# **Practical – 7**

**Aim:** **Write a program for Automatic grouping of similar objects into sets.**

* Code:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

data\_set = pd.read\_csv('/content/drive/MyDrive/temp/practical\_4\_2.csv')

X = data\_set[['Match\_Duration', 'Loot\_Collected', 'Enemies\_Defeated']]

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

k = 3

kmeans = KMeans(n\_clusters=k, random\_state=42)

clusters = kmeans.fit\_predict(X\_scaled)

centroids = kmeans.cluster\_centers\_

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X\_scaled)

centroids\_pca = pca.transform(centroids)

data\_set['Cluster'] = clusters

colors = ['red', 'green', 'blue']

labels = [f'Cluster {i}' for i in range(k)]

plt.figure(figsize=(10, 7))

for i in range(k):

plt.scatter(X\_pca[clusters == i, 0], X\_pca[clusters == i, 1],

c=colors[i], label=labels[i])

plt.scatter(centroids\_pca[:, 0], centroids\_pca[:, 1],

s=200, c='yellow', marker='X', label='Centroids')

new\_point = np.array([[26, 77, 15]])

new\_point\_scaled = scaler.transform(new\_point)

new\_point\_cluster = kmeans.predict(new\_point\_scaled)[0]

new\_point\_pca = pca.transform(new\_point\_scaled)

plt.scatter(new\_point\_pca[0, 0], new\_point\_pca[0, 1],

c='black', s=150, marker='\*', label=f'Data point {new\_point[0]} is in Cluster {new\_point\_cluster}')

plt.title("K-Means Clustering with PCA (Grouped)")

plt.xlabel("PCA Feature 1")

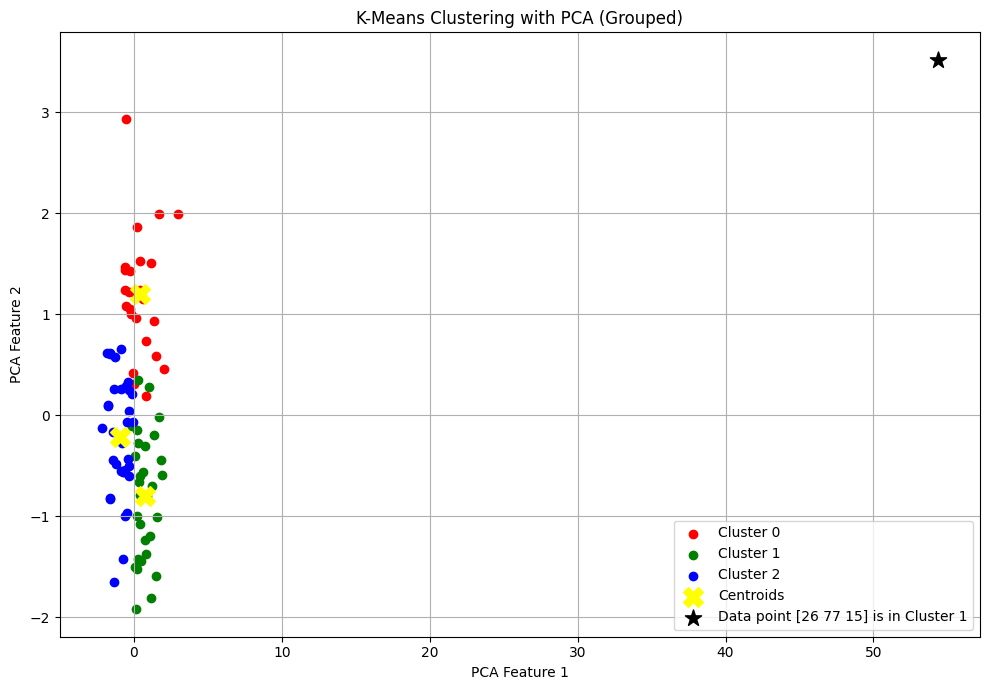
plt.ylabel("PCA Feature 2")

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

* Output