**NAME: Mohammad Hussam (2303.KHI.DEG.020)**

**PAIRING** WITH **: MAVIA ALAM KHAN (2303.KHI.DEG.017)**

**&**

**AQSA TAUHEED(2303.KHI.DEG.011)**

---

# ASSIGNMENT NO : 3.3

Perform k-means clusterization on the Iris dataset. Repeat the procedure on the dataset reduced with PCA, and then compare the results.

## Solution:

### Step#1:

First of all we have to import the necessary libraries.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
```

### Step#2:

After that we load the iris data set.

```
iris = load_iris()
X = iris.data
y = iris.target
```

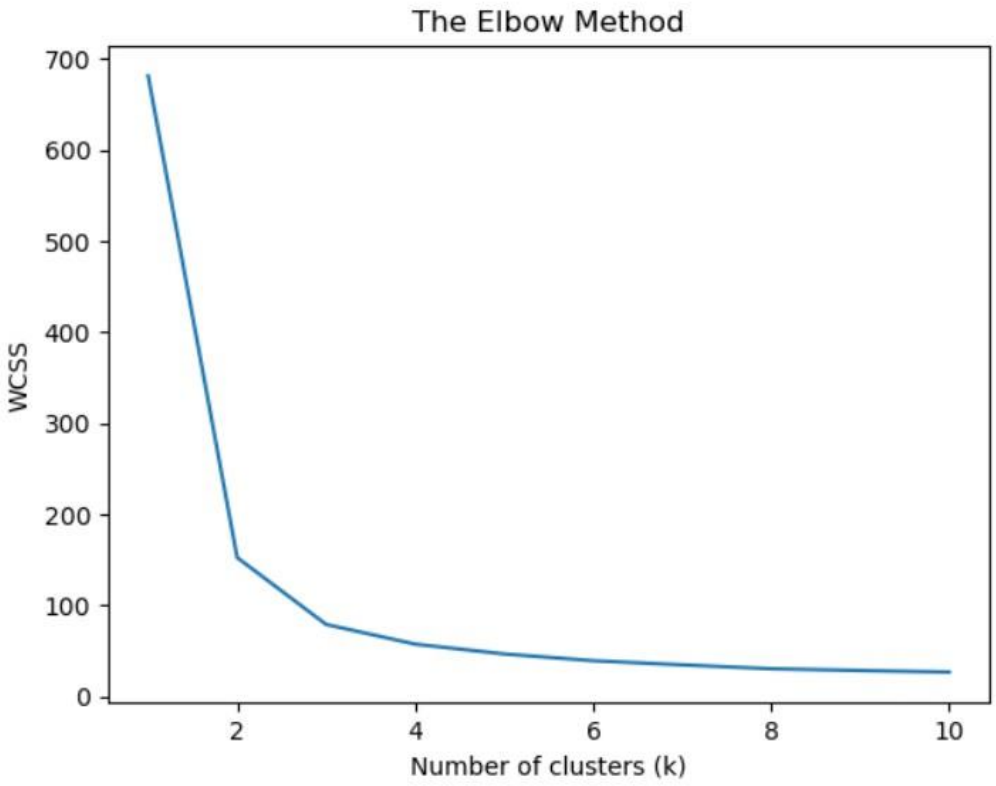
## Step#3:

We use elbow method to find optimal number of cluster in the original data set

```
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++', n_init=10, random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

```
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters (k)')
plt.ylabel('WCSS')
plt.show()
```

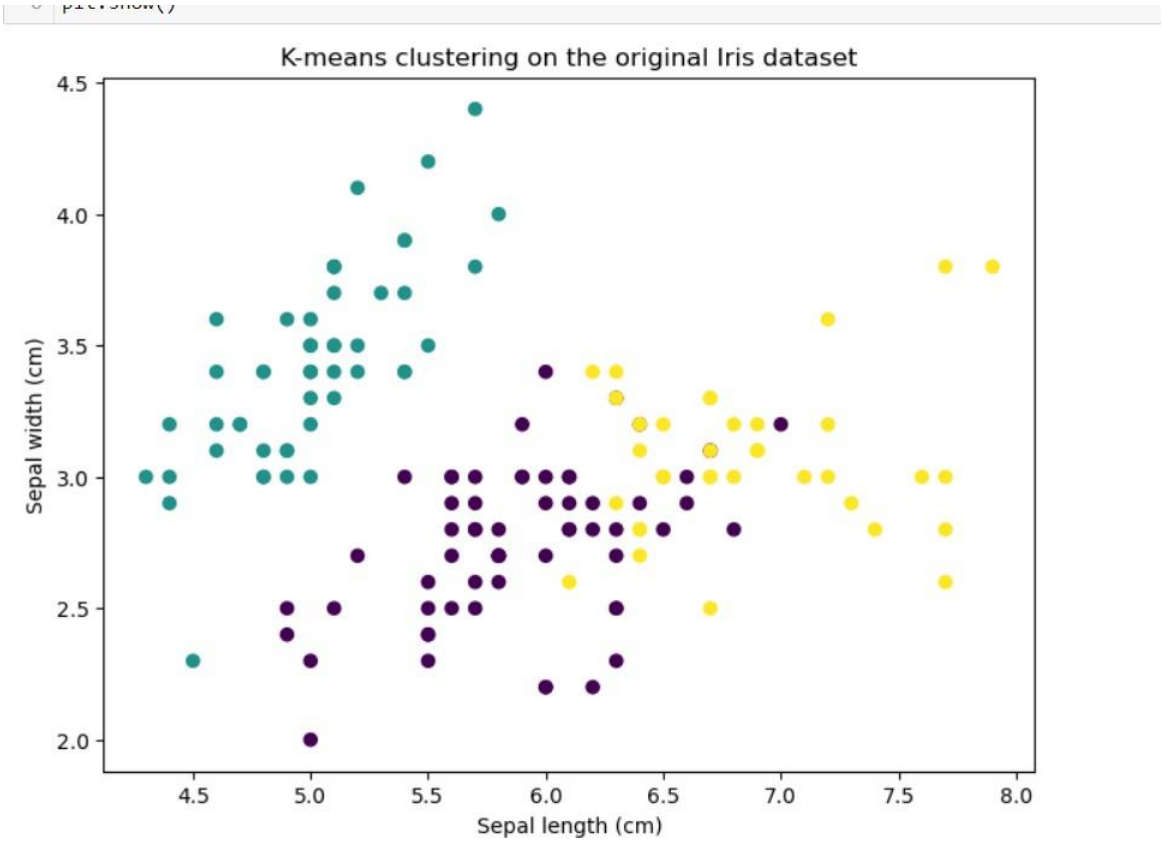
```
plt.show()
```




now in the above graph we see that the optimal number is 3 .so we provided, the n_clusters parameter is set to 3, which means we are performing K-means clustering with 3 clusters. This means that the algorithm will try to group the data points into three distinct clusters based on their similarity.

```
kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
y_pred_original = kmeans.fit_predict(X)
```

```
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=y_pred_original, cmap='viridis')
plt.title('K-means clustering on the original Iris dataset')
plt.xlabel('Sepal length (cm)')
plt.ylabel('Sepal width (cm)')
plt.show()
```




## Step # 4:

Applying PCA to reduce the dimensionality of the dataset .

Now first we use cumulative explained variance to find out the n number of components and then we reduced it .

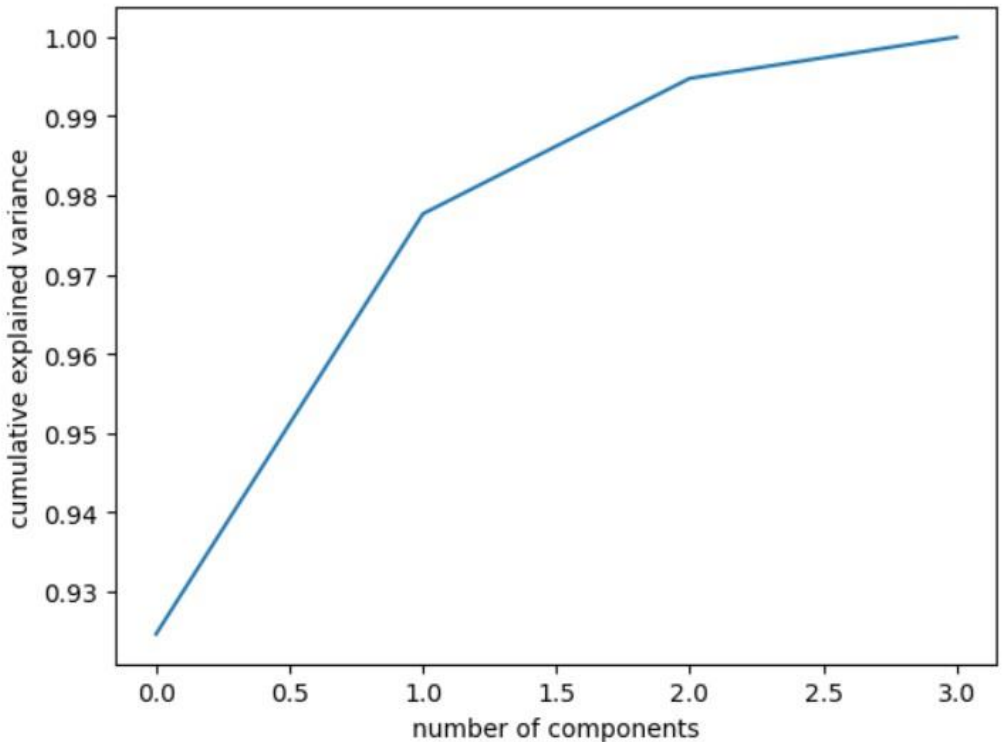
We reduce our data because beneficial when dealing with high-dimensional data, as it helps to simplify the dataset and reduce the computational complexity of subsequent analysis or modeling tasks.

```
]:  1 pca = PCA()
    2 pca.fit(X)
```

```
]: PCA()
```

8]:
```
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_variance_ratio = np.cumsum(explained_variance_ratio)
```

9]:
```
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance');
```



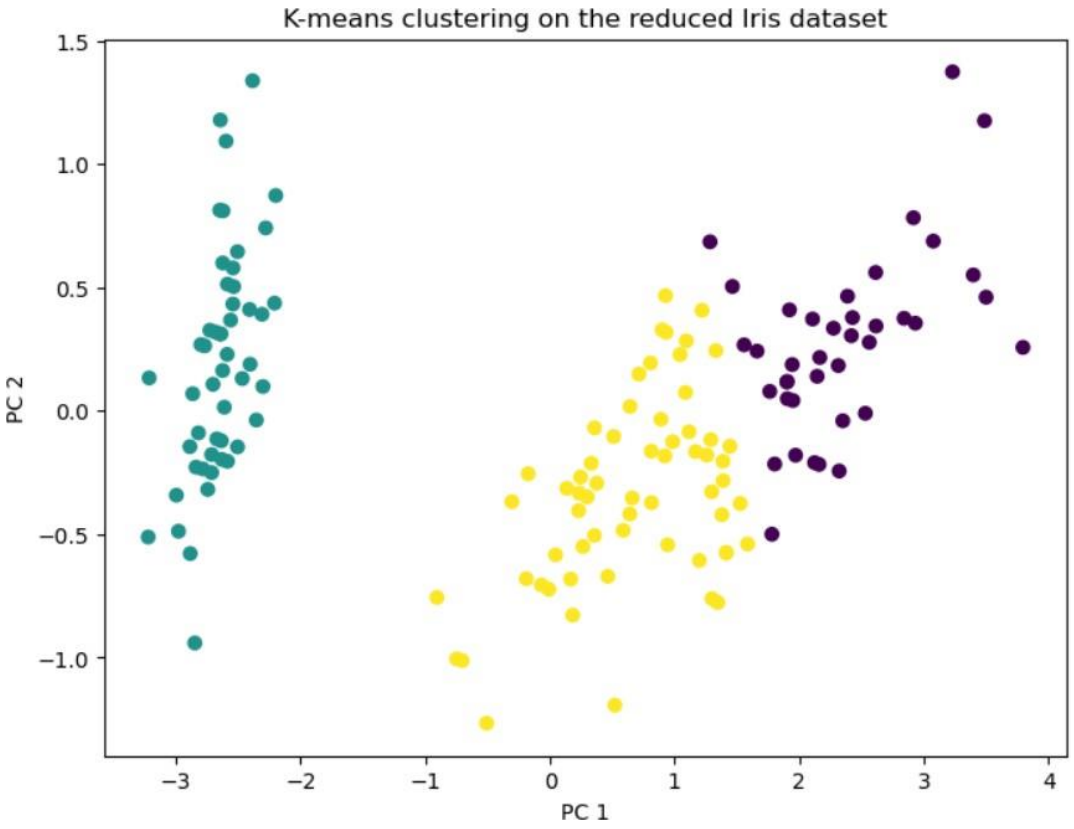

So in the above graph we see that number of components that is 2 It determines the number ofcomponents or dimensions that reduce our data. n_component=2 means you are reducing the dimensionality to 2 components.

```
pca = PCA(n_components=2)
X_reduced = pca.fit_transform(X)
```

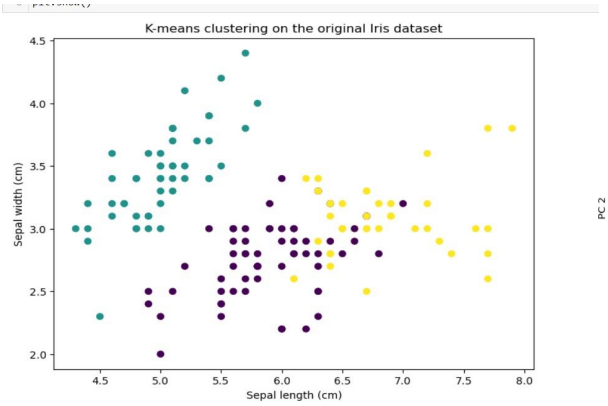
```
kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
y_pred_reduced = kmeans.fit_predict(X_reduced)
```

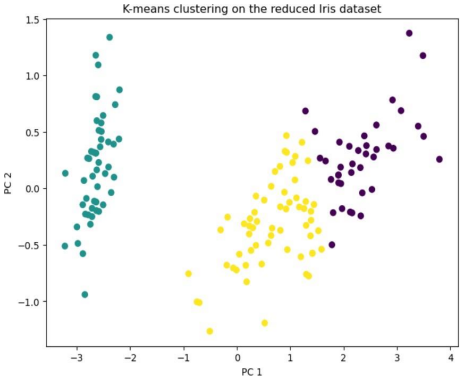
```
plt.figure(figsize=(8, 6))
plt.scatter(X_reduced[:, 0], X_reduced[:, 1], c=y_pred_reduced, cmap='viridis')
plt.title('K-means clustering on the reduced Iris dataset')
plt.xlabel('PC 1')
plt.ylabel('PC 2')
plt.show()
```

K-means clustering on the reduced Iris dataset




# COMPARISION:



Before PCA, the iris dataset had four features and k-means clustering was performed based on these features. After performing PCA, the dataset was transformed into two principal components, which capture most of the variance in the data. The k-means clustering algorithm was then applied to both the original and reduced datasets. The clustering results were similar in terms of the groupings of the flowers, but the reduced dataset plot was more compact and the separation between the clusters was more pronounced, indicating that the two principal components obtained from PCA are able to summarize the information in the original features effectively.