

```
import numpy as np
import pandas as pd
df=pd.read_csv("/content/drive/MyDrive/Dataset_ML/breast-cancer (1).csv")
df
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.1184
1	842517	M	20.57	17.77	132.90	1326.0	0.0847
2	84300903	M	19.69	21.25	130.00	1203.0	0.1096
3	84348301	M	11.42	20.38	77.58	386.1	0.1425
4	84358402	M	20.29	14.34	135.10	1297.0	0.1003
...
564	926424	M	21.56	22.39	142.00	1479.0	0.1110
565	926682	M	20.13	28.25	131.20	1261.0	0.0978
566	926954	M	16.60	28.08	108.30	858.1	0.0845
567	927241	M	20.60	29.33	140.10	1265.0	0.1178
568	92751	B	7.76	24.54	47.92	181.0	0.0526

569 rows × 32 columns



```
df.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840
1	842517	M	20.57	17.77	132.90	1326.0	0.08474
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960
3	84348301	M	11.42	20.38	77.58	386.1	0.14250
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030

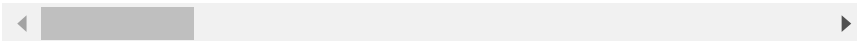
5 rows × 32 columns



```
df.tail()
```

	id	diagnosis	radius_mean	texture_mean	perimeter
564	926424	M	21.56	22.39	
565	926682	M	20.13	28.25	
566	926954	M	16.60	28.08	
567	927241	M	20.60	29.33	
568	92751	B	7.76	24.54	

5 rows × 32 columns



df.columns

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
      'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
      'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
      'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
      'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
      'fractal_dimension_se', 'radius_worst', 'texture_worst',
      'perimeter_worst', 'area_worst', 'smoothness_worst',
      'compactness_worst', 'concavity_worst', 'concave points_worst',
      'symmetry_worst', 'fractal_dimension_worst'],
      dtype='object')
```

```
df1=df.drop(['id'],axis=1)
df1
```

```
x=df1.iloc[:,1:]
x
```

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
0	17.99	10.38	122.80	1001.0	0.11840	0.27760
1	20.57	17.77	132.90	1326.0	0.08474	0.07864
2	19.69	21.25	130.00	1203.0	0.10960	0.15990
3	11.42	20.38	77.58	386.1	0.14250	0.28390
4	20.29	14.34	135.10	1297.0	0.10030	0.13280
...
564	21.56	22.39	142.00	1479.0	0.11100	0.11590
565	20.13	28.25	131.20	1261.0	0.09780	0.10340
566	16.60	28.08	108.30	858.1	0.08455	0.10230
567	20.60	29.33	140.10	1265.0	0.11780	0.27700
568	7.76	24.54	47.92	181.0	0.05263	0.04362

569 rows × 30 columns

```
y=df1.iloc[:,0].values
y
```

```
array([[M', M', M', M', M', M', M', M', M', M', M', M', M',  
       M', M', M', M', M', M', B', B', B', M', M', M', M',  
       M', M', M', M', M', M', M', M', B', M', B', B', B', B',  
       B', M', M', B', M', M', B', B', B', B', M', B', M',  
       M', B', B', B', B', B', M', B', M', M', B', M', B', M',  
       M', B', B', B', M', M', B', B', B', M', M', B', B', B',  
       B', M', B', B', M', M', B', B', B', B', B', B', B', B',  
       M', M', M', B', M', M', B', B', B', M', M', B', M',  
       B', M', M', B', M', M', B', B', M', B', B', M', B',  
       B', B', B', M', B', B', B', B', B', B', B', B', B', B',  
       M', B', B', M', M', B', B', B', B', B', B', B', B', B',  
       B', B', M', M', B', M', M', M', M', M', M', M', M', M',  
       M', M', M', M', M', M', M', B', B', B', B', B', B', M',  
       B', M', B', B', M', B', B', B', M', B', M', M', B', B',  
       B', B', B', B', B', B', B', B', B', B', B', B', M', B',  
       B', M', B', M', B', B', B', B', B', B', B', B', B', B',  
       B', B', B', B', B', B', M', B', B', B', M', B', B']])
```

```
'B', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'B', 'M', 'B', 'M',
'B', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'M', 'M', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'B', 'M', 'M', 'B', 'M', 'M', 'M', 'B', 'M', 'M', 'B', 'B', 'B',
'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'M',
'B', 'B', 'M', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B',
'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'M', 'B',
'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'B', 'M', 'B', 'M', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'M',
'B', 'B', 'M', 'B', 'M', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'B',
'B', 'B', 'B', 'B', 'B', 'M', 'M', 'B', 'B', 'B', 'B', 'B', 'B',
'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'B',
'B', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'M', 'B',
'B', 'B', 'B', 'B', 'M', 'M', 'B', 'M', 'B', 'M', 'B', 'B', 'B',
'B', 'B', 'M', 'B', 'B', 'M', 'B', 'M', 'B', 'M', 'M', 'B', 'B',
'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'M', 'B', 'M', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'B', 'B', 'B', 'M', 'M', 'M', 'M', 'M', 'B'], dtype=object)
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=42)
x_train
```

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
149	13.740	17.91	88.12	585.0	0.07944	0.06376
124	13.370	16.39	86.10	553.5	0.07115	0.07325
421	14.690	13.98	98.22	656.1	0.10310	0.18360
195	12.910	16.33	82.53	516.4	0.07941	0.05366
545	13.620	23.23	87.19	573.2	0.09246	0.06747
...
71	8.888	14.64	58.79	244.0	0.09783	0.15310
106	11.640	18.33	75.17	412.5	0.11420	0.10170
270	14.290	16.82	90.30	632.6	0.06429	0.02675
435	13.980	19.62	91.12	599.5	0.10600	0.11330
102	12.180	20.52	77.22	458.7	0.08013	0.04038

398 rows × 30 columns

```
x_test
```

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
204	12.470	18.60	81.09	481.9	0.09965	0.10580
70	18.940	21.31	123.60	1130.0	0.09009	0.10290
131	15.460	19.48	101.70	748.9	0.10920	0.12230
431	12.400	17.68	81.47	467.8	0.10540	0.13160
540	11.540	14.44	74.65	402.9	0.09984	0.11200
...
69	12.780	16.49	81.37	502.5	0.09831	0.05234
542	14.740	25.42	94.70	668.6	0.08275	0.07214
176	9.904	18.06	64.60	302.4	0.09699	0.12940
501	13.820	24.49	92.33	595.9	0.11620	0.16810

y_train

```
array(['B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'M',
      'M', 'M', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'M', 'B',
      'B', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'M',
      'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'M',
      'M', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'M',
      'M', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'B', 'M', 'B',
      'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'B', 'M', 'B', 'M',
      'B', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'M',
      'B', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'M',
      'B', 'B', 'M', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'B',
      'M', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'M', 'M', 'M', 'B', 'B',
      'B', 'B', 'B', 'B', 'B', 'B', 'M', 'M', 'B', 'B', 'M', 'B', 'B',
      'B', 'B', 'M', 'B', 'B', 'M', 'B', 'M', 'M', 'M', 'B', 'M', 'B',
      'B', 'B', 'M', 'B', 'M', 'M', 'B', 'M', 'M', 'B', 'B', 'M', 'B',
      'B', 'B', 'M', 'B', 'B', 'M', 'M', 'M', 'M', 'B', 'B', 'B', 'B',
      'B', 'B', 'B', 'B', 'B', 'B', 'M', 'M', 'M', 'M', 'B', 'B', 'B',
      'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B',
      'B', 'B', 'M', 'M', 'M', 'M', 'B', 'B', 'M', 'M', 'B', 'B', 'B',
      'M', 'M', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'M', 'B', 'B',
      'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B',
      'B', 'M', 'B', 'B', 'M', 'B', 'B', 'M', 'M', 'M', 'B', 'M', 'M',
      'B', 'M', 'B', 'B', 'B', 'B', 'B', 'M', 'B'], dtype=object)
```

y_test

```
array(['B', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'M', 'B',
      'M', 'B', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'M', 'B', 'B', 'B',
      'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'M',
      'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'M',
      'B', 'B', 'B', 'B', 'B', 'M', 'M', 'B', 'B', 'M', 'M', 'B', 'B',
      'B', 'M', 'M', 'B', 'B', 'M', 'M', 'B', 'M', 'B', 'B', 'B', 'M',
      'B', 'B', 'M', 'B', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'B', 'B', 'B',
      'B', 'B', 'B', 'B', 'B', 'M', 'M', 'B', 'M', 'M', 'B', 'M', 'M',
      'B', 'B', 'B', 'M', 'B', 'B', 'M', 'B', 'B', 'M', 'B', 'M', 'B',
      'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'M', 'B', 'B', 'M',
      'M', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'B',
      'B', 'M', 'B', 'M', 'M', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'M',
      'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B',
      'M', 'B'], dtype=object)
```

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(x_train)
x_train=scaler.fit_transform(x_train)
x_test=scaler.fit_transform(x_test)
x_test
```

```
array([[0.30094852, 0.39633558, 0.29206797, ..., 0.34879725, 0.37377388,
        0.19381928],
       [0.65568288, 0.52700096, 0.62493149, ..., 0.61477663, 0.25425916,
        0.04863957],
       [0.46488294, 0.43876567, 0.45344922, ..., 0.52027491, 0.32808467,
        0.14470944],
       ...,
       [0.16026098, 0.37029894, 0.1629473 , ..., 0.34054983, 0.27052142,
        0.38663084],
       [0.37496573, 0.68032787, 0.38007987, ..., 0.52268041, 0.53820341,
        0.400739  ],
       [0.3239761 , 0.17984571, 0.32229269, ..., 0.53642612, 0.2769747 ,
        0.39737991]])
```

```
x_train
```

```
array([[0.29624369, 0.27730808, 0.28381849, ..., 0.21839623, 0.15474078,
        0.12800949],
       [0.27812332, 0.22590463, 0.26940639, ..., 0.32576197, 0.09520993,
        0.18006104],
       [0.34276899, 0.14440311, 0.355879 , ..., 0.40203193, 0.24876799,
        0.31400475],
       ...,
       [0.32317939, 0.2404464 , 0.29937215, ..., 0.12093614, 0.17602996,
        0.05222109],
       [0.30799745, 0.33513696, 0.3052226 , ..., 0.66291727, 0.31815494,
        0.42777213],
       [0.21984426, 0.36557322, 0.20605023, ..., 0.2696299 , 0.22255076,
        0.11648016]])
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
k_model=KNeighborsClassifier(n_neighbors=7)
```

```

nb_model=GaussianNB()
sv_model=SVC()
lsb_model=[k_model,nb_model,sv_model]

```

```

from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
for i in lsb_model:
    print(i)
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    print("*****")
    print(classification_report(y_test,y_pred))
    print("*****")
    print("*****")
    print(confusion_matrix(y_test,y_pred))

```

```

KNeighborsClassifier(n_neighbors=7)
*****

```

	precision	recall	f1-score	support
B	0.99	0.95	0.97	108
M	0.93	0.98	0.95	63
accuracy			0.96	171
macro avg	0.96	0.97	0.96	171
weighted avg	0.97	0.96	0.97	171

```

*****
*****
[[103  5]
 [ 1 62]]
GaussianNB()
*****

```

	precision	recall	f1-score	support
B	1.00	0.73	0.84	108
M	0.68	1.00	0.81	63
accuracy			0.83	171
macro avg	0.84	0.87	0.83	171
weighted avg	0.88	0.83	0.83	171

```

*****
*****
[[79 29]
 [ 0 63]]
SVC()
*****

```

	precision	recall	f1-score	support
B	0.99	0.90	0.94	108
M	0.85	0.98	0.91	63
accuracy			0.93	171
macro avg	0.92	0.94	0.93	171
weighted avg	0.94	0.93	0.93	171

```

*****
*****

```

```
[[97 11]
 [ 1 62]]
```

✓ 0s completed at 12:56 PM

