In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd
%matplotlib inline
```

```
C:\Users\admin\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarn
ing: A NumPy version >=1.16.5 and <1.23.0 is required for this version of
SciPy (detected version 1.23.5
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()
```

In [3]:
```
cancer.keys()
```

Out[3]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_na
mes', 'filename'])
```

In [4]:
```
print(cancer['DESCR'])
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the cla
ss

    :Attribute Information:
        - radius (mean of distances from center to points on the perimet
er)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
```

In [5]:
```
cancer['data'].shape
```

Out[5]:
```
(569, 30)
```

In [6]:
```python
df = pd.DataFrame(cancer['data'], columns = cancer['feature_names'])
df.head()
```

Out[6]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	me symme
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.24
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.18
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.20
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.25
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.18

5 rows × 30 columns

In [7]:
```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler.fit(df)
```

Out[7]:
```
StandardScaler()
```

In [8]:
```python
scaled_data = scaler.transform(df)

scaled_data
```

Out[8]:
```
array([[ 1.09706398, -2.07333501,  1.26993369, ...,  2.29607613,
         2.75062224,  1.93701461],
       [ 1.82982061, -0.35363241,  1.68595471, ...,  1.0870843 ,
        -0.24388967,  0.28118999],
       [ 1.57988811,  0.45618695,  1.56650313, ...,  1.95500035,
         1.152255  ,  0.20139121],
       ...,
       [ 0.70228425,  2.0455738 ,  0.67267578, ...,  0.41406869,
        -1.10454895, -0.31840916],
       [ 1.83834103,  2.33645719,  1.98252415, ...,  2.28998549,
         1.91908301,  2.21963528],
       [-1.80840125,  1.22179204, -1.81438851, ..., -1.74506282,
        -0.04813821, -0.75120669]])
```

In [9]:
```python
from sklearn.decomposition import PCA
pca = PCA(n_components = 2)
```

In [10]:
```python
pca.fit(scaled_data)
```

Out[10]:
```
PCA(n_components=2)
```

In [11]:
```python
x_pca = pca.transform(scaled_data)
```

In [12]:
```python
scaled_data.shape
```

Out[12]:
```
(569, 30)
```

In [13]:
```
x_pca.shape
```

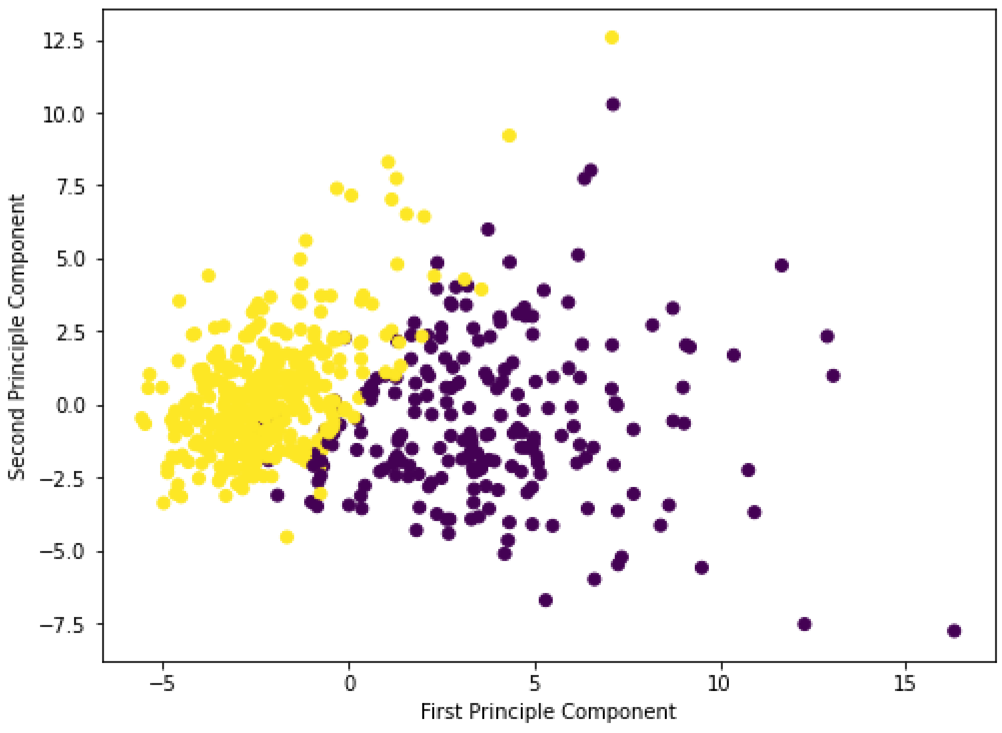
Out[13]:
```
(569, 2)
```

In [14]:
```
print(x_pca)
```

```
[[ 9.19283683  1.94858307]
 [ 2.3878018  -3.76817174]
 [ 5.73389628 -1.0751738 ]
 ...
 [ 1.25617928 -1.90229671]
 [10.37479406  1.67201011]
 [-5.4752433  -0.67063679]]
```

In [15]:
```
plt.figure(figsize = (8,6))
plt.scatter(x_pca[:,0], x_pca[:,1], c = cancer['target'])
plt.xlabel('First Principle Component')
plt.ylabel('Second Principle Component')
```

Out[15]:
```
Text(0, 0.5, 'Second Principle Component')
```


Second Principle Component
First Principle Component