CO544-Lab4-Ex

June 13, 2020

[78]:
```python
#Importing Matplotlib
import matplotlib
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d #importing modules for 3D plotting
```

[79]:
```python
from sklearn import datasets #import standard data sets
wine_dataset =datasets.load_wine()
```

[80]:
```python
from sklearn.decomposition import PCA
```

[81]:
```python
wine_data = wine_dataset["data"] #defining features values
wine_labels = wine_dataset["target"] #defining target variable values
```

[82]:
```python
print("data shape = ",wine_data.shape)
```

```
data shape =  (178, 13)
```

[83]:
```python
print("target shape = ",wine_labels.shape)
```

```
target shape =  (178,)
```

[84]:
```python
import numpy as np
```

[85]:
```python
labels = np.reshape(wine_labels,(178,1))
```

[86]:
```python
final_wine_data = np.concatenate([wine_data,labels],axis=1)
```

[87]:
```python
final_wine_data.shape
```

[87]:
```
(178, 14)
```

[88]:
```python
import pandas as pd
```

[89]:
```python
final_wine_dataset = pd.DataFrame(final_wine_data)
```

[90]:
```python
print(final_wine_dataset)
```

```
          0     1     2     3      4     5     6     7     8      9    10  \
0    14.23  1.71  2.43  15.6  127.0  2.80  3.06  0.28  2.29   5.64  1.04
1    13.20  1.78  2.14  11.2  100.0  2.65  2.76  0.26  1.28   4.38  1.05
2    13.16  2.36  2.67  18.6  101.0  2.80  3.24  0.30  2.81   5.68  1.03
3    14.37  1.95  2.50  16.8  113.0  3.85  3.49  0.24  2.18   7.80  0.86
4    13.24  2.59  2.87  21.0  118.0  2.80  2.69  0.39  1.82   4.32  1.04
..     …     …     …     …      …    …     …    …     …      …     …
173  13.71  5.65  2.45  20.5   95.0  1.68  0.61  0.52  1.06   7.70  0.64
174  13.40  3.91  2.48  23.0  102.0  1.80  0.75  0.43  1.41   7.30  0.70
175  13.27  4.28  2.26  20.0  120.0  1.59  0.69  0.43  1.35  10.20  0.59
176  13.17  2.59  2.37  20.0  120.0  1.65  0.68  0.53  1.46   9.30  0.60
177  14.13  4.10  2.74  24.5   96.0  2.05  0.76  0.56  1.35   9.20  0.61

       11      12   13
0    3.92  1065.0  0.0
1    3.40  1050.0  0.0
2    3.17  1185.0  0.0
3    3.45  1480.0  0.0
4    2.93   735.0  0.0
..    …      …    …
173  1.74   740.0  2.0
174  1.56   750.0  2.0
175  1.56   835.0  2.0
176  1.62   840.0  2.0
177  1.60   560.0  2.0

[178 rows x 14 columns]
```

[91]:
```
features = wine_dataset.feature_names
```

[92]:
```
print("features = \n",features)
```

```
features =
 ['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',
'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins',
'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
```

[93]:
```
features_labels = np.append(features,'label')
```

[94]:
```
final_wine_dataset.columns = features_labels
```

[95]:
```
print(final_wine_dataset)
```

```
   alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
0    14.23        1.71  2.43               15.6      127.0           2.80
1    13.20        1.78  2.14               11.2      100.0           2.65
2    13.16        2.36  2.67               18.6      101.0           2.80
3    14.37        1.95  2.50               16.8      113.0           3.85
```

```
4      13.24        2.59  2.87                21.0      118.0            2.80
..       …          …    …               …         …            …
173    13.71        5.65  2.45                20.5       95.0            1.68
174    13.40        3.91  2.48                23.0      102.0            1.80
175    13.27        4.28  2.26                20.0      120.0            1.59
176    13.17        2.59  2.37                20.0      120.0            1.65
177    14.13        4.10  2.74                24.5       96.0            2.05

     flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
0          3.06                  0.28             2.29             5.64  1.04
1          2.76                  0.26             1.28             4.38  1.05
2          3.24                  0.30             2.81             5.68  1.03
3          3.49                  0.24             2.18             7.80  0.86
4          2.69                  0.39             1.82             4.32  1.04
..          …                   …              …               …   …
173        0.61                  0.52             1.06             7.70  0.64
174        0.75                  0.43             1.41             7.30  0.70
175        0.69                  0.43             1.35            10.20  0.59
176        0.68                  0.53             1.46             9.30  0.60
177        0.76                  0.56             1.35             9.20  0.61

     od280/od315_of_diluted_wines  proline  label
0                            3.92   1065.0    0.0
1                            3.40   1050.0    0.0
2                            3.17   1185.0    0.0
3                            3.45   1480.0    0.0
4                            2.93    735.0    0.0
..                            …        …    …
173                          1.74    740.0    2.0
174                          1.56    750.0    2.0
175                          1.56    835.0    2.0
176                          1.62    840.0    2.0
177                          1.60    560.0    2.0

[178 rows x 14 columns]
```

[96]:
```
print("Unique values of label = ",final_wine_dataset['label'].unique())
```

```
Unique values of label =  [0. 1. 2.]
```

[97]:
```
print("Dataset datatypes =\n",final_wine_dataset.dtypes)
```

```
Dataset datatypes =
 alcohol                         float64
malic_acid                      float64
ash                             float64
alcalinity_of_ash               float64
magnesium                       float64
```

```
total_phenols                   float64
flavanoids                      float64
nonflavanoid_phenols            float64
proanthocyanins                 float64
color_intensity                 float64
hue                             float64
od280/od315_of_diluted_wines    float64
proline                         float64
label                           float64
dtype: object
```

[98]:
```
from sklearn.preprocessing import StandardScaler
x = final_wine_dataset.loc[:, features].values
x = StandardScaler().fit_transform(x) # normalizing the features
```

[99]:
```
print("Feature shape = ",x.shape)
```

```
Feature shape =  (178, 13)
```

[100]:
```
#print mean and standard deviation of normalized data
print("mean = ",np.mean(x))
print("standard deviation = ",np.std(x))
```

```
mean =  4.66735072755122e-16
standard deviation =  1.0
```

[101]:
```
feat_cols = ['feature'+str(i) for i in range(x.shape[1])]
```

[102]:
```
normalised_wine = pd.DataFrame(x,columns=feat_cols)
```

[103]:
```
print(normalised_wine)
```

```
     feature0  feature1  feature2  feature3  feature4  feature5  feature6  \
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
..        …         …         …         …         …         …         …
173  0.876275  2.974543  0.305159  0.301803 -0.332922 -0.985614 -1.424900
174  0.493343  1.412609  0.414820  1.052516  0.158572 -0.793334 -1.284344
175  0.332758  1.744744 -0.389355  0.151661  1.422412 -1.129824 -1.344582
176  0.209232  0.227694  0.012732  0.151661  1.422412 -1.033684 -1.354622
177  1.395086  1.583165  1.365208  1.502943 -0.262708 -0.392751 -1.274305

     feature7  feature8  feature9  feature10  feature11  feature12
0   -0.659563  1.224884  0.251717   0.362177   1.847920   1.013009
1   -0.820719 -0.544721 -0.293321   0.406051   1.113449   0.965242
```

```
2   -0.498407  2.135968  0.269020   0.318304   0.788587   1.395148
3   -0.981875  1.032155  1.186068  -0.427544   1.184071   2.334574
4    0.226796  0.401404 -0.319276   0.362177   0.449601  -0.037874
..        …         …         …          …          …          …
173  1.274310 -0.930179  1.142811  -1.392758  -1.231206  -0.021952
174  0.549108 -0.316950  0.969783  -1.129518  -1.485445   0.009893
175  0.549108 -0.422075  2.224236  -1.612125  -1.485445   0.280575
176  1.354888 -0.229346  1.834923  -1.568252  -1.400699   0.296498
177  1.596623 -0.422075  1.791666  -1.524378  -1.428948  -0.595160

[178 rows x 13 columns]
```

[104]:
```
#PCA analysis
pca_wine = PCA(n_components=3)
```

[105]:
```
principalComponents_wine = pca_wine.fit_transform(x)
```

[106]:
```
principal_wine_Df = pd.DataFrame(data = principalComponents_wine, columns =␣
 ↪['principal component 1', 'principal component 2','principal component 3'])
```

[107]:
```
#3 principal components
print("data frame with principal components = \n",principal_wine_Df)
```

```
data frame with principal components =
      principal component 1  principal component 2  principal component 3
0                 3.316751              -1.443463              -0.165739
1                 2.209465               0.333393              -2.026457
2                 2.516740              -1.031151               0.982819
3                 3.757066              -2.756372              -0.176192
4                 1.008908              -0.869831               2.026688
..                     …                      …                      …
173              -3.370524              -2.216289              -0.342570
174              -2.601956              -1.757229               0.207581
175              -2.677839              -2.760899              -0.940942
176              -2.387017              -2.297347              -0.550696
177              -3.208758              -2.768920               1.013914

[178 rows x 3 columns]
```

[108]:
```
print('Explained variation per principal component: {}'.format(pca_wine.
 ↪explained_variance_ratio_))
```

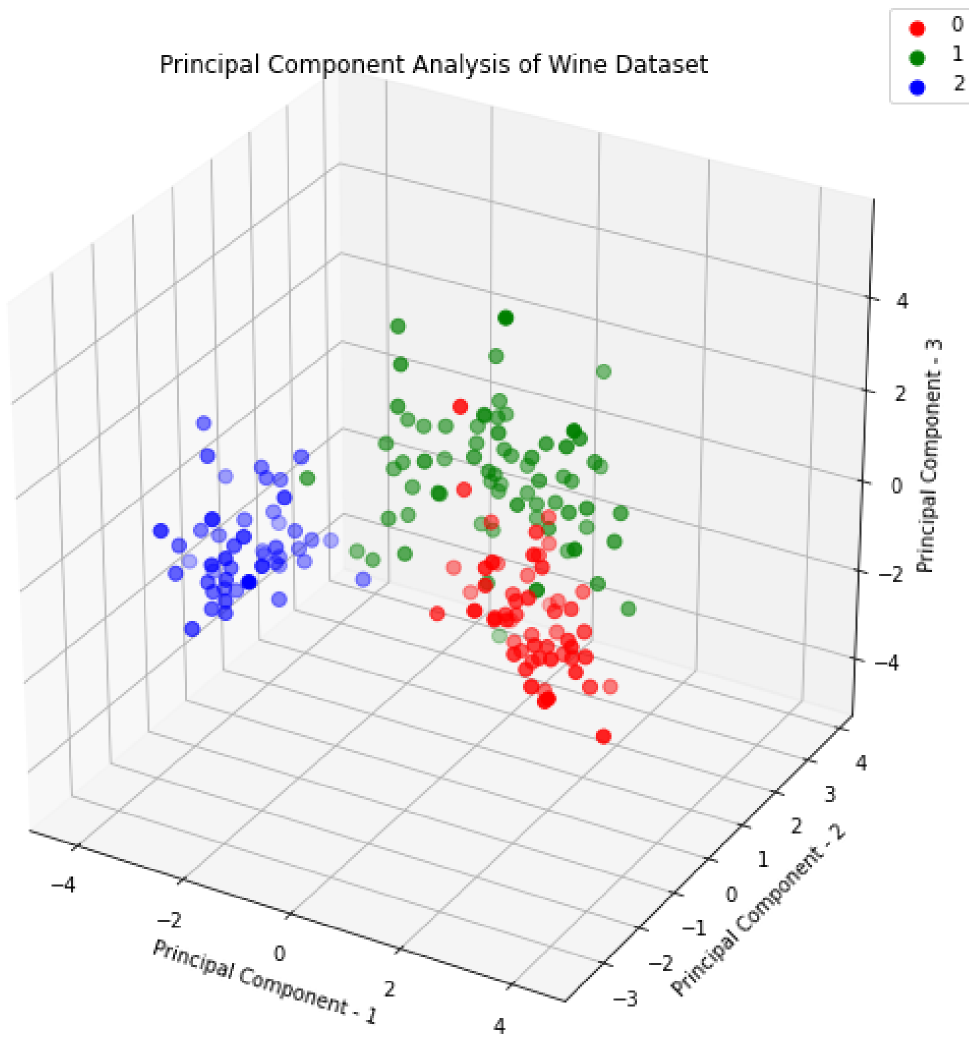
```
Explained variation per principal component: [0.36198848 0.1920749  0.11123631]
```

[109]:
```
#Visualize PCA in 3D

fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(111, projection='3d') #creating 3D subplot
```

```
ax.set_xlabel('Principal Component - 1')
ax.set_ylabel('Principal Component - 2')
ax.set_zlabel('Principal Component - 3')
ax.set_title("Principal Component Analysis of Wine Dataset")
targets = [0,1,2]
colors = ['r', 'g','b']
for target, color in zip(targets,colors):
    indicesToKeep = final_wine_dataset['label'] == target
    ax.scatter(principal_wine_Df.loc[indicesToKeep, 'principal component 1']
               , principal_wine_Df.loc[indicesToKeep, 'principal component 2']
                , principal_wine_Df.loc[indicesToKeep, 'principal component␣
 ↪3'],c = color,marker='o',s=50)

ax.legend(targets)
plt.show()
```



[ ]:

```
```