```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans


#load iris dataset
iris=load_iris()
X=iris.data
y=iris.target


iris


df=pd.DataFrame(X)
df
```

	0	1	2	3
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
...	...	...	...	...
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

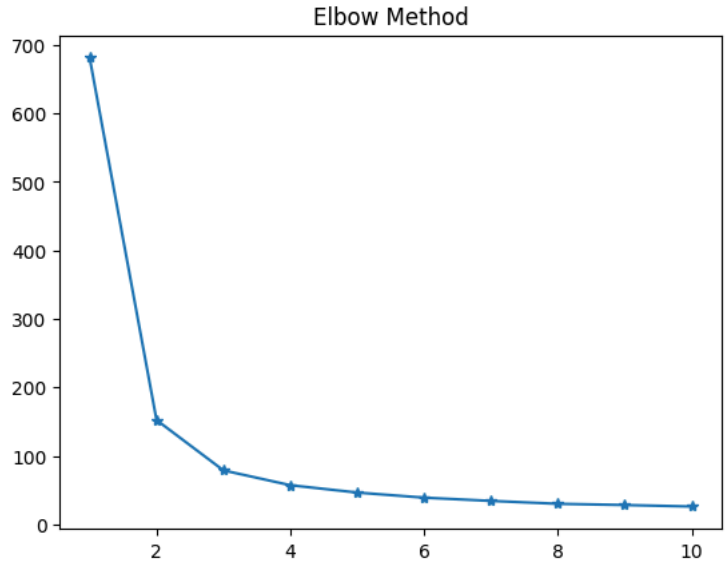
150 rows × 4 columns

```
#implement elbow method
wcss=[] #Within-Cluster Sum of Squares

for i in range(1,11):
  kmeans =KMeans(n_clusters=i,n_init=10,random_state=0)
  kmeans.fit(X)
  wcss.append(kmeans.inertia_)


#plot the elbow method
plt.plot(range(1,11),wcss,marker="*")
plt.title("Elbow Method")
```

```
Text(0.5, 1.0, 'Elbow Method')
```

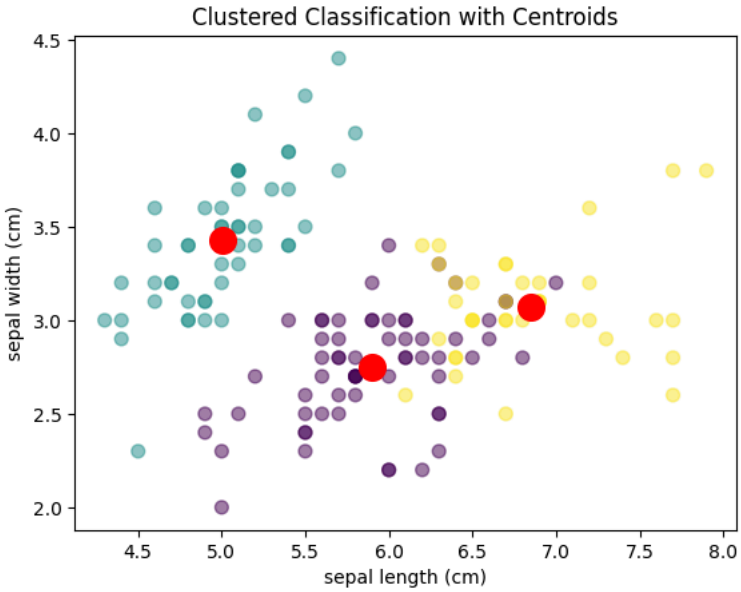


```
k_optimal=3
kmeans=KMeans(n_clusters=k_optimal,init='k-means++',max_iter=300,n_init=10,random_state=0)
kmeans.fit(X)
```

```
KMeans
KMeans(n_clusters=3, n_init=10, random_state=0)
```

```
#get centroids
centroids=kmeans.cluster_centers_
labels=kmeans.labels_
```

```
#visualize the clusters
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis', s=50, alpha=0.5)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, marker='o')
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.title('Clustered Classification with Centroids')
plt.show()
```



Start coding or generate with AI.