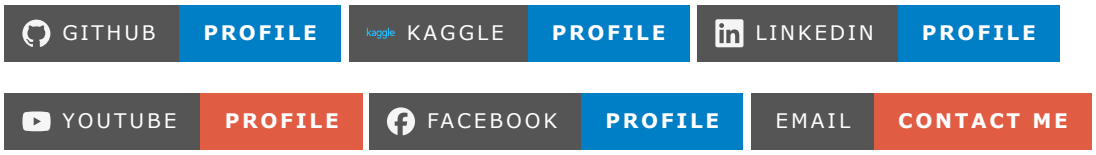
Principal Component Analysis (PCA)

Breast Cancer Data Exploration

In this project, you will mainly apply PCA on the two use-cases:

1. Data Visualization
2. Speeding ML algorithm

Aurther :Irfan Ullah Khan




```
from sklearn.datasets import load_breast_cancer
```

```
breast = load_breast_cancer()
```

```
breast_data = breast.data
```

```
breast_data.shape
```

```
(569, 30)
```

```
breast_labels = breast.target
```

```
breast_labels.shape
```

```
(569,)
```

```
import numpy as np
```

```
labels = np.reshape(breast_labels,(569,1))
```

```
final_breast_data = np.concatenate([breast_data,labels],axis=1)
```

```
final_breast_data.shape
```

```
(569, 31)
```

```
import pandas as pd
```

```
breast_dataset = pd.DataFrame(final_breast_data)
```

```
features = breast.feature_names
```

```
features
```

```
array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal dimension',
       'radius error', 'texture error', 'perimeter error', 'area error',
       'smoothness error', 'compactness error', 'concavity error',
       'concave points error', 'symmetry error',
       'fractal dimension error', 'worst radius', 'worst texture',
       'worst perimeter', 'worst area', 'worst smoothness',
       'worst compactness', 'worst concavity', 'worst concave points',
       'worst symmetry', 'worst fractal dimension'], dtype='<U23')
```

```
features_labels = np.append(features,'label')
```

```
breast_dataset.columns = features_labels
```

```
breast_dataset.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	symme
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1

5 rows × 31 columns

```
breast_dataset['label'].replace(0, 'Benign',inplace=True)
breast_dataset['label'].replace(1, 'Malignant',inplace=True)
```

```
breast_dataset.tail()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	sy
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	

5 rows × 31 columns

Visualizing the Breast Cancer data

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

```
x.shape
```

```
(569, 30)
```

```
np.mean(x),np.std(x)
```

```
(-6.118909323768877e-16, 1.0)
```

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

normalised_breast = pd.DataFrame(x,columns=feat_cols)

normalised_breast.tail()
```

	feature0	feature1	feature2	feature3	feature4	feature5	feature6	feature7	fe
564	2.110995	0.721473	2.060786	2.343856	1.041842	0.219060	1.947285	2.320965	-0
565	1.704854	2.085134	1.615931	1.723842	0.102458	-0.017833	0.693043	1.263669	-0
566	0.702284	2.045574	0.672676	0.577953	-0.840484	-0.038680	0.046588	0.105777	-0
567	1.838341	2.336457	1.982524	1.735218	1.525767	3.272144	3.296944	2.658866	2
568	-1.808401	1.221792	-1.814389	-1.347789	-3.112085	-1.150752	-1.114873	-1.261820	-0

5 rows × 30 columns

```
from sklearn.decomposition import PCA
pca_breast = PCA(n_components=2)
principalComponents_breast = pca_breast.fit_transform(x)

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

principal_breast_Df.tail()
```

	principal component 1	principal component 2
564	6.439315	-3.576817
565	3.793382	-3.584048
566	1.256179	-1.902297
567	10.374794	1.672010
568	-5.475243	-0.670637

```
print('Explained variation per principal component: {}'.format(pca_breast.explained_variance
```

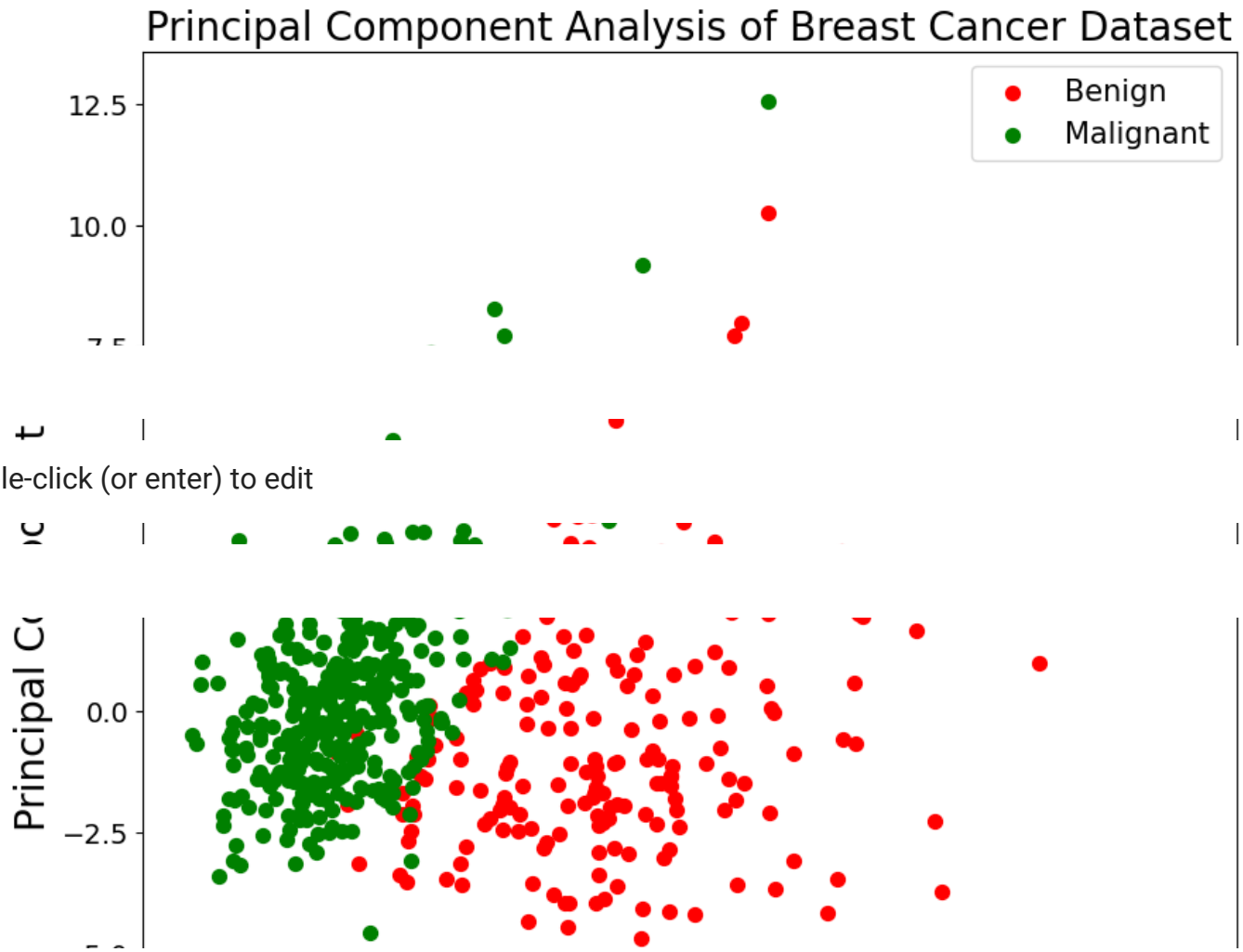
```
Explained variation per principal component: [0.44272026 0.18971182]
```

```
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 = ['Benign', 'Malignant']
colors = ['r', 'g']
for target, color in zip(targets,colors):
    indicesToKeep = breast_dataset['label'] == target
    plt.scatter(principal_breast_Df.loc[indicesToKeep, 'principal component 1']
               , principal_breast_Df.loc[indicesToKeep, 'principal component 2'], c = color,

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

```
<matplotlib.legend.Legend at 0x7f67dd1dc8e0>
<Figure size 640x480 with 0 Axes>
```



Double-click (or enter) to edit