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

In [34]:
```
print(cancer.DESCR)
```

```
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
     for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on
     Electronic Imaging: Science and Technology, volume 1905, pages 861-870,
     San Jose, CA, 1993.
   - O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and
     prognosis via linear programming. Operations Research, 43(4), pages 570-577,
     July-August 1995.
   - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques
     to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994)
     163-171.
```

In [2]:

```
from sklearn.datasets import load_breast_cancer

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

df.head()
```

Out[2]:

mean ıcavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

In [50]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
model = LogisticRegression()

x_train,x_test,y_train,y_test = train_test_split(df,cancer['target'],random_state=1)

model.fit(x_train,y_train)

print("Testing Accuracy : \n",classification_report(model.predict(x_test),y_test),"\n")

print("Training Accuracy : \n",classification_report(model.predict(x_train),y_train))
```

```
Testing Accuracy :
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        54
           1       0.99      0.98      0.98        89

    accuracy                           0.98       143
   macro avg       0.98      0.98      0.98       143
weighted avg       0.98      0.98      0.98       143


Training Accuracy :
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       155
           1       1.00      0.99      0.99       271

    accuracy                           0.99       426
   macro avg       0.99      0.99      0.99       426
weighted avg       0.99      0.99      0.99       426
```

In [17]:
```
from sklearn.preprocessing import StandardScaler

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

Out[17]:
```
StandardScaler()
```

In [19]:
```
scaled_data = scaler.transform(df)
```

In [20]:
```
from sklearn.decomposition import PCA
```

In [40]:
```
pca = PCA(n_components=4)
```

In [41]:
```
pca_df = pca.fit_transform(df)
```

In [42]:
```
pca_df = pd.DataFrame(pca_df,columns = ['PC1','PC2','PC3','PC4'])
```

In [43]:
```
pca_df.head()
```

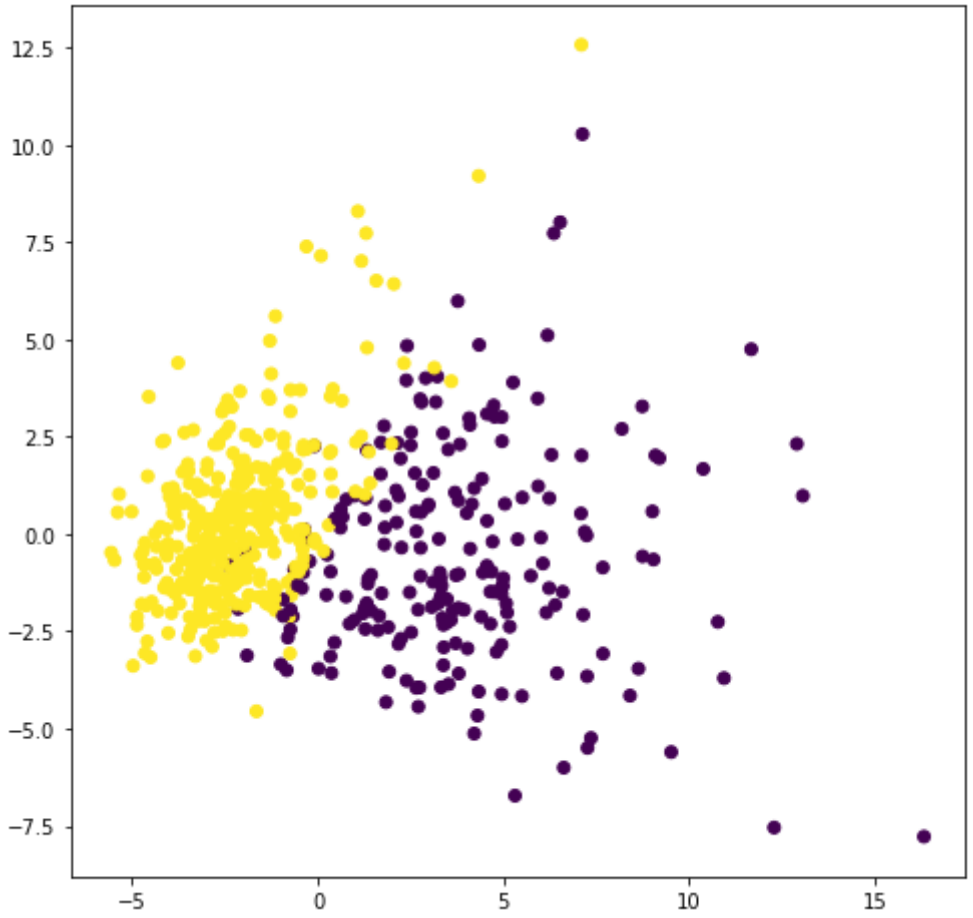
Out[43]:

	PC1	PC2	PC3	PC4
0	9.192837	1.948583	-1.123167	3.633731
1	2.387802	-3.768172	-0.529292	1.118264
2	5.733896	-1.075174	-0.551747	0.912083
3	7.122953	10.275589	-3.232790	0.152547
4	3.935302	-1.948072	1.389767	2.940640

In [44]:
```
plt.figure(figsize = (8,8))

plt.scatter(pca_df['PC1'],pca_df['PC2'],c = cancer['target'])
```

Out[44]: <matplotlib.collections.PathCollection at 0x20ef8dd4508>



In [45]:
```
pca.explained_variance_ratio_.sum()
```

Out[45]:
```
0.7923850582446045
```

In [51]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
model = LogisticRegression()

x_train,x_test,y_train,y_test = train_test_split(pca_df,cancer['target'],random_state=1)

model.fit(x_train,y_train)

print("Testing Accuracy : \n",classification_report(model.predict(x_test),y_test),"\n")

print("Training Accuracy : \n",classification_report(model.predict(x_train),y_train))
```

```
Testing Accuracy :
               precision    recall  f1-score   support

           0       0.93      0.96      0.94        53
           1       0.98      0.96      0.97        90

    accuracy                           0.96       143
   macro avg       0.95      0.96      0.96       143
weighted avg       0.96      0.96      0.96       143


Training Accuracy :
               precision    recall  f1-score   support

           0       0.96      0.97      0.96       154
           1       0.99      0.97      0.98       272

    accuracy                           0.97       426
   macro avg       0.97      0.97      0.97       426
weighted avg       0.97      0.97      0.97       426
```

In [ ]:
```
1
```