In [1]:

```
import pandas as pd
import seaborn as sns
%matplotlib inline
```

In [2]:

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

In [3]:

```
cancer = load_breast_cancer()
```

In [31]:

```
cancer.keys()
```

Out[31]:

```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])
```

In [32]:

```
print(cancer['DESCR'])
```

```
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree.  Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

.. topic:: References

   - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction
```

In [35]:

```
df=pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
```

In [36]:

```
df.head()
```

Out[36]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	per
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	

5 rows × 30 columns

In [37]:

```
df.shape
```

Out[37]:

```
(569, 30)
```

In [38]:

```
from sklearn.preprocessing import StandardScaler
```

In [39]:

```
scaler=StandardScaler()
```

In [40]:

```
scaler.fit(df)
```

Out[40]:

```
StandardScaler()
```

In [41]:

```
sd=scaler.transform(df)
```

In [42]:

```
from sklearn.decomposition import PCA
```

In [43]:

```
pca=PCA(n_components=2)
```

In [44]:

```
pca.fit(sd)
```

Out[44]:

```
PCA(n_components=2)
```

In [45]:

```
x=pca.transform(sd)
```

In [46]:

```
sd.shape
```

Out[46]:

```
(569, 30)
```

In [47]:

```
x.shape
```

Out[47]:

```
(569, 2)
```

In [48]:

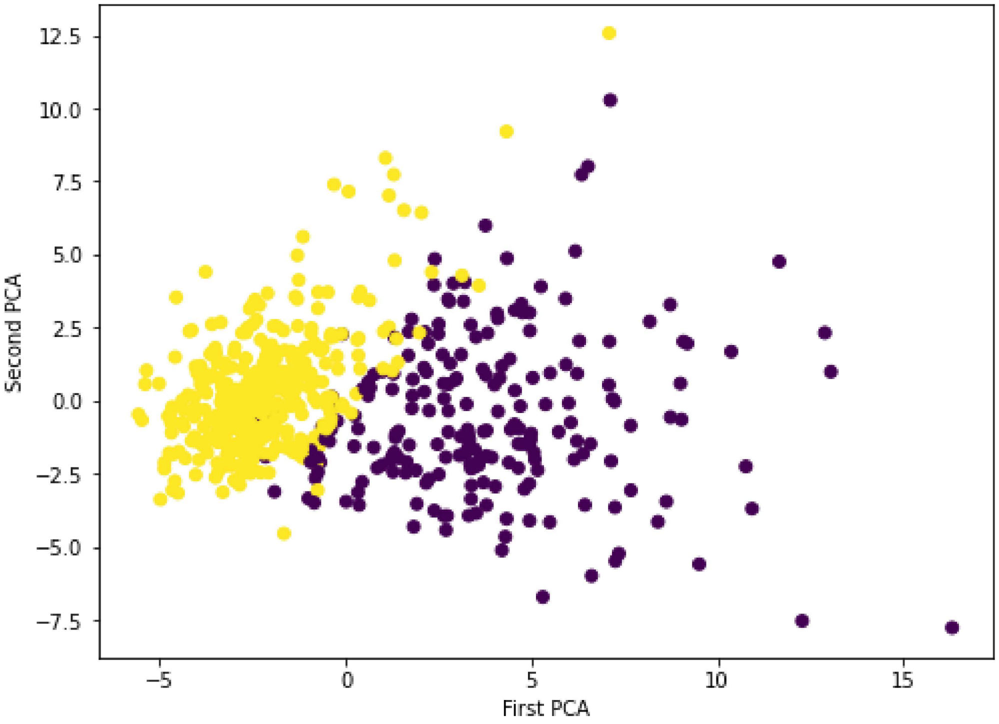
```
import matplotlib.pyplot as plt
```

In [51]:

```
plt.figure(figsize=(8,6))
plt.scatter(x[:,0],x[:,1],c=cancer['target'])
plt.xlabel('First PCA')
plt.ylabel('Second PCA')
```

Out[51]:

```
Text(0, 0.5, 'Second PCA')
```



In [ ]: