In [ ]:

```
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
```

In [ ]:

```
data = pd.read_csv('echocardiogram1.csv')
# Printing the dataset shape
print ("Dataset Length: ", len(data))
print ("Dataset Shape: ", data.shape)
# Printing the dataset obseravtions
print ("Dataset:\n ",data.head())
```

```
Dataset Length:  60
Dataset Shape:  (60, 12)
Dataset:
      survie  still_alive   age  pceffusion  fraction_shorting    epss   lvdd  \
0    19.0            0  72.0           0              0.380   6.000  4.100
1    16.0            0  55.0           0              0.260   4.000  3.420
2    57.0            0  60.0           0              0.253  12.062  4.603
3    19.0            1  57.0           0              0.160  22.000  5.750
4    26.0            0  68.0           0              0.260   5.000  4.310

   wall_motion_score  wall_motion_index   mult  group  alive_at_1
0               14.0               1.70  0.588      1           0
1               14.0               1.00  1.000      1           0
2               16.0               1.45  0.788      1           0
3               18.0               2.25  0.571      1           0
4               12.0               1.00  0.857      1           0
```

In [ ]:

```
# data['class'],_ = pd.factorize(data['class'])
## data.info()
# Separating the target variable
X = data.iloc[:, 0:11].values
y = data.iloc[:, 11].values
# Splitting the dataset into train and test
# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random
```

In [ ]:

```
data.head()
```

Out[ ]:

	survie	still_alive	age	pceffusion	fraction_shorting	epss	lvdd	wall_motion_score	wall_moti
0	19.0	0	72.0	0	0.380	6.000	4.100	14.0	
1	16.0	0	55.0	0	0.260	4.000	3.420	14.0	
2	57.0	0	60.0	0	0.253	12.062	4.603	16.0	
3	19.0	1	57.0	0	0.160	22.000	5.750	18.0	
4	26.0	0	68.0	0	0.260	5.000	4.310	12.0	

In [ ]:

```
X[0]
```

Out[ ]:

```
array([19.   ,  0.   , 72.   ,  0.   ,  0.38 ,  6.   ,  4.1  , 14.   ,
        1.7  ,  0.588,  1.   ])
```

In [ ]:

```
y[0]
```

Out[ ]:

```
0
```

In [ ]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_s
```

In [ ]:

```
gnb = GaussianNB()
gnb.fit(X_train, y_train)
# making predictions on the testing set
y_pred = gnb.predict(X_test)
```

In [ ]:

```
print("Results Using Gaussian Naive Bayes Classifier\n\n")
print("Confusion Matrix:\n ", confusion_matrix(y_test, y_pred))
print ("Accuracy : ", accuracy_score(y_test,y_pred)*100)
print("Report :\n ", classification_report(y_test, y_pred))
```

```
Results Using Gaussian Naive Bayes Classifier


Confusion Matrix:
  [[10  0]
 [ 0  5]]
Accuracy :  100.0
Report :
                precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         5

    accuracy                           1.00        15
   macro avg       1.00      1.00      1.00        15
weighted avg       1.00      1.00      1.00        15
```

In [ ]: