```
import pandas as pd
from sklearn import preprocessing
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from sklearn import metrics

df=pd.read_csv("/content/Cancer_DS.csv")
df
```

```
           id diagnosis  radius_mean  texture_mean  perimeter_mean
area_mean  \
0      842302         M        17.99         10.38          122.80
1001.0
1      842517         M        20.57         17.77          132.90
1326.0
2    84300903         M        19.69         21.25          130.00
1203.0
3    84348301         M        11.42         20.38           77.58
386.1
4    84358402         M        20.29         14.34          135.10
1297.0
..        ...       ...          ...           ...             ...
...
564    926424         M        21.56         22.39          142.00
1479.0
565    926682         M        20.13         28.25          131.20
1261.0
566    926954         M        16.60         28.08          108.30
858.1
567    927241         M        20.60         29.33          140.10
1265.0
568     92751         B         7.76         24.54           47.92
181.0

     smoothness_mean  compactness_mean  concavity_mean  concave
points_mean  \
0            0.11840           0.27760         0.30010
0.14710
1            0.08474           0.07864         0.08690
0.07017
2            0.10960           0.15990         0.19740
0.12790
3            0.14250           0.28390         0.24140
0.10520
4            0.10030           0.13280         0.19800
0.10430
..               ...               ...             ...
```

```
...
564          0.11100           0.11590         0.24390
0.13890
565          0.09780           0.10340         0.14400
0.09791
566          0.08455           0.10230         0.09251
0.05302
567          0.11780           0.27700         0.35140
0.15200
568          0.05263           0.04362         0.00000
0.00000

     ...  texture_worst  perimeter_worst  area_worst  smoothness_worst
\
0    ...          17.33           184.60      2019.0            0.16220

1    ...          23.41           158.80      1956.0            0.12380

2    ...          25.53           152.50      1709.0            0.14440

3    ...          26.50            98.87       567.7            0.20980

4    ...          16.67           152.20      1575.0            0.13740

..   ...            ...              ...         ...                ...

564  ...          26.40           166.10      2027.0            0.14100

565  ...          38.25           155.00      1731.0            0.11660

566  ...          34.12           126.70      1124.0            0.11390

567  ...          39.42           184.60      1821.0            0.16500

568  ...          30.37            59.16       268.6            0.08996


     compactness_worst  concavity_worst  concave points_worst
symmetry_worst  \
0              0.66560           0.7119                0.2654
0.4601
1              0.18660           0.2416                0.1860
0.2750
2              0.42450           0.4504                0.2430
0.3613
3              0.86630           0.6869                0.2575
0.6638
4              0.20500           0.4000                0.1625
0.2364
..                 ...              ...                   ...
```

```
...
564             0.21130           0.4107                 0.2216
0.2060
565             0.19220           0.3215                 0.1628
0.2572
566             0.30940           0.3403                 0.1418
0.2218
567             0.86810           0.9387                 0.2650
0.4087
568             0.06444           0.0000                 0.0000
0.2871

     fractal_dimension_worst  Unnamed: 32
0                    0.11890          NaN
1                    0.08902          NaN
2                    0.08758          NaN
3                    0.17300          NaN
4                    0.07678          NaN
..                       ...          ...
564                  0.07115          NaN
565                  0.06637          NaN
566                  0.07820          NaN
567                  0.12400          NaN
568                  0.07039          NaN

[569 rows x 33 columns]
```

```
df.isnull().sum()
```

```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
```

```
fractal_dimension_se          0
radius_worst                  0
texture_worst                 0
perimeter_worst               0
area_worst                    0
smoothness_worst              0
compactness_worst             0
concavity_worst               0
concave points_worst          0
symmetry_worst                0
fractal_dimension_worst       0
Unnamed: 32                 569
dtype: int64
```

```
df.drop(["id","Unnamed: 32"],axis=1,inplace=True)
```

```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()

df["diagnosis"] = label_encoder.fit_transform(df['diagnosis']) # 0 -
B, 1 - M
df['diagnosis']
```

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

```
x = df.drop(columns=['diagnosis'])
y = df['diagnosis']

x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.3,random_state=1)

classifier=GaussianNB()
model=GaussianNB()
model.fit(x_train,y_train)
```

```
GaussianNB()
```

```
y_pred=model.predict(x_test)

print('Accuracy:',metrics.accuracy_score(y_test,y_pred))
```

```
Accuracy: 0.9473684210526315
```

```
x_pred=model.predict(x_train)
print('Accuracy:',metrics.accuracy_score(y_train,x_pred))
```

```
Accuracy: 0.9396984924623115
```