Detection of Breast Cancer using ML

@Author: Saurabh Bobde: 10 nag

```
import sklearn as SK
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import pandas
```

```
pwd
```

```
'E:\Classifiers\Machinelearning\Breast-cancer-Detection\nag-ML'
```

```
cd E:\Classifiers\Machinelearning\Breast-cancer-Detection\nag-ML
```

```
import os
ls -lrtsh *.csv
```

```
total 1M
 8K -rwxr-xr-x+  1 Saurabh  Saurabh  8K Jun 13 10:20 breast_cancer_training_NL.csv*
 8K -rwxr-xr-x+  1 Saurabh  Saurabh  8K Jun 13 10:20 breast_cancer_testing_NL.csv*
 1M -rwxr-xr-x+  1 Saurabh  Saurabh  1M Jun 13 10:20 data.csv*
```

```
df=pd.read_csv("data.csv")
```

```
df.head(5)
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790
3	84358402	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	843786	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430
5	844779	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.1578	0.08089
6	845636	M	18.25	19.98	119.60	1040.0	0.09463	0.10900	0.1127	0.07400

7 rows × 33 columns

```
df.shape
```

```
(569, 33)
```

```
#Printing number of empty 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
```

```
#drop all columns with missing values
df=df.dropna(axis=1)
```

```
df.shape
```

```
(569, 32)
```

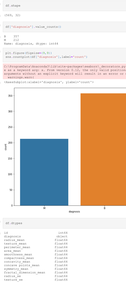
```
#Checking for diagnosis
df.value_counts()
```

```
diagnosis  value_counts
B          357             1
M          212             1
Name: count, dtype: int64
```

```
sns.figure(figsize=(5,5))
sns.countplot(df['diagnosis'],palette='husl')
```

```
C:\Users\saurabh\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without explicit keyword arguments will raise an error or misinterpretation.
  warnings.warn(f"Pass the following variable as a keyword arg: {x}. "
```

```
<seaborn.axisgrid.FacetGrid at 0x1f9f7ed8d00>,
<Figure size 432x432 with 1 Axes>
```




```
df.dtypes
```

```
id                         int64
diagnosis                 object
radius_mean               float64
texture_mean              float64
perimeter_mean            float64
area_mean                 float64
smoothness_mean           float64
compactness_mean          float64
concavity_mean            float64
concave points_mean       float64
symmetry_mean             float64
fractal_dimension_mean    float64
radius_se                 float64
texture_se                float64
perimeter_se              float64
area_se                   float64
smoothness_se             float64
compactness_se            float64
concavity_se              float64
concave points_se         float64
symmetry_se               float64
fractal_dimension_se      float64
radius_worst              float64
texture_worst             float64
perimeter_worst           float64
area_worst                float64
smoothness_worst          float64
compactness_worst         float64
concavity_worst           float64
concave points_worst      float64
symmetry_worst            float64
fractal_dimension_worst   float64
dtype: object
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

```
df.describe()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.0
std	125203.589467	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.0
min	8670000.000000	6.981000	9.710000	43.790000	143.500000	0.052630	0.019300	0.000000	0.0
25%	869218.000000	11.700000	16.170000	75.170000	420.300000	0.086670	0.064920	0.029560	0.0
50%	906024.000000	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.0
75%	8813129.000000	15.780000	21.450000	104.100000	782.700000	0.105300	0.130400	0.130700	0.0
max	91205750.000000	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.2

```
#Seperating the independent and dependent
from sklearn.preprocessing import LabelEncoder
labelencoder_X=LabelEncoder()
df[df.columns[1]]=labelencoder_X.fit_transform(df[df.columns[1]])
df.iloc[:,1:5].head()
```

```
array([[1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1]])
```

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
diagnosis	1.000000	0.730029	0.415180	0.742636	0.708984	0.358560	0.596534	0.696360	0.776614
radius_mean	0.730029	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822529
texture_mean	0.415180	0.323782	1.000000	0.329533	0.321086	0.023389	0.236702	0.302379	0.293464
perimeter_mean	0.742636	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716162	0.850977
area_mean	0.708984	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823269
smoothness_mean	0.358560	0.170581	0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553115
compactness_mean	0.596534	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.831135
concavity_mean	0.696360	0.676764	0.302379	0.716162	0.685983	0.521984	0.883121	1.000000	0.921391
concave points_mean	0.776614	0.822529	0.293464	0.850977	0.823269	0.553115	0.831135	0.921391	1.000000
symmetry_mean	0.330499	0.147369	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462497
fractal_dimension_mean	-0.012838	-0.07697	-0.076437	-0.261477	-0.283715	0.584792	0.565369	0.336783	0.166028

```
df.iloc[:,1:5].corr()
```

```
<ipykernel_jupyter_zmq_kernel_kernel_interface_Kernel>:1:
SyntaxWarning: invalid '\d' in regular expression
<ipykernel_jupyter_zmq_kernel_kernel_interface_Kernel>:1:
SyntaxWarning: invalid '\d' in regular expression
<ipykernel_jupyter_zmq_kernel_kernel_interface_Kernel>:1:
SyntaxWarning: invalid '\d' in regular expression
<ipykernel_jupyter_zmq_kernel_kernel_interface_Kernel>:1:
SyntaxWarning: invalid '\d' in regular expression
```

```
df.corr()
```

```
df.iloc[:,1:5].corr().T.T.corr().T
```

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
diagnosis	1	0.73	0.41	0.74	0.71	0.36	0.59	0.69	0.78
radius_mean	0.73	1	0.32	1	0.99	0.17	0.51	0.68	0.82
texture_mean	0.41	0.32	1	0.33	0.32	0.02	0.24	0.3	0.29
perimeter_mean	0.74	1	0.33	1	0.99	0.21	0.56	0.72	0.85
area_mean	0.71	0.99	0.32	0.99	1	0.18	0.5	0.69	0.82
smoothness_mean	0.36	0.17	0.02	0.21	0.18	1	0.66	0.52	0.55
compactness_mean	0.59	0.51	0.24	0.56	0.5	0.66	1	0.88	0.83
concavity_mean	0.69	0.68	0.3	0.72	0.69	0.52	0.88	1	0.92
concave points_mean	0.78	0.82	0.29	0.85	0.82	0.55	0.83	0.92	1
symmetry_mean	0.33	0.15	0.07	0.18	0.15	0.56	0.6	0.5	0.46
fractal_dimension_mean	-0.01	-0.08	-0.08	-0.26	-0.28	0.58	0.57	0.34	0.17

```
df.iloc[:,1:5].corr().T.T.corr().T.T.corr().T
```

```
<ipykernel_jupyter_zmq_kernel_kernel_interface_Kernel>:1:
SyntaxWarning: invalid '\d' in regular expression
```