**NAME : MANGESH A. GHADWAJE**

**ROLL NO:24**

**BATCH : B2**

**COURSE: ML PRACTICAL**

**Assginment No. 2**

**Problem Statement :**

**Implement Principal Component Analysis (PCA) on a given dataset Iris to reduce its dimensionality while retaining at least 95% of the variance.**

**Code :**

***import pandas as pd***

***from sklearn.preprocessing import StandardScaler, LabelEncoder***

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

***from sklearn.impute import SimpleImputer***

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

***df = pd.read\_csv('iris\_dataset.csv', na\_values=['??', '', '/', '###', '$', '&'])***

***X = df.iloc[:, :-1]***

***y = df.iloc[:, -1] # Target (last column)***

***print(df)***

***print(X.isnull().sum())***

***imputer = SimpleImputer(strategy='mean')***

***X\_imputed = imputer.fit\_transform(X)***

***X\_imputed\_df = pd.DataFrame(X\_imputed, columns=X.columns)***

***print("Missing values after filling:")***

***print(X\_imputed\_df.isnull().sum())***

***print(X\_imputed\_df)***

***label\_encoder = LabelEncoder()***

***y\_encoded = label\_encoder.fit\_transform(y)***

***scaler = StandardScaler()***

***X\_scaled = scaler.fit\_transform(X\_imputed\_df)***

***pca = PCA(0.95)***

***principal\_components = pca.fit\_transform(X\_scaled)***

***plt.figure(figsize=(8, 6))***

***scatter = plt.scatter(principal\_components[:, 0], principal\_components[:, 1], c=y\_encoded, cmap='viridis')***

***plt.title('PCA of IRIS Data')***

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

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

***plt.show()***

**Output :**

