Task_2

K-means_clustering

Import the libaries

In [8]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Import the dataset

In [9]:

```python
df=pd.read_csv(r"C:\Users\VAISHNAVI\Downloads\iris.csv")
```

In [10]:

```python
df.head()
```

Out[10]:

	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

In [11]:

```python
df1=df.iloc[:,1:6]
```

In [12]:

```python
df1.head()
```

Out[12]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [13]:

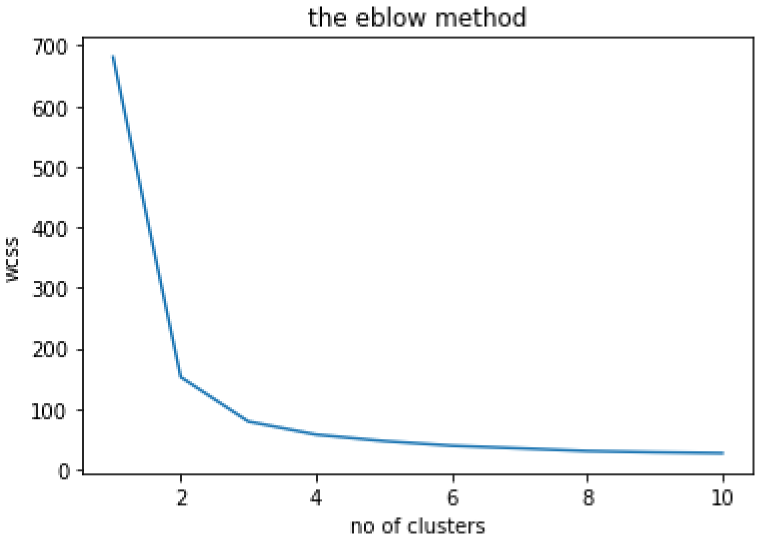
```python
X=df1.iloc[:,0:4].values

```

Elbow method

In [14]:

```
from sklearn.cluster import KMeans
wcss=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.title("the eblow method")
plt.xlabel("no of clusters")
plt.ylabel("wcss")
plt.show()
```

```
C:\Users\VAISHNAVI\Anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when th
ere are less chunks than available threads. You can avoid it by setting the env
ironment variable OMP_NUM_THREADS=1.
  warnings.warn(
```



Kmeans_clustering

In [15]:

```
kmeans=KMeans(n_clusters=3,init='k-means++',random_state=42)
y_means=kmeans.fit_predict(X)
```

In [16]:

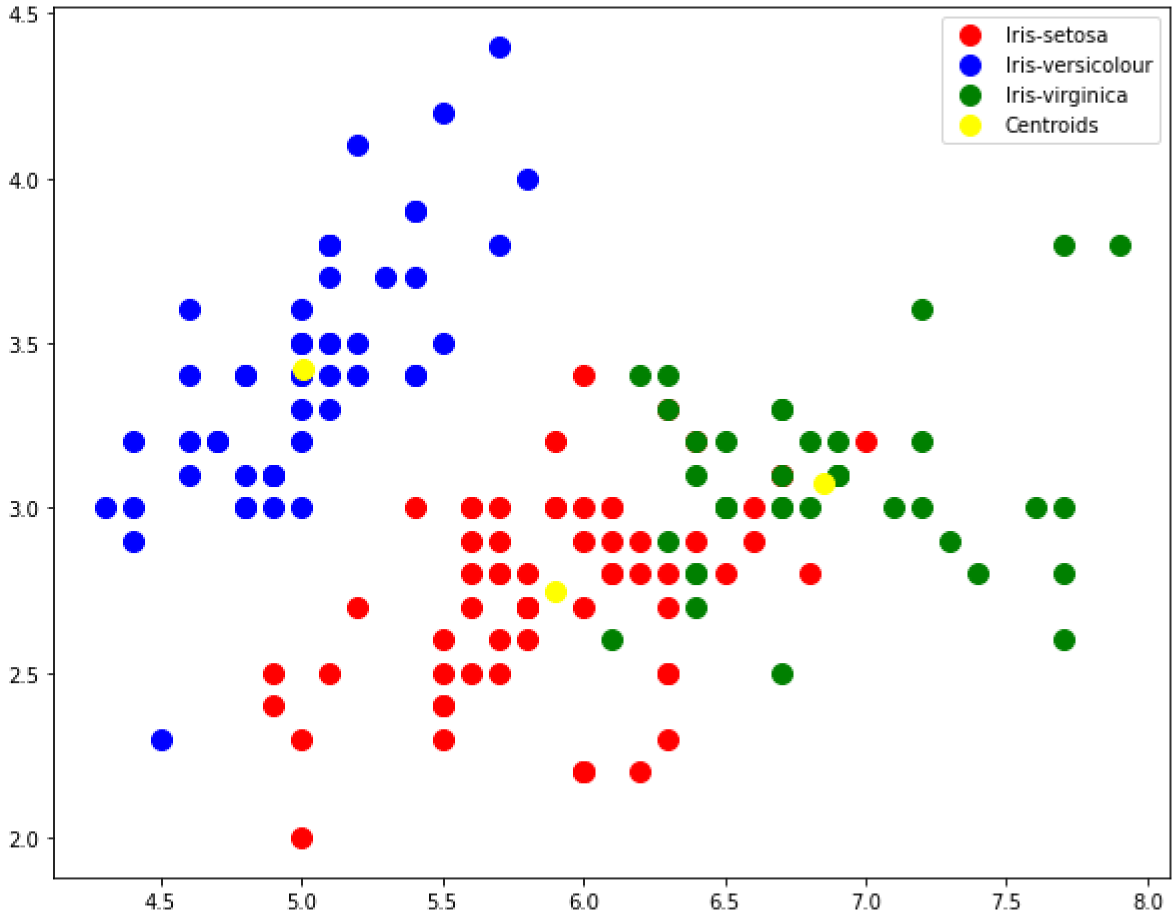
```
print(y_means)
```

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]
```

Visualise the result

In [17]:

```python
# Visualising the clusters - On the first two columns
plt.figure(figsize=(10,8))
plt.scatter(X[y_means == 0, 0],X[y_means == 0, 1],s=100,c='red',label='Iris-
plt.scatter(X[y_means == 1, 0],X[y_means == 1, 1],s =100,c='blue',label='Iri
plt.scatter(X[y_means == 2, 0],X[y_means == 2, 1],s=100, c='green', label='I
# Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0],kmeans.cluster_centers_[:,1],s=100
plt.legend(loc=1)
plt.show()
```



In [ ]:

```python

```