In [6]:
```
'''
NAME: MANE SHIVRAJ PANDURANG
ROLL NO.37
COURSE: AI&DS, SUB:ML(Machine Learning)
CLASS: BE
'''
```

Out[6]:
```
'\nNAME: MANE SHIVRAJ PANDURANG\nROLL NO.37\nCOURSE: AI&DS, SUB:ML(Machine Learning)\nCLASS: BE \n'
```

In [7]:
```
'''
PRACTICAL NO-01:
    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.'''
```

Out[7]:
```
'\nPRACTICAL NO-01:\n    To use PCA Algorithm for dimensionality reduction. You have a dataset that includes\nmeas
urements for different variables on wine (alcohol, ash, magnesium, and so on).\nApply PCA algorithm & transform th
is data so that most variations in the measurements\nof the variables are captured by a small number of principal
components so that it is\neasier to distinguish between red and white wine by inspecting these principal\ncomponen
ts.'
```

In [8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
```

In [9]:
```
data=pd.read_csv('Wine.csv')
```

In [10]:
```
data.head()
```

Out[10]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	C
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	

In [11]:
```
data.tail()
```

Out[11]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.41
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.35
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.46
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.35

In [12]:
```
data.shape
```

Out[12]:
```
(178, 14)
```

In [13]:
```
data.describe()
```

Out[13]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proan
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	

In [14]:

```
data['Customer_Segment'].unique()
```

Out[14]:

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

In [15]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Alcohol               178 non-null    float64
 1   Malic_Acid            178 non-null    float64
 2   Ash                   178 non-null    float64
 3   Ash_Alcanity          178 non-null    float64
 4   Magnesium             178 non-null    int64  
 5   Total_Phenols         178 non-null    float64
 6   Flavanoids            178 non-null    float64
 7   Nonflavanoid_Phenols  178 non-null    float64
 8   Proanthocyanins       178 non-null    float64
 9   Color_Intensity       178 non-null    float64
 10  Hue                   178 non-null    float64
 11  OD280                 178 non-null    float64
 12  Proline               178 non-null    int64  
 13  Customer_Segment      178 non-null    int64  
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

In [16]:

```
data.isnull().sum()
```

Out[16]:

```
Alcohol                 0
Malic_Acid              0
Ash                     0
Ash_Alcanity            0
Magnesium               0
Total_Phenols           0
Flavanoids              0
Nonflavanoid_Phenols    0
Proanthocyanins         0
Color_Intensity         0
Hue                     0
OD280                   0
Proline                 0
Customer_Segment        0
dtype: int64
```

No null values

In [17]:

```
x=data.drop('Customer_Segment',axis=1)
y=data['Customer_Segment']
```

In [18]:

```
x
```

Out[18]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82
...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.41
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.35
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.46
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.35

178 rows × 13 columns

In [19]:
```
x.shape
```

Out[19]:
```
(178, 13)
```

In [20]:
```
x_standardized = (x - x.mean()) / x.std()
```

In [21]:
```
y
```

Out[21]:
```
0      1
1      1
2      1
3      1
4      1
      ..
173    3
174    3
175    3
176    3
177    3
Name: Customer_Segment, Length: 178, dtype: int64
```

In [22]:
```
pca=PCA(n_components=3)
```

In [23]:
```
x_pca=pca.fit_transform(x)
```

In [24]:
```
x_pca.shape
```

Out[24]:
```
(178, 3)
```

13 columns got reduced to 3 columns

In [25]:
```
pca_df = pd.DataFrame(x_pca, columns = ['pca_col1','pca_col2','pca_col3'])
```

In [26]:
```
pca_df
```

Out[26]:

	pca_col1	pca_col2	pca_col3
0	318.562979	21.492131	3.130735
1	303.097420	-5.364718	6.822835
2	438.061133	-6.537309	-1.113223
3	733.240139	0.192729	-0.917257
4	-11.571428	18.489995	-0.554422
...	...	...	...
173	-6.980211	-4.541137	-2.474707
174	3.131605	2.335191	-4.309931
175	88.458074	18.776285	-2.237577
176	93.456242	18.670819	-1.788392
177	-186.943190	-0.213331	-5.630510

178 rows × 3 columns

In [28]:

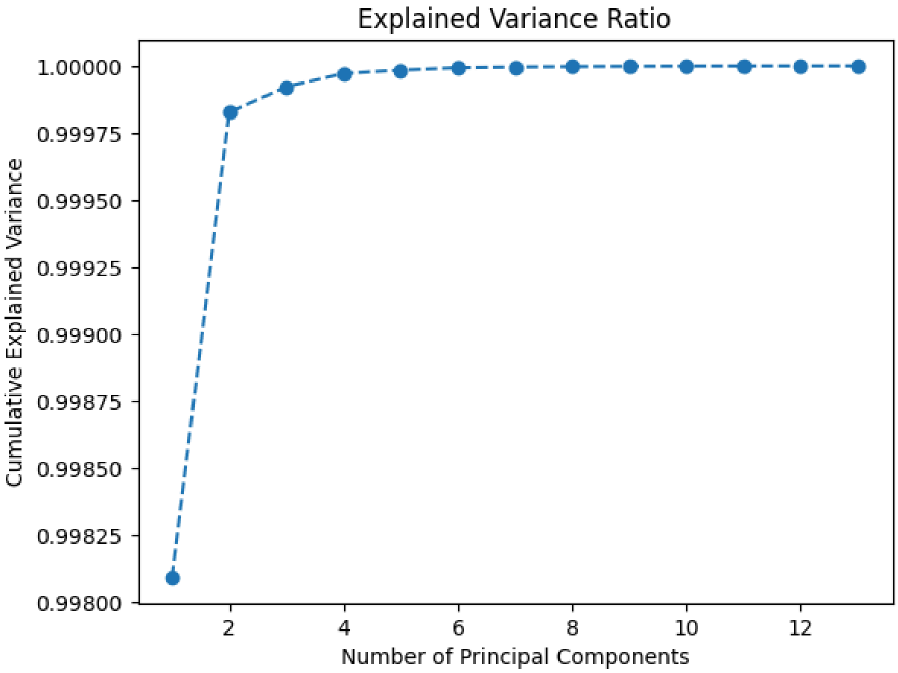
```
pca.explained_variance_ratio_
```

Out[28]:

```
array([9.98091230e-01, 1.73591562e-03, 9.49589576e-05])
```

In [30]:

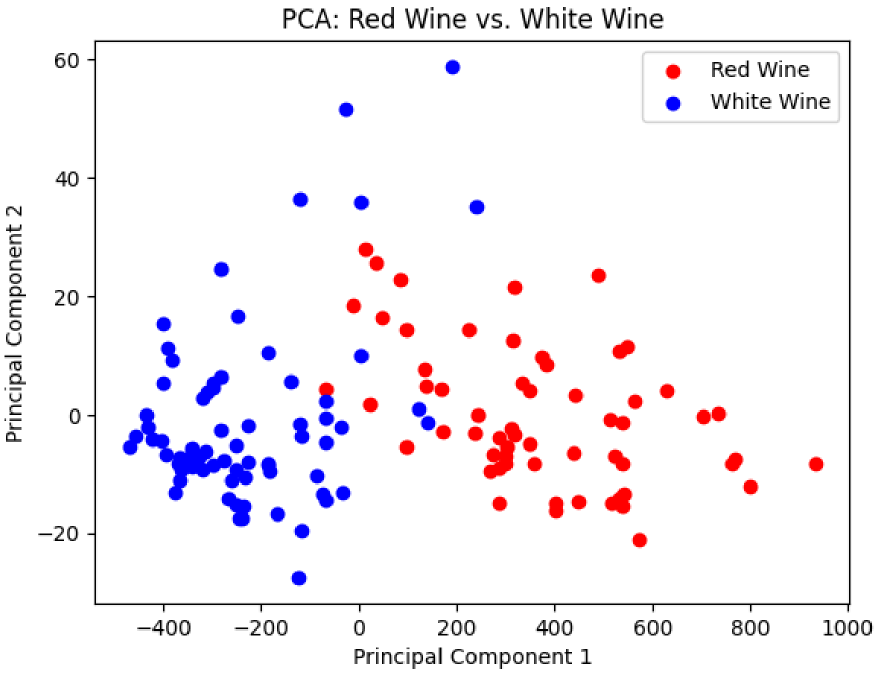
```
pca = PCA()
X_pca = pca.fit_transform(x)
explained_variance_ratio = pca.explained_variance_ratio_
plt.plot(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio.cumsum(), marker='o',
linestyle='--')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('Explained Variance Ratio')
plt.show()
```



In [31]:

```
n_components = 12 # Choose the desired number of principal components you want to reduce a dimention to
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 Wine')
plt.scatter(X_pca[white_indices, 0], X_pca[white_indices, 1], c='blue', label='White Wine')
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
Principal Component 2
Principal Component 1