In [66]:
```
import pandas as pd
import numpy as np
```

In [67]:
```
df = pd.read_csv('breast_cancer.csv')
```

In [68]:
```
df.head(10)
```

Out[68]:

| | Sample code number | Clump Thickness | Uniformity of Cell Size | Uniformity of Cell Shape | Marginal Adhesion | Single Epithelial Cell Size | Bare Nuclei | Bland Chromatin | Normal Nucleoli | Mitoses | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1000025 | 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| **1** | 1002945 | 5 | 4 | 4 | 5 | 7 | 10 | 3 | 2 | 1 | 2 |
| **2** | 1015425 | 3 | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 1 | 2 |
| **3** | 1016277 | 6 | 8 | 8 | 1 | 3 | 4 | 3 | 7 | 1 | 2 |
| **4** | 1017023 | 4 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 1 | 2 |
| **5** | 1017122 | 8 | 10 | 10 | 8 | 7 | 10 | 9 | 7 | 1 | 4 |
| **6** | 1018099 | 1 | 1 | 1 | 1 | 2 | 10 | 3 | 1 | 1 | 2 |
| **7** | 1018561 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| **8** | 1033078 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 5 | 2 |
| **9** | 1033078 | 4 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 |

In [69]:
```
df.shape
```

Out[69]:
```
(683, 11)
```

In [70]:
```
# ANOTHER METHOD TO DOWNLOAD A FILE FROM UCI MACHINE LEARNING REPOSITORY
df1 = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data',
                  names =[' Sample code number','Clump Thickness',' Uniformity of Cell Size',' Uniformity of Cell Shape','Marginal Adhesion',' Single Epithelial Cell Size','Bare Nuc
```

In [71]:
```
df1.head(10)
```

Out[71]:

| | Sample code number | Clump Thickness | Uniformity of Cell Size | Uniformity of Cell Shape | Marginal Adhesion | Single Epithelial Cell Size | Bare Nuclei | Bland Chromatin | Normal Nucleoli | Mitoses | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1000025 | 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| **1** | 1002945 | 5 | 4 | 4 | 5 | 7 | 10 | 3 | 2 | 1 | 2 |
| **2** | 1015425 | 3 | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 1 | 2 |
| **3** | 1016277 | 6 | 8 | 8 | 1 | 3 | 4 | 3 | 7 | 1 | 2 |
| **4** | 1017023 | 4 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 1 | 2 |
| **5** | 1017122 | 8 | 10 | 10 | 8 | 7 | 10 | 9 | 7 | 1 | 4 |
| **6** | 1018099 | 1 | 1 | 1 | 1 | 2 | 10 | 3 | 1 | 1 | 2 |
| **7** | 1018561 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| **8** | 1033078 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 5 | 2 |
| **9** | 1033078 | 4 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 |

In [72]:
```
df1.shape
```

Out[72]:
```
(699, 11)
```

In [73]:
```
X = df.iloc[: , 1:-1].values
y = df.iloc[: , -1].values
```

In [74]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test =train_test_split(X,y,test_size=0.2,random_state = 0)
```

In [75]:
```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression( random_state = 0)
classifier.fit(X_train,y_train)
```

Out[75]:
```
LogisticRegression(random_state=0)
```

In [76]:
```
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1),y_test.reshape(len(y_test),1)),1))
```

```
[[2 2]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [4 4]
 [4 2]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [2 2]
 [4 2]
 [4 4]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [2 2]
 [2 4]
 [2 4]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [2 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [4 4]
 [4 4]
 [2 2]
 [4 4]
 [2 2]
 [4 4]
 [2 2]
 [2 2]
 [2 2]
 [4 4]
 [2 2]]
```

In [77]:
```
#applying confusion matrix and checking accuracy

from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test,y_pred)
print(cm)
accuracy_score(y_test,y_pred)
```

```
[[84  3]
 [ 3 47]]
```

Out[77]:
```
0.9562043795620438
```

In [78]:
```
#applying k-fold cross validation technique

from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X=X, y=y, cv=10)
print('accuracy:{:.2f}%'.format(accuracies.mean()*100))
print('standard deviation:{:.2f}%'.format(accuracies.std()*100))
```

```
accuracy:96.70%
standard deviation:2.50%
```

In [ ]: