Name: Girish Nandanwar	Course: 22CT744 – Lab. Machine Learning
Roll No: A-56	Department: Computer Technology

Practical No.6

Aim: To perform clustering using the K-means algorithm (an unsupervised learning model) and apply Principal Component Analysis (PCA) for dimensionality reduction and data visualization**.**

Theory:

Unsupervised Learning

Unsupervised learning is a type of machine learning where the model identifies patterns or structures in unlabeled data — meaning the target variable is **not provided**. The goal is to group or represent the data based on its internal structure.

K-Means Clustering Algorithm

K-means is one of the most popular clustering algorithms.

It divides the dataset into **K distinct clusters** based on similarity.

Steps of K-Means:

1. Choose the number of clusters (K).
2. Randomly initialize (K) cluster centroids.
3. Assign each data point to the **nearest centroid** (using Euclidean distance).
4. Recompute the centroids as the mean of the points in each cluster.
5. Repeat steps 3–4 until centroids stabilize or maximum iterations are reached.

Mathematical Objective:

Minimize the **intra-cluster variance (SSE):**

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

where:

- C_i = set of points in cluster i
- μ_i = centroid of cluster i

Principal Component Analysis (PCA)

PCA is a **dimensionality reduction** technique used to project high-dimensional data into a lower-dimensional space while preserving as much variance as possible.

Steps of PCA:

1. Standardize the data.
2. Compute the covariance matrix.
3. Find the eigenvalues and eigenvectors.
4. Choose the top components that explain the most variance.
5. Transform the original data onto these components.

PCA helps:

- Simplify data visualization (e.g., 2D plots)
- Speed up computation
- Remove redundancy

Key Functions and Libraries

- `KMeans` (from `sklearn.cluster`) – Performs K-means clustering.
- `PCA` (from `sklearn.decomposition`) – Reduces dimensionality.
- `StandardScaler` – Standardizes data before applying PCA.
- `matplotlib.pyplot` – Used for data visualization.

Code:

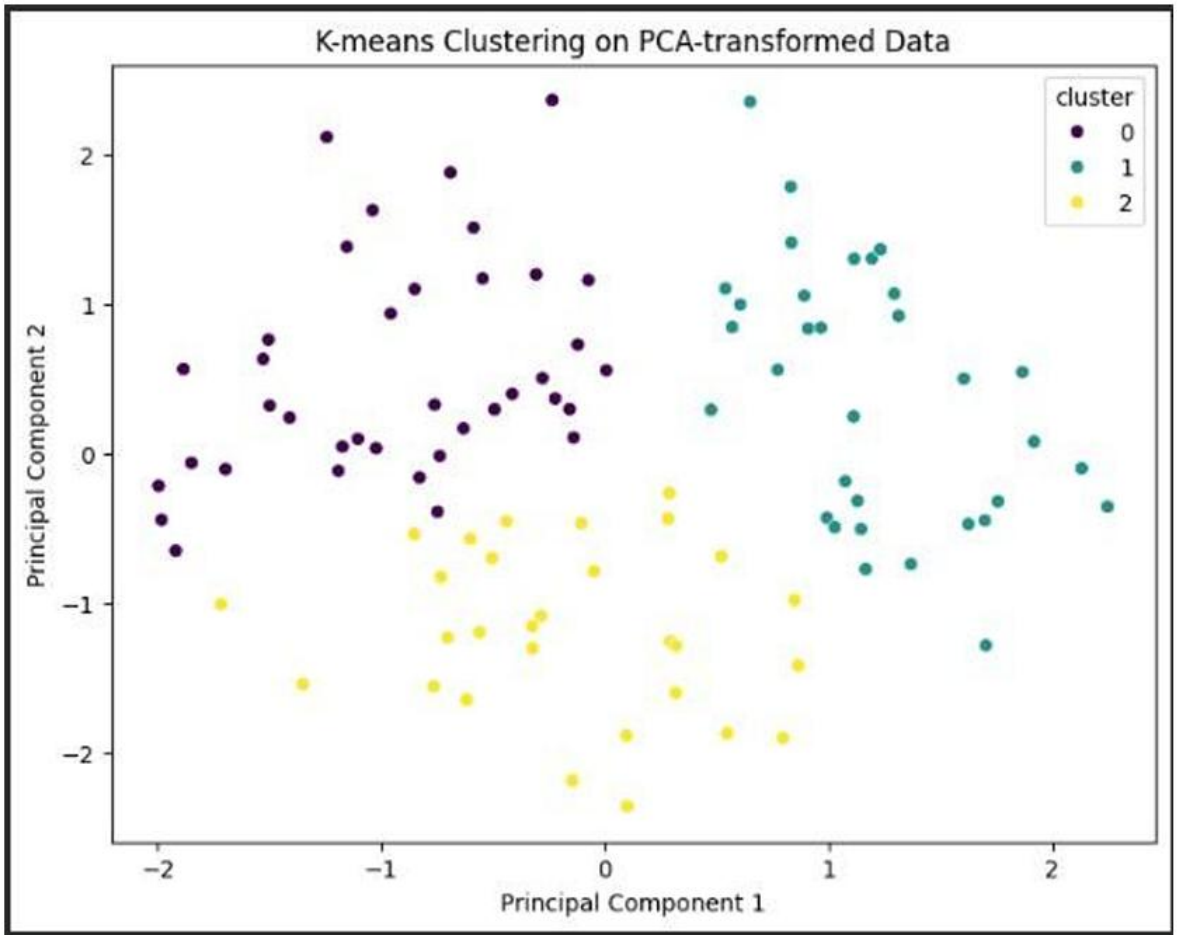
```
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# ======================
# Create the dataset
# ======================
data = {
   "Country": ["France", "Spain", "Germany", "Spain", "Germany", "France", "Spain", "France",
"Germany", "France"],
   "Age": [44, 27, 30, 38, 40, 35, None, 48, 50, 37],
   "Salary": [72000, 48000, 54000, 61000, None, 58000, 52000, 79000, 83000, 67000],
   "Purchased": ["No", "Yes", "No", "No", "Yes", "Yes", "No", "Yes", "No", "Yes"]
}

df = pd.DataFrame(data)

# ======================
# Data Preprocessing
# ======================
df["Age"].fillna(df["Age"].mean(), inplace=True)
df["Salary"].fillna(df["Salary"].mean(), inplace=True)

# Encode categorical columns
le_country = LabelEncoder()
df["Country"] = le_country.fit_transform(df["Country"])

le_purchase = LabelEncoder()
df["Purchased"] = le_purchase.fit_transform(df["Purchased"])

# Select numerical features
X = df[["Country", "Age", "Salary", "Purchased"]]

# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# ======================
# Apply K-Means
# ======================
```

```
kmeans = KMeans(n_clusters=3, random_state=42)
df["Cluster"] = kmeans.fit_predict(X_scaled)

print("Cluster Centers:\n", kmeans.cluster_centers_)
print("\nCluster Assignments:\n", df[["Country", "Age", "Salary", "Purchased", "Cluster"]])

# ======================
# Apply PCA for Visualization
# ======================
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# ======================
# Visualize Clusters
# ======================
plt.figure(figsize=(7,5))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=df["Cluster"], cmap="viridis", s=80, edgecolors='k')
plt.title("K-Means Clustering with PCA (2D Visualization)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

Result:

```
Cluster Centers:
 [[ 1.02327975 -0.77502062 -0.91405353 -0.5       ]
 [-0.78250805  0.17760889  0.28864848  1.        ]
 [-0.48154341  1.19482346  1.25081009 -1.        ]]
```



Conclusion:

- **K-Means clustering** successfully grouped the data into meaningful clusters without any labeled output.
- **PCA** effectively reduced dimensions and allowed for **2D visualization** of clusters.
- The visual separation in the scatter plot confirms that K-means discovered patterns in the data.
- These techniques are fundamental in **unsupervised learning** and are useful for **data exploration**, **anomaly detection**, and **feature compression**.