In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use("seaborn")
import pickle
```

```
C:\Users\VAISHN~1\AppData\Local\Temp/ipykernel_11224/1242344652.py:5: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_8-<style>'. Alternatively, directly use the seaborn API instead.
  plt.style.use("seaborn")
```

In [2]:
```
df=pd.read_csv(r"C:\Users\VAISHNAVI\Downloads\Breast_cancer_data.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
0	17.99	10.38	122.80	1001.0	0.11840	0
1	20.57	17.77	132.90	1326.0	0.08474	0
2	19.69	21.25	130.00	1203.0	0.10960	0
3	11.42	20.38	77.58	386.1	0.14250	0
4	20.29	14.34	135.10	1297.0	0.10030	0

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
mean_radius        0
mean_texture       0
mean_perimeter     0
mean_area          0
mean_smoothness    0
diagnosis          0
dtype: int64
```

In [5]:
```
x=df.iloc[:,:5]
y=df.iloc[:,5]
```

In [6]:
```
1 x
```

Out[6]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness
0	17.99	10.38	122.80	1001.0	0.11840
1	20.57	17.77	132.90	1326.0	0.08474
2	19.69	21.25	130.00	1203.0	0.10960
3	11.42	20.38	77.58	386.1	0.14250
4	20.29	14.34	135.10	1297.0	0.10030
...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100
565	20.13	28.25	131.20	1261.0	0.09780
566	16.60	28.08	108.30	858.1	0.08455
567	20.60	29.33	140.10	1265.0	0.11780
568	7.76	24.54	47.92	181.0	0.05263

569 rows × 5 columns

In [7]:
```
1 y
```

Out[7]:
```
0      0
1      0
2      0
3      0
4      0
      ..
564    0
565    0
566    0
567    0
568    1
Name: diagnosis, Length: 569, dtype: int64
```

In [8]:
```
1 from sklearn.model_selection import train_test_split
2 X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=40,random_state
```

In [9]:
```
1 X_train.head()
```

Out[9]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness
104	10.49	19.29	67.41	336.1	0.09989
353	15.08	25.74	98.00	716.6	0.10240
422	11.61	16.02	75.46	408.2	0.10880
211	11.84	18.94	75.51	428.0	0.08871
275	11.89	17.36	76.20	435.6	0.12250

In [10]:
```
1 X_test.head()
```

Out[10]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness
204	12.47	18.60	81.09	481.9	0.09965
70	18.94	21.31	123.60	1130.0	0.09009
131	15.46	19.48	101.70	748.9	0.10920
431	12.40	17.68	81.47	467.8	0.10540
540	11.54	14.44	74.65	402.9	0.09984

In [11]:
```
1 y_train.head()
```

Out[11]:
```
104    1
353    0
422    1
211    1
275    1
Name: diagnosis, dtype: int64
```

In [12]:
```
1 y_test.head()
```

Out[12]:
```
204    1
70     0
131    0
431    1
540    1
Name: diagnosis, dtype: int64
```

In [13]:
```
1 from sklearn.preprocessing import StandardScaler
2 sc = StandardScaler()
3 x_train = sc.fit_transform(X_train)
4 x_test = sc.transform(X_test)
```

In [14]:
```
1 x_train
```

Out[14]:
```
array([[-1.03117326e+00, -1.29455649e-03, -1.01096323e+00,
        -9.01727583e-01,  2.58639436e-01],
       [ 2.72424571e-01,  1.50111751e+00,  2.50231024e-01,
         1.75803691e-01,  4.36659683e-01],
       [-7.13084028e-01, -7.62982535e-01, -6.79070008e-01,
        -6.97548858e-01,  8.90575849e-01],
       ...,
       [ 4.80580600e-02, -5.76636852e-01, -6.72320615e-02,
        -6.20744352e-02, -2.26626924e+00],
       [-3.99844949e-02,  7.55730377e-02, -3.34243043e-02,
        -1.55809744e-01,  6.91987526e-01],
       [-5.51199330e-01,  2.85211931e-01, -6.06507017e-01,
        -5.54538794e-01, -1.14282673e+00]])
```

In [15]:
```
1 x_test
```

Out[15]:
```
array([[-0.46883694, -0.16201771, -0.44695089, -0.48883912,  0.24161758],
       [ 1.36869638,  0.46922829,  1.30569271,  1.3465039 , -0.43641969],
       [ 0.3803477 ,  0.04296254,  0.40277822,  0.26727349,  0.91894561],
       [-0.48871752, -0.37631524, -0.43128389, -0.52876866,  0.64943289],
       [-0.7329646 , -1.13101526, -0.71246548, -0.71255784,  0.25509322],
       [ 1.84015007,  2.33734376,  1.98597075,  1.72880803,  1.52889546],
       [ 2.24060169,  0.60665823,  2.27869645,  2.34332652,  0.71326485],
       [ 0.97960509, -0.98892668,  0.95112355,  0.85120766,  0.15792679],
       [-0.22174977, -0.80025167, -0.22431445, -0.3809444 ,  0.81965145],
       [-0.06270515, -0.62322327, -0.12206659, -0.15609293, -1.98328088],
       [-0.25867084,  1.38698078, -0.32285169, -0.3302537 , -0.59103489],
       [ 0.75523858, -0.11543129,  0.71611841,  0.65637414, -0.53855083],
       [ 0.24118366,  0.10818353,  0.14715859,  0.10104199, -0.85345517],
       [ 0.6075543 ,  0.60199959,  0.64190626,  0.48787714,  1.43669374],
       [-0.50575801, -1.63647792, -0.53559318, -0.52678635, -0.44067516],
       [ 1.4141377 ,  1.62923017,  1.53245206,  1.35216766,  1.79131574],
       [ 0.19574235, -1.07045291,  0.11252626,  0.07413911, -0.81728373],
       [-1.12489598,  0.06858507, -1.12145688, -0.97082552,  0.28771844],
       [-2.02775818, -1.36627668, -1.98479156, -1.44714814,  1.47215594],
       [ 1.82594965,  0.36440885,  1.89114412,  1.85057898,  0.59269336],
       [-0.09110598, -0.81655692, -0.0618723 , -0.19942073,  0.3160882 ],
       [-0.57391999, -0.36699796, -0.57146239, -0.58993732,  0.47212188],
       [ 2.12699839,  0.69517243,  2.16325533,  2.13093534,  1.45087862],
       [-1.16181705,  0.46224033, -1.1849495 , -0.9832858 , -1.06906535],
       [-0.71592411,  1.20995238, -0.72978164, -0.67206192, -1.52510925],
       [-0.1677882 , -1.94627762, -0.1657693 , -0.26993461,  2.33034119],
       [-0.23879026, -1.29872637, -0.25317473, -0.31920936, -0.89459132],
       [-1.08797491,  1.93670054, -1.08270165, -0.94335626, -0.42152557],
       [-0.38647455, -0.106114  , -0.4147923 , -0.44721045,  0.03664606],
       [ 0.87452204,  1.21694034,  0.91814037,  0.77842828,  0.77000437],
       [-0.84940798,  0.73244157, -0.84274903, -0.78193896, -0.04137078],
       [-0.58528032, -1.52699983, -0.62217403, -0.58314081, -0.222228  ],
       [-1.24361142, -0.04089301, -1.236898  , -1.03312693,  0.79837413],
       [-0.12518697, -0.69077358, -0.17195365, -0.22377492, -0.2463423 ],
       [-1.11921581, -0.41125506, -1.10537758, -0.96686088,  0.69907997],
       [-0.73864477, -0.12707789, -0.76647543, -0.69528335, -0.07115903],
       [ 0.53939232,  0.91878725,  0.44400719,  0.40575255, -1.00594263],
       [-0.67616296, -1.22884674, -0.72936935, -0.6434599 , -1.28467553],
       [ 1.1727307 ,  0.15942859,  1.14077682,  1.09163448, -0.11442291],
       [ 0.14178078,  1.10047429,  0.10881565,  0.02316523, -0.46124323]])
```

In [16]:
```
1 from sklearn.naive_bayes import GaussianNB
2 clf = GaussianNB()
3 clf.fit(x_train, y_train)
```

Out[16]:
```
GaussianNB()
```

In [17]:
```
1 y_pred = clf.predict(x_test)
```

In [18]:
```
1 y_pred
```

Out[18]:
```
array([1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1], dtype=int64)
```

In [19]:
```
1 with open('model.pickle', 'wb') as f:
2     pickle.dump(clf, f)
```

In [ ]:
```
1
```

In [20]:
```
1 from sklearn.metrics import confusion_matrix
2 cm = confusion_matrix(y_test, y_pred)
```

In [21]:
```
1 cm
```

Out[21]:
```
array([[13,  1],
       [ 0, 26]], dtype=int64)
```

In [ ]:
```
1
```

In [ ]:
```
1
```