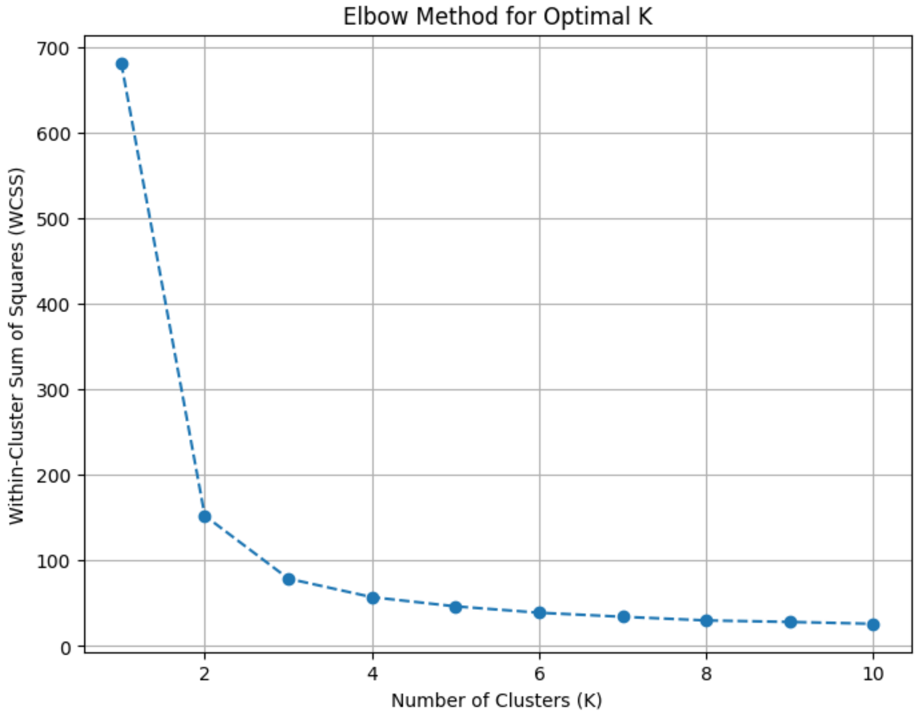
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as malplt
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
```

Load the Iris dataset

```
irisData = load_iris()
```

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(irisData.data)
    wcss.append(kmeans.inertia_)
```

```
malplt.figure(figsize=(8, 6))
malplt.plot(range(1, 11), wcss, marker='o', linestyle='--')
malplt.xlabel('Number of Clusters (K)')
malplt.ylabel('Within-Cluster Sum of Squares (WCSS)')
malplt.title('Elbow Method for Optimal K')
malplt.grid()
malplt.show()
```



```
kValue = 3 #Choosing the optimal K value as 3
```

```
kmeans = KMeans(n_clusters=kValue, init='k-means++', max_iter=300, n_init=10, random_state=0)
kmeans.fit(irisData.data) #K-Means clustering
```

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

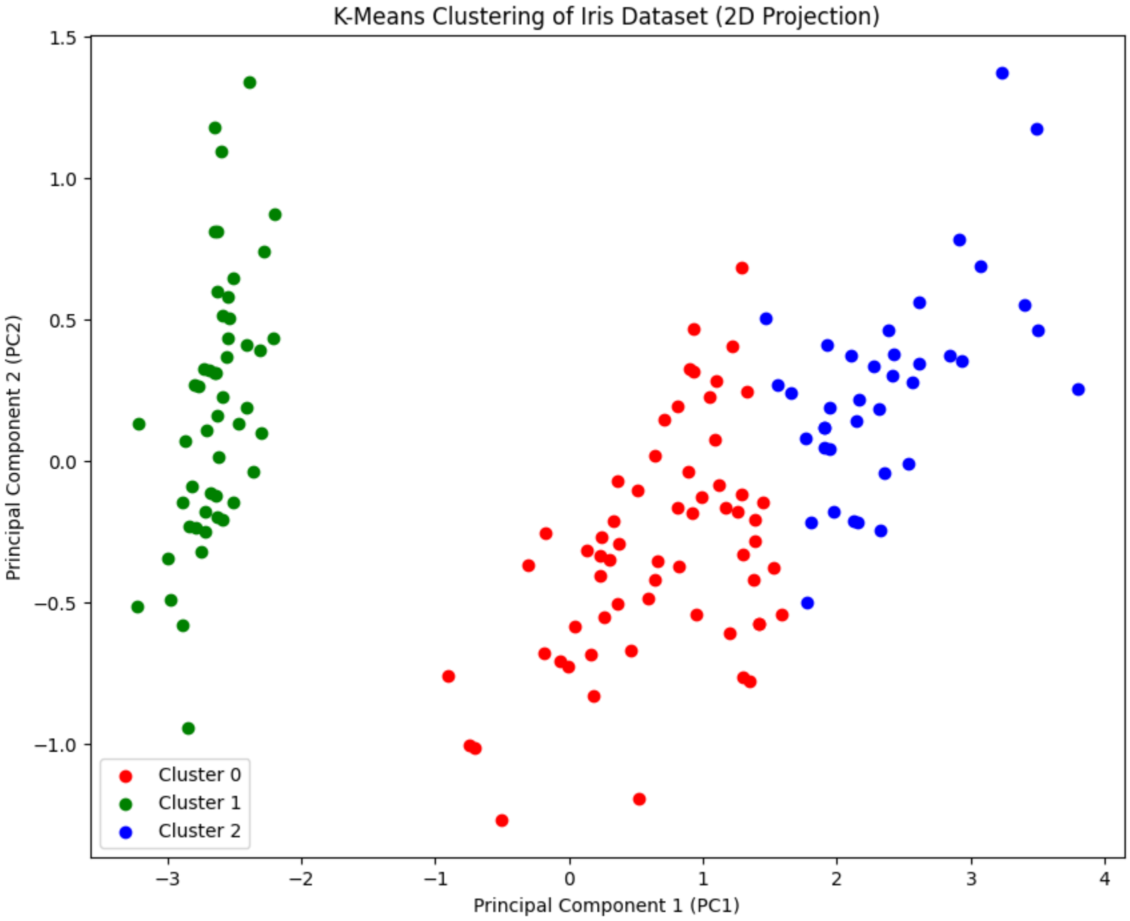
```
iris_df = pd.DataFrame(X_2d, columns=['PC1', 'PC2']) #give labels to the data
iris_df['Cluster'] = kmeans.labels_
```

```
pca = PCA(n_components=2)
X_2d = pca.fit_transform(irisData.data)
```

Plotting the cluster in 2d format

```
malplt.figure(figsize=(10, 8))
colors = ['r', 'g', 'b']
for cluster in range(kValue):
    cluster_data = iris_df[iris_df['Cluster'] == cluster]
    malplt.scatter(cluster_data['PC1'], cluster_data['PC2'], c=colors[cluster], label=f'Cluster {cluster}')

malplt.title('K-Means Clustering of Iris Dataset (2D Projection)')
malplt.xlabel('Principal Component 1 (PC1)')
malplt.ylabel('Principal Component 2 (PC2)')
malplt.legend()
malplt.show()
```


K-Means Clustering of Iris Dataset (2D Projection)
Principal Component 2 (PC2)
● Cluster 0
● Cluster 1
● Cluster 2
Principal Component 1 (PC1)