APSSDC

Andhra Pradesh State Skill Development Corporation

(https://apssdc.in)

Dimensionality Reduction

Principal Component Analysis (PCA)

Brief primer and history

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation (https://en.wikipedia.org/wiki/Orthogonal_transformation) to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated (https://en.wikipedia.org/wiki/Correlation_and_dependence) variables called principal components. The number of distinct principal components is equal to the smaller of the number of original variables or the number of observations minus one. This transformation is defined in such a way that the first principal component has the largest possible variance (https://en.wikipedia.org/wiki/Variance) (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal (https://en.wikipedia.org/wiki/Orthogonal) the preceding components. The resulting vectors are an uncorrelated orthogonal basis set (https://en.wikipedia.org/wiki/Orthogonal_basis_set).

PCA is sensitive to the relative scaling of the original variables.

PCA was invented in 1901 by Karl Pearson (https://en.wikipedia.org/wiki/Karl_Pearson) as an analogue of the principal axis theorem in mechanics; it was later independently developed and named by Harold Hotelling (https://en.wikipedia.org/wiki/Harold_Hotelling) in the 1930s.

Dataset_Link (https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation/Datasets/master/wine.data.csv)

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df = pd.read_csv('https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation/
```

In [3]:

```
df.head()
```

Out[3]:

	Class	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proa
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	

In [4]:

```
df.columns
```

Out[4]:

```
Index(['Class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash',
       'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
       'Proanthocyanins', 'Color intensity', 'Hue',
       'OD280/OD315 of diluted wines', 'Proline'],
      dtype='object')
```

In [6]:

```
df.shape
```

Out[6]:

```
(178, 14)
```

1. Alcohol - The type of wine, into one of three classes, 1 (59 obs), 2(71 obs), and 3 (48 obs)
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
Class                           0
Alcohol                         0
Malic acid                      0
Ash                             0
Alcalinity of ash               0
Magnesium                       0
Total phenols                   0
Flavanoids                      0
Nonflavanoid phenols            0
Proanthocyanins                 0
Color intensity                 0
Hue                             0
OD280/OD315 of diluted wines    0
Proline                         0
dtype: int64
```

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Class                         178 non-null    int64  
 1   Alcohol                       178 non-null    float64
 2   Malic acid                    178 non-null    float64
 3   Ash                           178 non-null    float64
 4   Alcalinity of ash             178 non-null    float64
 5   Magnesium                     178 non-null    int64  
 6   Total phenols                 178 non-null    float64
 7   Flavanoids                    178 non-null    float64
 8   Nonflavanoid phenols          178 non-null    float64
 9   Proanthocyanins               178 non-null    float64
 10  Color intensity               178 non-null    float64
 11  Hue                           178 non-null    float64
 12  OD280/OD315 of diluted wines  178 non-null    float64
 13  Proline                       178 non-null    int64  
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

In [8]:

```
df.duplicated().sum()
```

Out[8]:

```
0
```

In [11]:

```
df.mean()
```

Out[11]:

```
Class                           1.938202
Alcohol                        13.000618
Malic acid                      2.336348
Ash                             2.366517
Alcalinity of ash              19.494944
Magnesium                      99.741573
Total phenols                   2.295112
Flavanoids                      2.029270
Nonflavanoid phenols            0.361854
Proanthocyanins                 1.590899
Color intensity                 5.058090
Hue                             0.957449
OD280/OD315 of diluted wines    2.611685
Proline                       746.893258
dtype: float64
```

In [9]:

```
from sklearn.preprocessing import StandardScaler
```

In [20]:

```
x = df.drop('Class', axis = 1)
y = df['Class']

ss = StandardScaler()

scaData = ss.fit_transform(x)
```

In [16]:

```
scaData[:,0].std()
```

Out[16]:

```
0.9999999999999997
```

In [17]:

```
scaData[:,0].mean()
```

Out[17]:

```
1.5967252488991015e-16
```

In [18]:

```
from sklearn.decomposition import PCA
```

In [22]:

```
model = PCA()
model.fit(scaData)
```

Out[22]:

```
PCA()
```

In [23]:

```
model.explained_variance_ratio_
```

Out[23]:

```
array([0.36198848, 0.1920749 , 0.11123631, 0.0706903 , 0.06563294,
       0.04935823, 0.04238679, 0.02680749, 0.02222153, 0.01930019,
       0.01736836, 0.01298233, 0.00795215])
```

In [24]:

```
len(model.explained_variance_ratio_)
```

Out[24]:

```
13
```

In [26]:

```
model.n_components_
```

Out[26]:

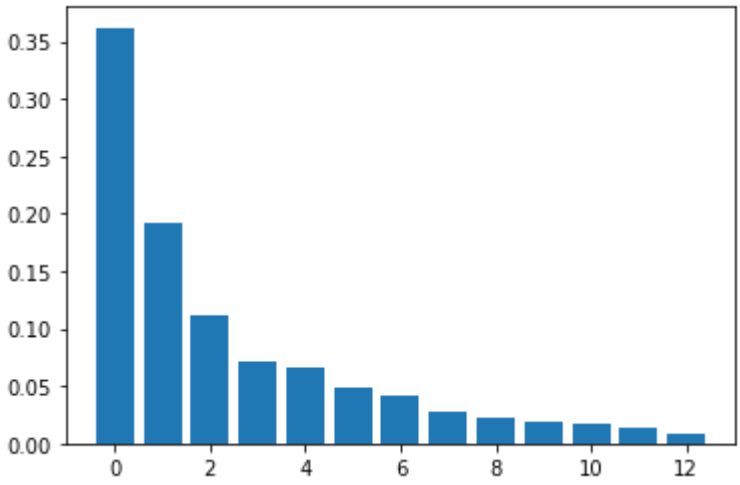
```
13
```

In [28]:

```
plt.bar(range(model.n_components_), model.explained_variance_ratio_)
```

Out[28]:

```
<BarContainer object of 13 artists>
```



In [38]:

```
plt.figure(figsize=(10,6))


plt.scatter(scaData[:,0], scaData[:, 1], c = df['Class'], edgecolors='k',alpha=0.75,s=150)
plt.grid(True)
plt.title("Class separation using first two principal components\n",fontsize=20)
plt.xlabel("Principal component-1",fontsize=15)
plt.ylabel("Principal component-2",fontsize=15)
plt.show()
```