pca-21mis1152

February 21, 2024

21MIS1152 Rajeev Sekar

[16]:
```
# Import necessary libraries
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import seaborn as sns
```

Loading the dataset

[17]:
```
df=pd.read_csv("BC.csv")
df.head()
```

[17]:
```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   …  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0  …          17.33           184.60      2019.0            0.1622
1  …          23.41           158.80      1956.0            0.1238
2  …          25.53           152.50      1709.0            0.1444
3  …          26.50            98.87       567.7            0.2098
4  …          16.67           152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0             0.6656           0.7119                0.2654          0.4601
1             0.1866           0.2416                0.1860          0.2750
2             0.4245           0.4504                0.2430          0.3613
```

```
3             0.8663           0.6869                0.2575           0.6638
4             0.2050           0.4000                0.1625           0.2364

   fractal_dimension_worst  Unnamed: 32
0                  0.11890          NaN
1                  0.08902          NaN
2                  0.08758          NaN
3                  0.17300          NaN
4                  0.07678          NaN

[5 rows x 33 columns]
```

Data preprocessing

[18]:

```
#First col is id, it is dropped and the last col is unnamed, that is also
 ↪dropped
df = df.iloc[:,1:-1]
df.head()
```

[18]:

```
  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0         M        17.99         10.38          122.80     1001.0
1         M        20.57         17.77          132.90     1326.0
2         M        19.69         21.25          130.00     1203.0
3         M        11.42         20.38           77.58      386.1
4         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
0         0.2419  …         25.38          17.33           184.60
1         0.1812  …         24.99          23.41           158.80
2         0.2069  …         23.57          25.53           152.50
3         0.2597  …         14.91          26.50            98.87
4         0.1809  …         22.54          16.67           152.20

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000
```

```
   concave points_worst  symmetry_worst  fractal_dimension_worst
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 31 columns]
```

[19]:
```
df.shape
```

[19]:
```
(569, 31)
```

[21]:
```
#Separating the feature and target columns
X=df.iloc[:,1:]
Y=df["diagnosis"]
X.head()
```

[21]:
```
   radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean  \
0        17.99         10.38          122.80     1001.0          0.11840
1        20.57         17.77          132.90     1326.0          0.08474
2        19.69         21.25          130.00     1203.0          0.10960
3        11.42         20.38           77.58      386.1          0.14250
4        20.29         14.34          135.10     1297.0          0.10030

   compactness_mean  concavity_mean  concave points_mean  symmetry_mean  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069
3           0.28390          0.2414              0.10520         0.2597
4           0.13280          0.1980              0.10430         0.1809

   fractal_dimension_mean  …  radius_worst  texture_worst  perimeter_worst  \
0                 0.07871  …         25.38          17.33           184.60
1                 0.05667  …         24.99          23.41           158.80
2                 0.05999  …         23.57          25.53           152.50
3                 0.09744  …         14.91          26.50            98.87
4                 0.05883  …         22.54          16.67           152.20

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   concave points_worst  symmetry_worst  fractal_dimension_worst
```

```
0                 0.2654          0.4601                    0.11890
1                 0.1860          0.2750                    0.08902
2                 0.2430          0.3613                    0.08758
3                 0.2575          0.6638                    0.17300
4                 0.1625          0.2364                    0.07678

[5 rows x 30 columns]
```

There are totaly 30 features in this dataset all of which are numerical. We standardize the dataset first then apply PCA to reduce 30 features to 2 principle components

Scaling the data

[24]:
```
scalar = StandardScaler()
X = pd.DataFrame(scalar.fit_transform(X)) #scaling the data
X.head()
```

[24]:
```
          0         1         2         3         4         5         6  \
0  1.097064 -2.073335  1.269934  0.984375  1.568466  3.283515  2.652874
1  1.829821 -0.353632  1.685955  1.908708 -0.826962 -0.487072 -0.023846
2  1.579888  0.456187  1.566503  1.558884  0.942210  1.052926  1.363478
3 -0.768909  0.253732 -0.592687 -0.764464  3.283553  3.402909  1.915897
4  1.750297 -1.151816  1.776573  1.826229  0.280372  0.539340  1.371011

          7         8         9  …        20        21        22        23  \
0  2.532475  2.217515  2.255747  …  1.886690 -1.359293  2.303601  2.001237
1  0.548144  0.001392 -0.868652  …  1.805927 -0.369203  1.535126  1.890489
2  2.037231  0.939685 -0.398008  …  1.511870 -0.023974  1.347475  1.456285
3  1.451707  2.867383  4.910919  … -0.281464  0.133984 -0.249939 -0.550021
4  1.428493 -0.009560 -0.562450  …  1.298575 -1.466770  1.338539  1.220724

         24        25        26        27        28        29
0  1.307686  2.616665  2.109526  2.296076  2.750622  1.937015
1 -0.375612 -0.430444 -0.146749  1.087084 -0.243890  0.281190
2  0.527407  1.082932  0.854974  1.955000  1.152255  0.201391
3  3.394275  3.893397  1.989588  2.175786  6.046041  4.935010
4  0.220556 -0.313395  0.613179  0.729259 -0.868353 -0.397100

[5 rows x 30 columns]
```

PCA is applied to reduce 30 features to 2 principle components

[28]:
```
from sklearn.decomposition import PCA
pca_breast = PCA(n_components=2)
X_PCA = pca_breast.fit_transform(X)
X_PCA
```

[28]:
```
array([[ 9.19283683,  1.94858307],
       [ 2.3878018 , -3.76817174],
       [ 5.73389628, -1.0751738 ],
       …,
       [ 1.25617928, -1.90229671],
       [10.37479406,  1.67201011],
       [-5.4752433 , -0.67063679]])
```

[30]:
```python
PCA_df = pd.DataFrame(data = X_PCA
             , columns = ['principal component 1', 'principal component 2'])
PCA_df.head()
```

[30]:

	principal component 1	principal component 2
0	9.192837	1.948583
1	2.387802	-3.768172
2	5.733896	-1.075174
3	7.122953	10.275589
4	3.935302	-1.948072

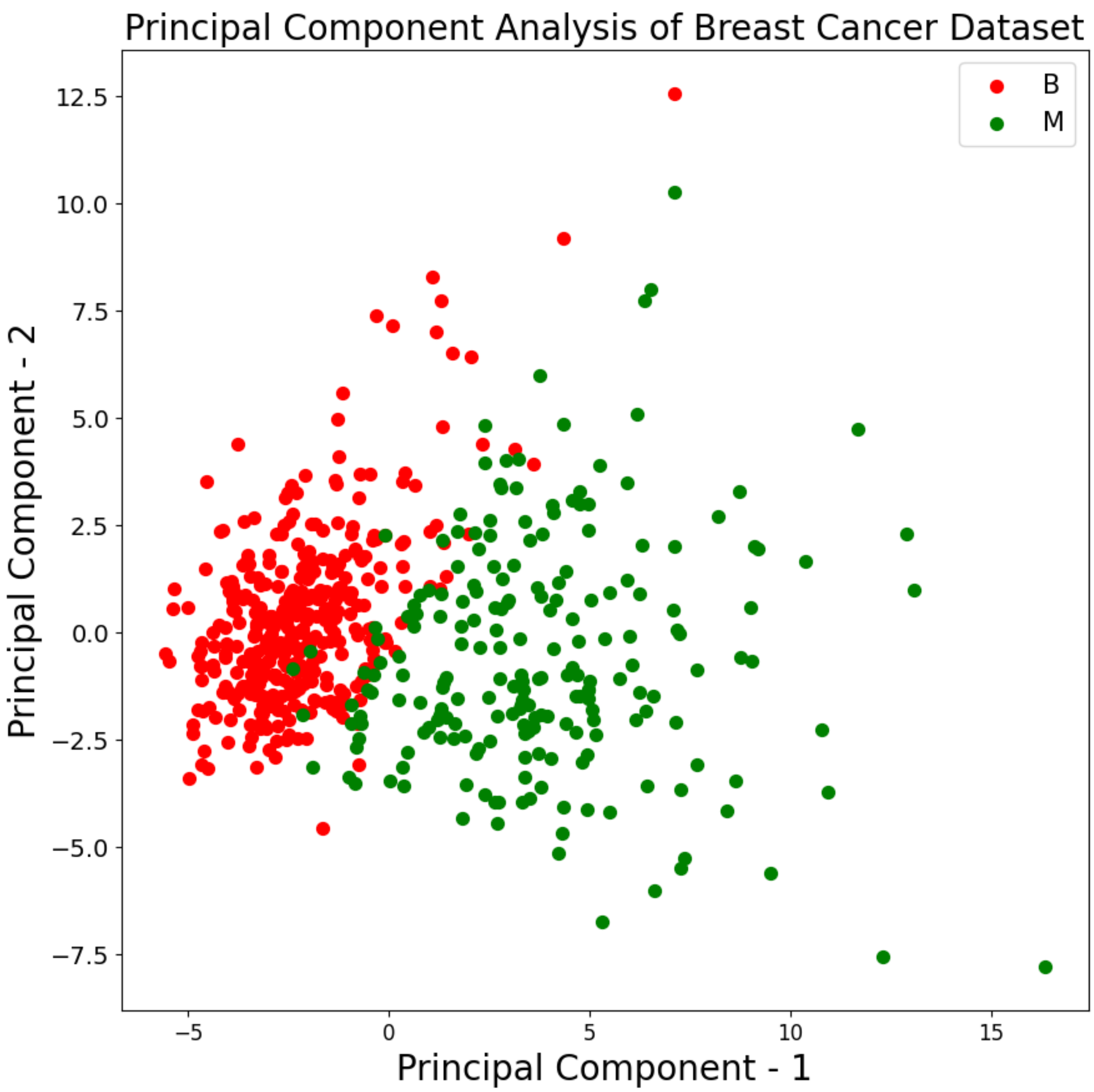
Plotting the bening and malignant data points with respect to the PCs to check how much information is retained from the original dataset

[32]:
```python
import matplotlib.pyplot as plt
plt.figure()
plt.figure(figsize=(10,10))
plt.xticks(fontsize=12)
plt.yticks(fontsize=14)
plt.xlabel('Principal Component - 1',fontsize=20)
plt.ylabel('Principal Component - 2',fontsize=20)
plt.title("Principal Component Analysis of Breast Cancer Dataset",fontsize=20)
targets = ['B', 'M']
colors = ['r', 'g']
for target, color in zip(targets,colors):
    indicesToKeep = Y == target
    plt.scatter(PCA_df.loc[indicesToKeep, 'principal component 1']
               , PCA_df.loc[indicesToKeep, 'principal component 2'], c = color,␣
 ↪s = 50)

plt.legend(targets,prop={'size': 15})
```

[32]:
```
<matplotlib.legend.Legend at 0x216da5a6ed0>

<Figure size 640x480 with 0 Axes>
```



We can see that the 2 classes (bening and malignant) is projected in 2d whereas actually it had 30 features and the data points are to a great extent, linearly separable => the meaning of the features is not lost to an extent where it affects the diognosis

[ ]: