**NAME : MANGESH A. GHADWAJE**

**ROLL NO:24**

**BATCH : B2**

**COURSE: ML PRACTICAL**

**Assginment No. 2**

**Problem Statement :**

**Implement Principal Component Analysis (PCA) using python.**

**Code :**

***import numpy as np***

***import pandas as pd***

***import matplotlib.pyplot as plt***

***from sklearn.datasets import load\_iris***

***from sklearn.preprocessing import StandardScaler***

***from sklearn.decomposition import PCA***

***from sklearn import metrics***

***'''***

***Imports necessary libraries:***

***numpy and pandas are used for data manipulation.***

***matplotlib.pyplot is used for plotting.***

***load\_iris from sklearn.datasets loads the Iris dataset.***

***StandardScaler standardizes features by removing the mean and scaling to unit variance.***

***PCA from sklearn.decomposition performs Principal Component Analysis.***

***metrics is imported but not used in this snippet.***

***'''***

***# Load the Iris dataset***

***data = load\_iris()  #load\_iris() loads the dataset and returns a dictionary-like object.***

***X = data.data  #X contains the feature data (e.g., sepal length, sepal width).***

***y = data.target  #y contains the target labels (species of iris).***

***target\_names = data.target\_names   #target\_names holds the names of the iris species.***

***# Standardize the data***

***scaler = StandardScaler()   #StandardScaler() creates an instance of the scaler.***

***X\_standardized = scaler.fit\_transform(X)  #fit\_transform(X) standardizes X by removing the mean and scaling to unit variance, resulting in X\_standardized.***

***# Initialize PCA***

***pca = PCA() #Creates an instance of PCA without specifying the number of components.***

***# Fit PCA on the standardized data***

***pca.fit(X\_standardized)  #Computes the principal components and the explained variance from the standardized data.***

***# Calculate the cumulative explained variance ratio***

***explained\_variance\_ratio = pca.explained\_variance\_ratio\_  #explained\_variance\_ratio\_ contains the proportion of variance explained by each principal component.***

***cumulative\_explained\_variance\_ratio = np.cumsum(explained\_variance\_ratio)  #np.cumsum(explained\_variance\_ratio) computes the cumulative sum of these ratios to understand how much variance is explained by the first n components.***

***# Determine the number of components needed to retain at least 95% variance***

***n\_components = np.argmax(cumulative\_explained\_variance\_ratio >= 0.95) + 1***

***#np.argmax(cumulative\_explained\_variance\_ratio >= 0.95) finds the index of the first component where the cumulative explained variance reaches or exceeds 95%.***

***#+1 adjusts the index to account for zero-based indexing, giving the number of components needed.***

***# Transform the data to the new feature space***

***pca = PCA(n\_components=n\_components) #PCA(n\_components=n\_components) creates a PCA instance with the calculated number of components.***

***X\_pca = pca.fit\_transform(X\_standardized) #fit\_transform(X\_standardized) projects the data onto the selected principal components, resulting in X\_pca.***

***# Output the results***

***print(f'Number of components selected: {n\_components}')***

***print(f'Explained variance ratio of each component: {explained\_variance\_ratio}')***

***print(f'Cumulative explained variance ratio: {cumulative\_explained\_variance\_ratio}')***

***# Plot the results***

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

***colors = ['navy', 'turquoise', 'darkorange']***

***for color, i, target\_name in zip(colors, [0, 1, 2], target\_names):***

***plt.scatter(X\_pca[y == i, 0], X\_pca[y == i, 1], color=color, label=target\_name)***

***plt.title(f'PCA of Iris Dataset\nComponents: {n\_components}')***

***plt.xlabel('Principal Component 1')***

***plt.ylabel('Principal Component 2')***

***plt.legend(loc='best')***

***plt.grid()***

***plt.show()***

***'''***

***plt.figure(figsize=(10, 7)) creates a figure with a specified size.***

***colors defines a list of colors for different iris species.***

***The for loop creates scatter plots for each class in different colors.***

***X\_pca[y == i, 0] and X\_pca[y == i, 1] select the first and second principal components for each class.***

***color and label differentiate the classes in the plot.***

***plt.title() sets the title of the plot.***

***plt.xlabel() and plt.ylabel() label the axes.***

***plt.legend() adds a legend to the plot.***

***plt.grid() enables the grid.***

***plt.show() displays the plot.***

***'''***

**Output :**

Number of components selected: 2

Explained variance ratio of each component: [0.72962445 0.22850762 0.03668922 0.00517871]

Cumulative explained variance ratio: [0.72962445 0.95813207 0.99482129 1. ]

