries

In [7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
```

Loading Dataset

In [2]:
```
iris = load_iris()
X = iris.data
```

Converting dataset to DataFrame

In [8]:
```
df= pd.DataFrame(data=iris.data,columns=iris.feature_names)
```

In [9]:
```
df.head()
```

Out[9]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [10]:
```
df.tail()
```

Out[10]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

In [11]:

```
df.shape
```

Out[11]:

```
(150, 4)
```

In [12]:

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

Out[12]:

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
dtype: int64
```

In [13]:

```
df.describe()
```

Out[13]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

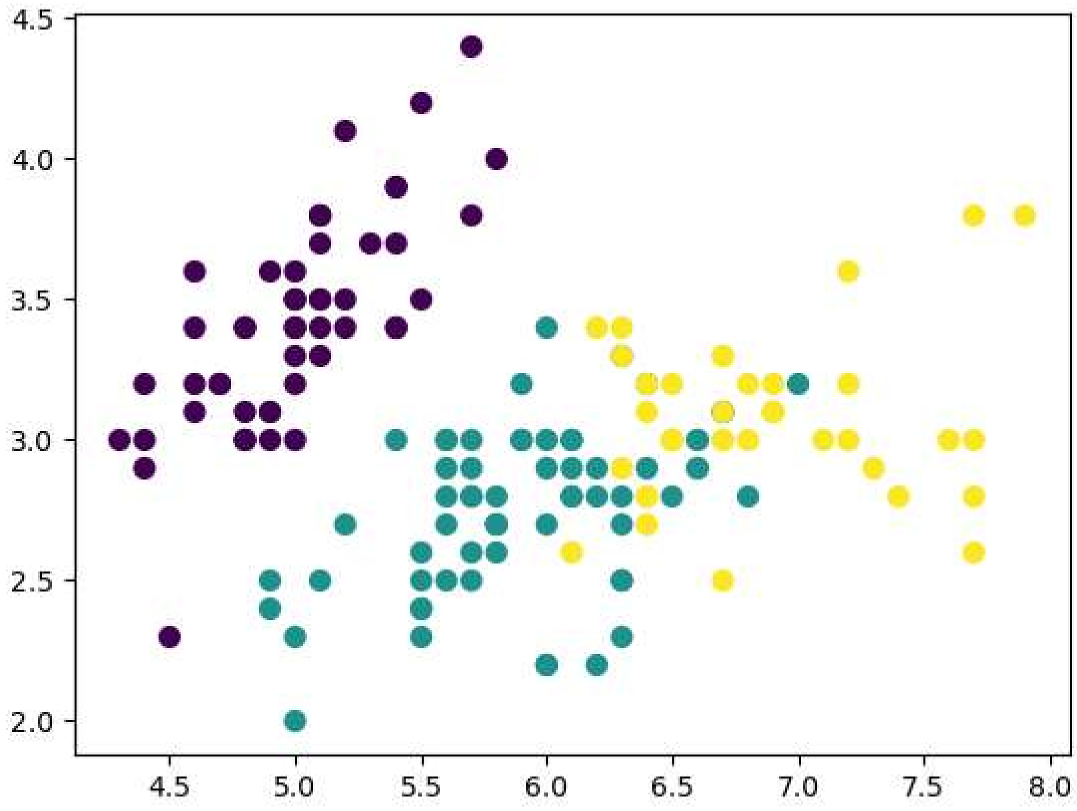
In [3]:

```
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1
334: UserWarning: KMeans is known to have a memory leak on Windows with
MKL, when there are less chunks than available threads. You can avoid i
t by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Clusters plotting

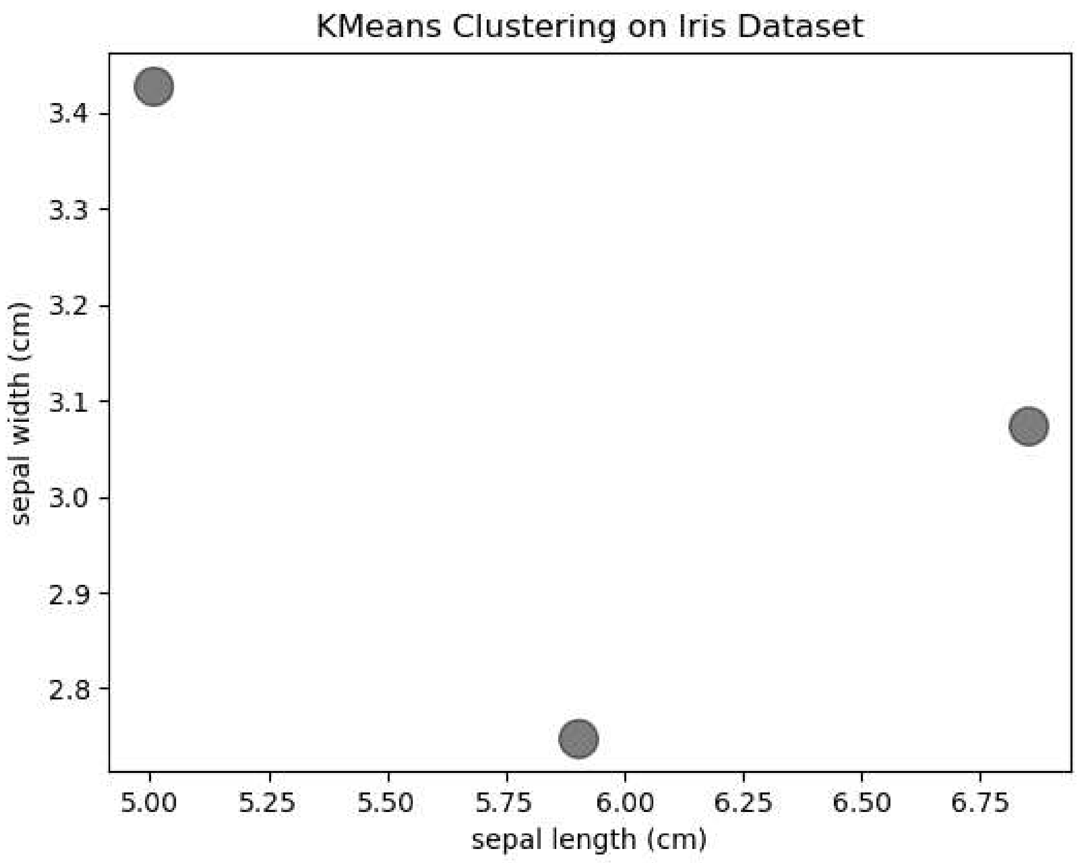
```
In [4]:  plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')

Out[4]: <matplotlib.collections.PathCollection at 0x22d452b6cd0>
```



Plotting the centroids of the clusters

In [5]:

```
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.title("KMeans Clustering on Iris Dataset")
plt.show()
```



In [ ]:

```
```