Name : Tavhare Ruchita Sharad

class : BE AI&DS

Roll No.: 61

Subject: Computer Laboratory-I (ML)

Title:To use PCA Algorithm for dimensionality reduction. You have a dataset that includes measurements for different variables on wine (alcohol, ash, magnesium, and so on). Apply PCA algorithm & transform this data so that most variations in the measurements of the variables are captured by a small number of principal components so that it is easier to distinguish between red and white wine by inspecting these principal components.

In [3]:

```python
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

In [5]:

```python
df = pd.read_csv("Wine.csv")
```

In [13]:

```python
df.head(5)
```

Out[13]:

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	

In [17]:

```python
df.Wine.unique()
```

Out[17]:

```
array([1, 2, 3], dtype=int64)
```

In [19]:

```python
print(df.isnull().sum())       #checking is null
```

```
Wine                    0
Alcohol                 0
Malic.acid              0
Ash                     0
Acl                     0
Mg                      0
Phenols                 0
Flavanoids              0
Nonflavanoid.phenols    0
Proanth                 0
Color.int               0
Hue                     0
OD                      0
Proline                 0
dtype: int64
```

In [23]:

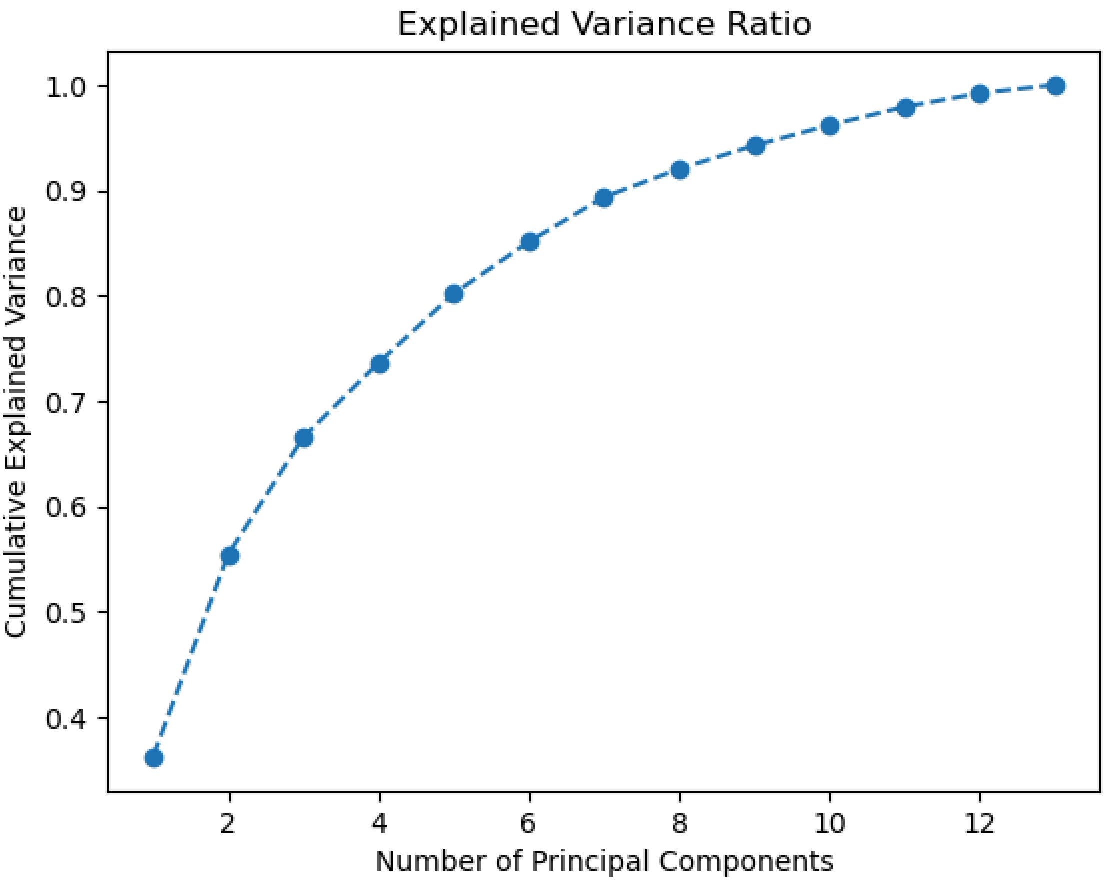
```
X = df.drop('Wine', axis=1)  # Features
y = df['Wine']  # Target variable
for col in X.columns:
    sc = StandardScaler()                                  #Standardize fea
    X[col] = sc.fit_transform(X[[col]])                     #Fit to data, t
X.head(5)
```

Out[23]:

	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nor
0	1.518613	-0.562250	0.232053	-1.169593	1.913905	0.808997	1.034819	
1	0.246290	-0.499413	-0.827996	-2.490847	0.018145	0.568648	0.733629	
2	0.196879	0.021231	1.109334	-0.268738	0.088358	0.808997	1.215533	
3	1.691550	-0.346811	0.487926	-0.809251	0.930918	2.491446	1.466525	
4	0.295700	0.227694	1.840403	0.451946	1.281985	0.808997	0.663351	

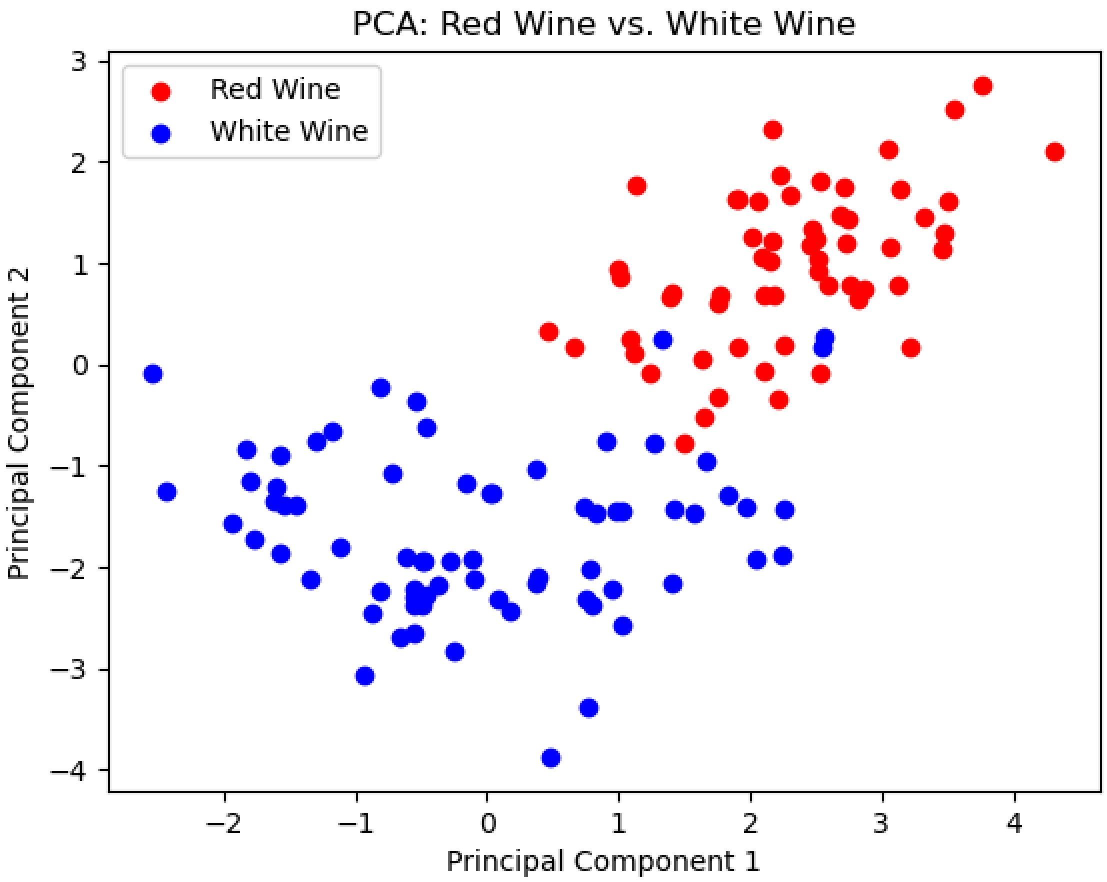
In [28]:

```
pca = PCA()
X_pca = pca.fit_transform(X)
explained_variance_ratio = pca.explained_variance_ratio_
plt.plot(range(1, len(explained_variance_ratio) + 1), explained_variance_ratic
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('Explained Variance Ratio')
plt.show()
```



In [38]:

```
n_components= 12  # Choose the desired number of principal components you want
pca = PCA(n_components=n_components)
X_pca = pca.fit_transform(X)
X_pca.shape
X.shape
red_indices = y[y == 1].index
white_indices = y[y == 2].index
plt.scatter(X_pca[red_indices, 0], X_pca[red_indices, 1], c='red', label='Red
plt.scatter(X_pca[white_indices, 0], X_pca[white_indices, 1], c='blue', label=
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
plt.title('PCA: Red Wine vs. White Wine')
plt.show()
```


PCA: Red Wine vs. White Wine
● Red Wine
● White Wine
3
2
1
0
−1
−2
−3
−4
−2
−1
0
1
2
3
4
Principal Component 1
Principal Component 2

```
In [ ]:
```