Chapter 16 - exercise 2: Skin Color

```
from google.colab import drive
drive.mount("/content/gdrive", force_remount=True)

path = '/content/gdrive/My Drive/LDS6_MachineLearning/'
```

```
Mounted at /content/gdrive
```

Cho dữ liệu skin nằm trong tập tin Skin_NonSkin.txt.

- Bộ dữ liệu phân loại da (Skin Segmentation) được tạo thành từ 3 không gian màu B, G, R. Dữ liệu sử dụng kết cấu da từ hình ảnh khuôn mặt với sự đa dạng về độ tuổi, giới tính,...

  Có (245057 * 4) sample với 3 cột đầu là B,G,R (x1,x2, và x3 features), cột thứ tư là cla

  Áp dụng thuật toán PCA để trực quan hóa dữ liệu với 2 thành phần thay vì 3 t

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn import svm
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler


data = pd.read_csv(path + "practice/Chapter16_PCA/Skin_NonSkin.txt", sep='\t', header= None)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245057 entries, 0 to 245056
Data columns (total 4 columns):
0    245057 non-null int64
1    245057 non-null int64
2    245057 non-null int64
3    245057 non-null int64
dtypes: int64(4)
memory usage: 7.5 MB
```

```
data.head(3)
```

	0	1	2	3
0	74	85	123	1
1	73	84	122	1
2	72	83	121	1

```
X = data.iloc[:, :-1]
y = data.iloc[:, -1]
```

Trực quan hóa dữ liệu

```
X.head(3)
```

	0	1	2
0	74	85	123
1	73	84	122
2	72	83	121

```
X = X.astype('float')
```

```
X = StandardScaler().fit_transform(X)
```

```
X = pd.DataFrame(data = X, columns = [0, 1, 2])
X.head(3)
```

	0	1	2
0	-0.820256	-0.792567	-0.002441
1	-0.836318	-0.809250	-0.016223
2	-0.852381	-0.825933	-0.030004

```
y = np.array(data[3])
y = pd.DataFrame(data = y, columns = ['result'])
y.head(3)
```

	result
0	1
1	1
2	1

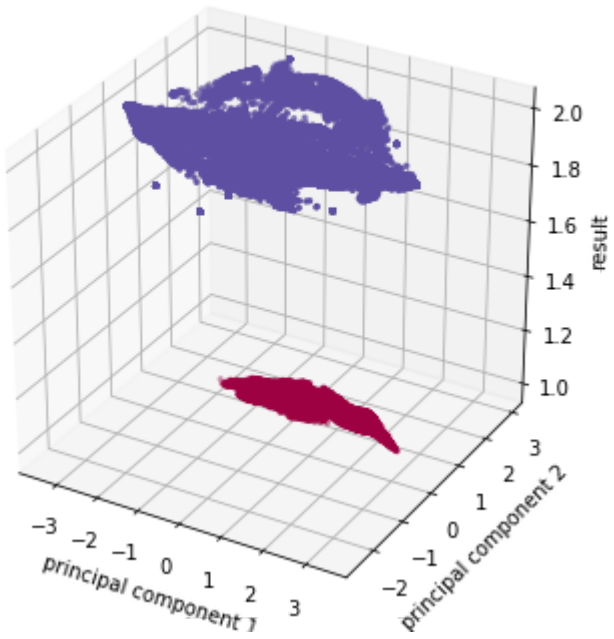
```
pca = PCA(n_components=2)

principalComponents = pca.fit_transform(X)

principalDf = pd.DataFrame(data = principalComponents,
                           columns = ['principal component 1',
                                      'principal component 2'])

principalDf.head(3)
```

	principal component 1	principal component 2
0	-0.975032	0.583094
1	-1.001979	0.583404
2	-1.028925	0.583714

```
finalDf = pd.concat([principalDf, y], axis = 1)
finalDf.head(3)
```

	principal component 1	principal component 2	result
0	-0.975032	0.583094	1
1	-1.001979	0.583404	1
2	-1.028925	0.583714	1

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(finalDf['principal component 1'],
           finalDf['principal component 2'],
           finalDf['result'],
           c=finalDf['result'],
           marker = '.', cmap=plt.cm.Spectral)
ax.set_xlabel('principal component 1')
ax.set_ylabel('principal component 2')
ax.set_zlabel('result')
plt.show()
```




```
pca.explained_variance_ratio_
```

```
array([0.784023  , 0.17671028])
```

```
pca.explained_variance_ratio_.sum()
```

```
0.960733283855423
```