

```
In [50]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import seaborn as sns
```

```
In [51]: data= pd.read_csv(r"C:\Users\mayur\data.csv")
data.head()
```

```
Out[51]:
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	co
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 × 33 columns

```
In [52]: data.shape
```

```
Out[52]: (569, 33)
```

```
In [53]: data.columns
```

```
Out[53]: 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', 'Unnamed: 32'],
dtype='object')
```

```
In [54]: data.isnull().sum()
```

```
Out[54]: 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
```

```
In [55]: data.drop('Unnamed: 32', axis = 1, inplace = True)
```

```
In [56]: data.head()
```

```
Out[56]:
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	co
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

```
In [57]: data.corr()
```

```
Out[57]:
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
id	1.000000	0.074626	0.099770	0.073159	0.096893	-0.000120
radius_mean	0.074626	1.000000	0.323782	0.997855	0.987357	0.000000
texture_mean	0.099770	0.323782	1.000000	0.329533	0.321086	-0.000000

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
perimeter_mean	0.073159	0.997855	0.329533	1.000000	0.986507	0.
area_mean	0.096893	0.987357	0.321086	0.986507	1.000000	0.
smoothness_mean	-0.012968	0.170581	-0.023389	0.207278	0.177028	1.
compactness_mean	0.000096	0.506124	0.236702	0.556936	0.498502	0.
concavity_mean	0.050080	0.676764	0.302418	0.716136	0.685983	0.
concave points_mean	0.044158	0.822529	0.293464	0.850977	0.823269	0.
symmetry_mean	-0.022114	0.147741	0.071401	0.183027	0.151293	0.
fractal_dimension_mean	-0.052511	-0.311631	-0.076437	-0.261477	-0.283110	0.
radius_se	0.143048	0.679090	0.275869	0.691765	0.732562	0.
texture_se	-0.007526	-0.097317	0.386358	-0.086761	-0.066280	0.
perimeter_se	0.137331	0.674172	0.281673	0.693135	0.726628	0.
area_se	0.177742	0.735864	0.259845	0.744983	0.800086	0.
smoothness_se	0.096781	-0.222600	0.006614	-0.202694	-0.166777	0.
compactness_se	0.033961	0.206000	0.191975	0.250744	0.212583	0.
concavity_se	0.055239	0.194204	0.143293	0.228082	0.207660	0.
concave points_se	0.078768	0.376169	0.163851	0.407217	0.372320	0.
symmetry_se	-0.017306	-0.104321	0.009127	-0.081629	-0.072497	0.
fractal_dimension_se	0.025725	-0.042641	0.054458	-0.005523	-0.019887	0.
radius_worst	0.082405	0.969539	0.352573	0.969476	0.962746	0.
texture_worst	0.064720	0.297008	0.912045	0.303038	0.287489	0.
perimeter_worst	0.079986	0.965137	0.358040	0.970387	0.959120	0.
area_worst	0.107187	0.941082	0.343546	0.941550	0.959213	0.
smoothness_worst	0.010338	0.119616	0.077503	0.150549	0.123523	0.
compactness_worst	-0.002968	0.413463	0.277830	0.455774	0.390410	0.
concavity_worst	0.023203	0.526911	0.301025	0.563879	0.512606	0.
concave points_worst	0.035174	0.744214	0.295316	0.771241	0.722017	0.
symmetry_worst	-0.044224	0.163953	0.105008	0.189115	0.143570	0.
fractal_dimension_worst	-0.029866	0.007066	0.119205	0.051019	0.003738	0.

31 rows × 31 columns

In [58]:

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

```
In [59]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

```
In [60]: print(len(x_train))
         print(len(y_train))
         print(len(x_test))
         print(len(y_test))
```

```
455
455
114
114
```

```
In [61]: scaler=StandardScaler()
         scaler.fit(x_train)
         x_train=scaler.transform(x_train)
         x_test=scaler.transform(x_test)
```

```
In [62]: classifier=KNeighborsClassifier(n_neighbors=4)
         classifier.fit(x_train,y_train)
```

```
Out[62]: KNeighborsClassifier(n_neighbors=4)
```

```
In [63]: y_p=classifier.predict(x_test)
         print(classification_report(y_test,y_p))
         print(confusion_matrix(y_test,y_p))
         print(classifier.score(x_test,y_test))
         print(accuracy_score(y_test,y_p)*100)
```

	precision	recall	f1-score	support
B	0.92	1.00	0.96	67
M	1.00	0.87	0.93	47
accuracy			0.95	114
macro avg	0.96	0.94	0.94	114
weighted avg	0.95	0.95	0.95	114

```
[[67  0]
 [ 6 41]]
0.9473684210526315
94.73684210526315
```