NAME:-THATIKONDA AKSHAY

COLLEGE:-GURUNANAK TECHINCAL CAMPUS

YEAR OF STUDY:-2ND YEAER(E.C.E)

MAJOR PROJECT IS ON BREAST CANCER DATASET (EDA AND ACCURACY)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('https://raw.githubusercontent.com/Akshay0809/dataset/main/data.csv') #
df
```

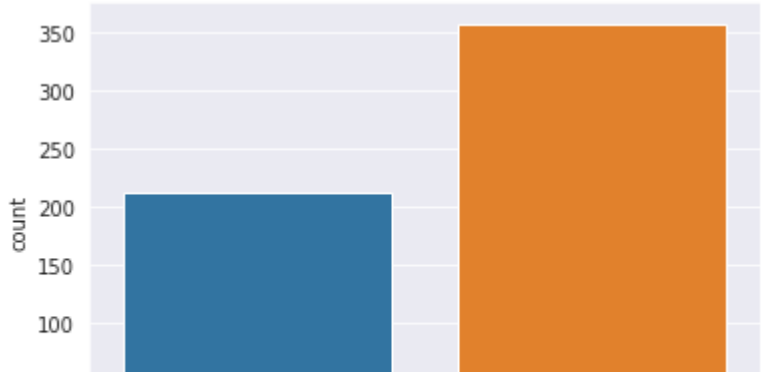
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	sm
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	
...	...	...	...	...	...	...	
564	926424	M	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	

569 rows × 33 columns

EDA (EXPLORATORY DATA ANALYSIS)

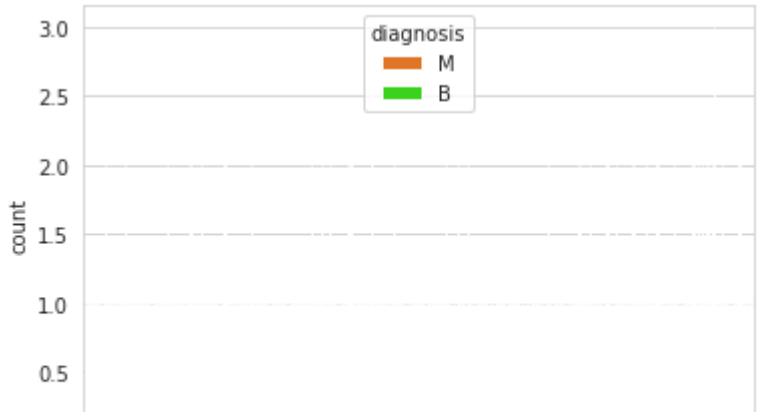
```
sns.set_style('darkgrid')
sns.countplot(x='diagnosis',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd347623110>
```



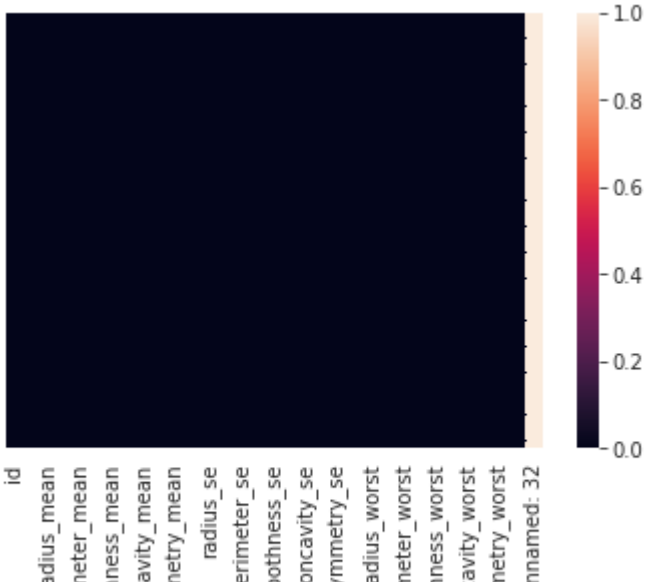

```
sns.set_style('whitegrid')
sns.countplot(x='perimeter_mean',hue='diagnosis',data=df,palette='gist_ncar_r')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd30d822cd0>
```




```
sns.heatmap(df.isnull(),yticklabels=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd3468f9790>
```

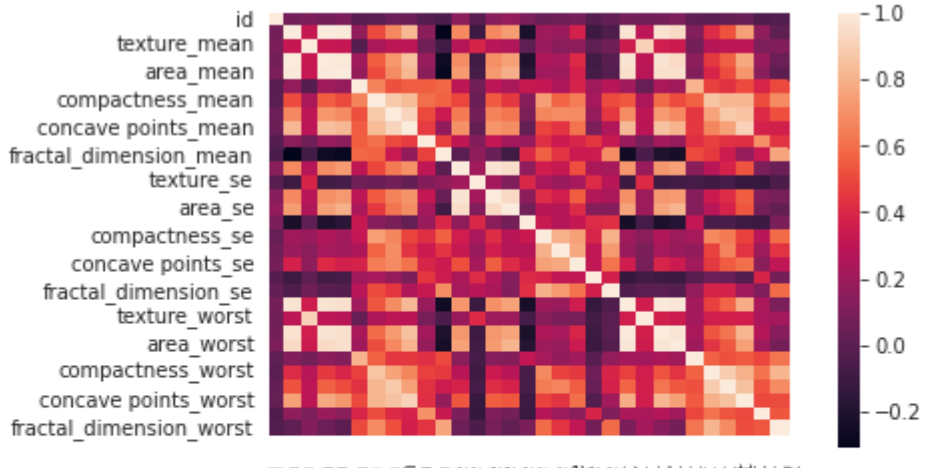



```
df['diagnosis'].hist(color='indigo',bins=60,figsize=(10,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd3406446d0>
```

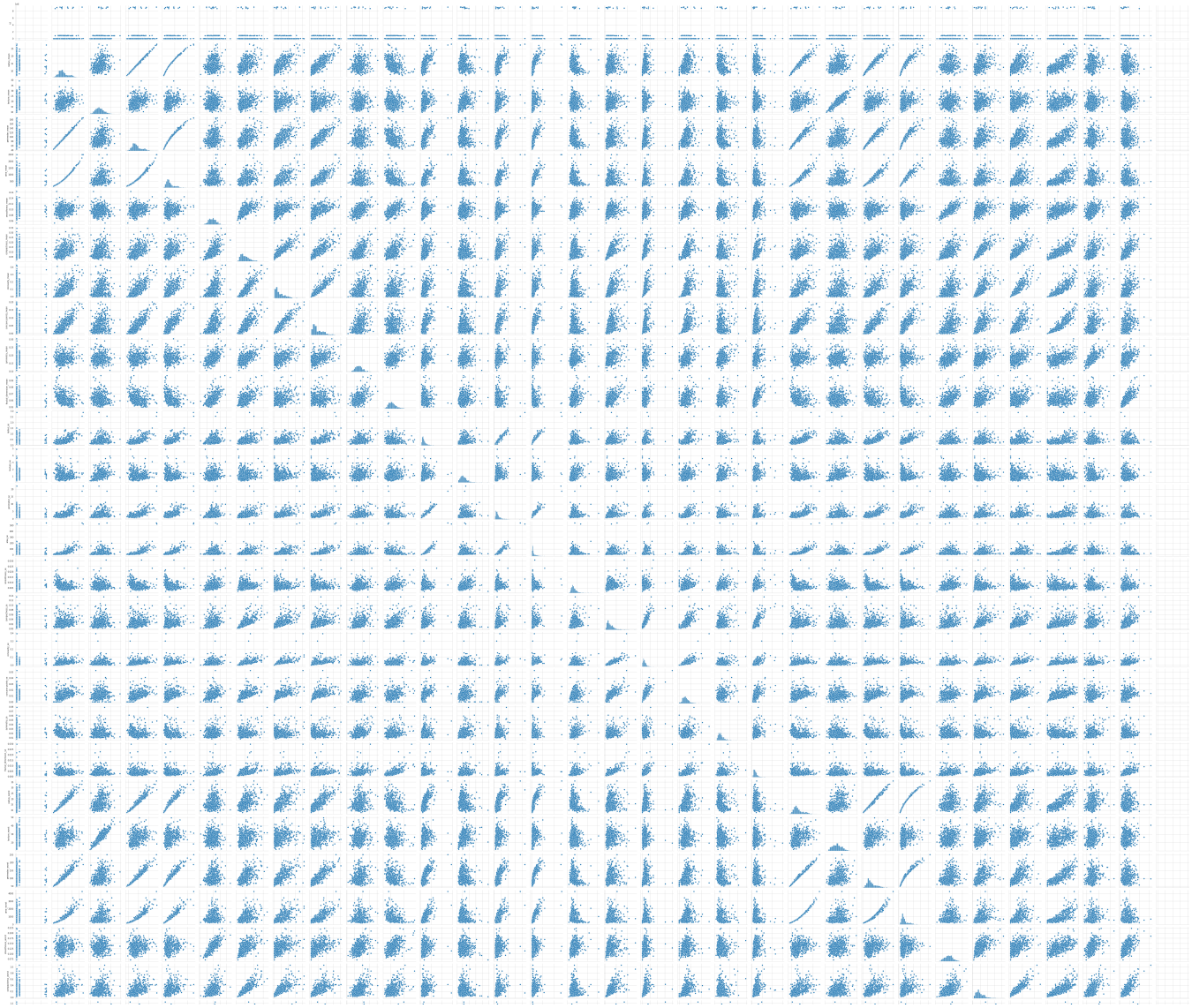

```
corelation=df.corr()
sns.heatmap(corelation,xticklabels=corelation.columns,annot=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd325445d50>
```



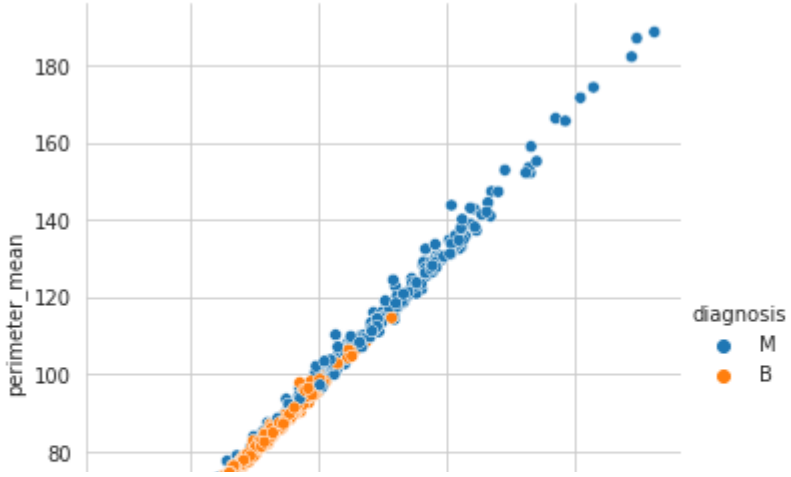

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7fd32534e2d0>
```



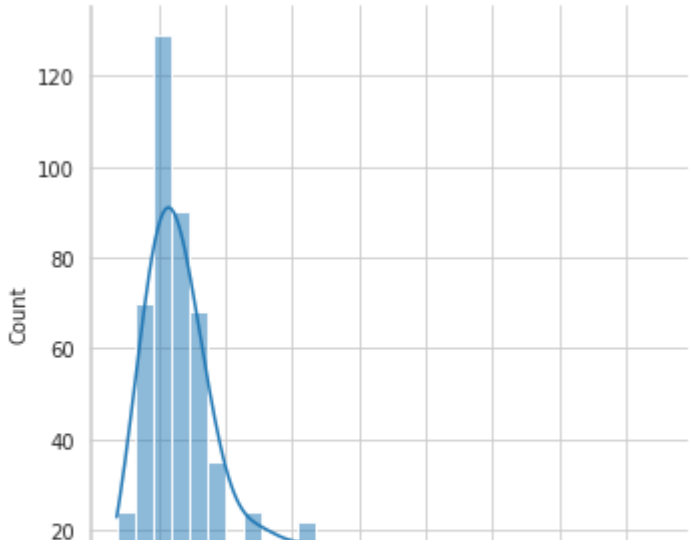
```
sns.relplot(x='radius_mean',y='perimeter_mean',hue='diagnosis',data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd31104bad0>
```



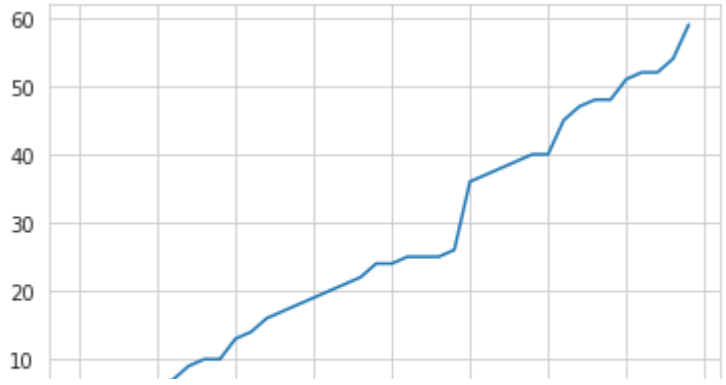
```
sns.displot(df['area_worst'],kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd30cf177d0>
```




```
np.random.seed(0)
x1=np.random.seed(0)
x1=np.random.randint(1,60,40)#in range of 1-60 we taking 40 elements randomly
x1=np.sort(x1)#arranging the elements in ascending order
plt.plot(x1)
```

```
[<matplotlib.lines.Line2D at 0x7fd30a5fec50>]
```



COVERTING STRING TO A NUMERICAL VALUE BY DUMMY VARIABLE

```
dummy=pd.get_dummies(df['diagnosis'])
```

```
dummy.head()  #TUMORS are two types 1ST:-BENIGN TUMOR DENOTED WITH B. ASSIGNED B VALU
              #2:-MALIGNANT TUMOR DENOTED BY M. ASSIGED M VALUE =1
```

	B	M
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1

```
df2=pd.concat((df,dummy),axis=1)
df2
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	sm
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	

569 rows × 35 columns

```
df3=df2.drop(['diagnosis'],axis=1)
df3
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	17.99	10.38	122.80	1001.0	0.118
1	842517	20.57	17.77	132.90	1326.0	0.084
2	84300903	19.69	21.25	130.00	1203.0	0.109
3	84348301	11.42	20.38	77.58	386.1	0.142
4	84358402	20.29	14.34	135.10	1297.0	0.100
...	...	...	...	...	...	
564	926424	21.56	22.39	142.00	1479.0	0.111
565	926682	20.13	28.25	131.20	1261.0	0.097
566	926954	16.60	28.08	108.30	858.1	0.084
567	927241	20.60	29.33	140.10	1265.0	0.117
568	92751	7.76	24.54	47.92	181.0	0.052

569 rows × 34 columns

```
df4=df3.drop(['B'],axis=1)
df4
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	17.99	10.38	122.80	1001.0	0.118

```
df5=df4.rename(columns={"M":"diagnosis"})
df5
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	17.99	10.38	122.80	1001.0	0.118
1	842517	20.57	17.77	132.90	1326.0	0.084
2	84300903	19.69	21.25	130.00	1203.0	0.109
3	84348301	11.42	20.38	77.58	386.1	0.142
4	84358402	20.29	14.34	135.10	1297.0	0.100
...	...	...	...	...	...	
564	926424	21.56	22.39	142.00	1479.0	0.111
565	926682	20.13	28.25	131.20	1261.0	0.097
566	926954	16.60	28.08	108.30	858.1	0.084
567	927241	20.60	29.33	140.10	1265.0	0.117
568	92751	7.76	24.54	47.92	181.0	0.052

569 rows × 33 columns

```
df5.shape
```

```
(569, 33)
```

```
df5.columns
```

```
Index(['id', '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',
       'diagnosis'],
      dtype='object')
```

```
df5.isnull().sum()
```

```
id                           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
diagnosis                    0
dtype: int64
```

```
df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   radius_mean              569 non-null    float64
 2   texture_mean             569 non-null    float64
 3   perimeter_mean           569 non-null    float64
 4   area_mean                569 non-null    float64
 5   smoothness_mean          569 non-null    float64
 6   compactness_mean         569 non-null    float64
 7   concavity_mean           569 non-null    float64
 8   concave points_mean      569 non-null    float64
```

```
 9   symmetry_mean            569 non-null    float64
 10  fractal_dimension_mean   569 non-null    float64
 11  radius_se                569 non-null    float64
 12  texture_se               569 non-null    float64
 13  perimeter_se             569 non-null    float64
 14  area_se                  569 non-null    float64
 15  smoothness_se            569 non-null    float64
 16  compactness_se           569 non-null    float64
 17  concavity_se             569 non-null    float64
 18  concave points_se        569 non-null    float64
 19  symmetry_se              569 non-null    float64
 20  fractal_dimension_se     569 non-null    float64
 21  radius_worst             569 non-null    float64
 22  texture_worst            569 non-null    float64
 23  perimeter_worst          569 non-null    float64
 24  area_worst               569 non-null    float64
 25  smoothness_worst         569 non-null    float64
 26  compactness_worst        569 non-null    float64
 27  concavity_worst          569 non-null    float64
 28  concave points_worst     569 non-null    float64
 29  symmetry_worst           569 non-null    float64
 30  fractal_dimension_worst  569 non-null    float64
 31  Unnamed: 32              0 non-null      float64
 32  diagnosis                569 non-null    uint8
dtypes: float64(31), int64(1), uint8(1)
memory usage: 142.9 KB
```

```
df5.describe()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothn
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	5
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	

```
df_num=df5.drop(['Unnamed: 32'],axis=1)
df_num
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	17.99	10.38	122.80	1001.0	0.118
1	842517	20.57	17.77	132.90	1326.0	0.084
2	84300903	19.69	21.25	130.00	1203.0	0.109
3	84348301	11.42	20.38	77.58	386.1	0.142
4	84358402	20.29	14.34	135.10	1297.0	0.100
...	...	...	...	...	...	
564	926424	21.56	22.39	142.00	1479.0	0.111
565	926682	20.13	28.25	131.20	1261.0	0.097
566	926954	16.60	28.08	108.30	858.1	0.084
567	927241	20.60	29.33	140.10	1265.0	0.117
568	92751	7.76	24.54	47.92	181.0	0.052

569 rows × 32 columns

```
df_num.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   radius_mean              569 non-null    float64
 2   texture_mean             569 non-null    float64
 3   perimeter_mean           569 non-null    float64
 4   area_mean                569 non-null    float64
 5   smoothness_mean          569 non-null    float64
 6   compactness_mean         569 non-null    float64
 7   concavity_mean           569 non-null    float64
 8   concave points_mean      569 non-null    float64
 9   symmetry_mean            569 non-null    float64
 10  fractal_dimension_mean   569 non-null    float64
 11  radius_se                569 non-null    float64
 12  texture_se               569 non-null    float64
 13  perimeter_se             569 non-null    float64
 14  area_se                  569 non-null    float64
 15  smoothness_se            569 non-null    float64
 16  compactness_se           569 non-null    float64
 17  concavity_se             569 non-null    float64
 18  concave points_se        569 non-null    float64
```

```
 19  symmetry_se              569 non-null    float64
 20  fractal_dimension_se     569 non-null    float64
 21  radius_worst             569 non-null    float64
 22  texture_worst            569 non-null    float64
 23  perimeter_worst          569 non-null    float64
 24  area_worst               569 non-null    float64
 25  smoothness_worst         569 non-null    float64
 26  compactness_worst        569 non-null    float64
 27  concavity_worst          569 non-null    float64
 28  concave points_worst     569 non-null    float64
 29  symmetry_worst           569 non-null    float64
 30  fractal_dimension_worst  569 non-null    float64
 31  diagnosis                569 non-null    uint8
dtypes: float64(30), int64(1), uint8(1)
memory usage: 138.5 KB
```

ASSIGNING INPUTS(X) FROM THE DATA

```
x=df_num.iloc[:,0:31].values
x
```

```
array([[8.4230200e+05, 1.7990000e+01, 1.0380000e+01, ..., 2.6540000e-01,
        4.6010000e-01, 1.1890000e-01],
       [8.4251700e+05, 2.0570000e+01, 1.7770000e+01, ..., 1.8600000e-01,
        2.7500000e-01, 8.9020000e-02],
       [8.4300903e+07, 1.9690000e+01, 2.1250000e+01, ..., 2.4300000e-01,
        3.6130000e-01, 8.7580000e-02],
       ...,
       [9.2695400e+05, 1.6600000e+01, 2.8080000e+01, ..., 1.4180000e-01,
        2.2180000e-01, 7.8200000e-02],
       [9.2724100e+05, 2.0600000e+01, 2.9330000e+01, ..., 2.6500000e-01,
        4.0870000e-01, 1.2400000e-01],
       [9.2751000e+04, 7.7600000e+00, 2.4540000e+01, ..., 0.0000000e+00,
        2.8710000e-01, 7.0390000e-02]])
```

ASSIGNING OUTPUT(Y) FROM THE GIVEN DATA

```
y=df_num.iloc[:,31].values
y
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
       0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1,
```

```
 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1,
 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1,
 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,
 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0,
 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,
 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0,
 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0],
dtype=uint8)
```

DATA SHOULD BE TRAIN,TEST AND SPLIT

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)  #HEl
```

```
x.shape
```

```
(569, 31)
```

```
x_train.shape
```

```
(455, 31)
```

```
x_test.shape
```

```
(114, 31)
```

```
y_train.shape
```

```
(455,)
```

```
y.shape
```

```
(569,)
```

```
#scaling/normalization
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
```

```
x_train=scaler.fit_transform(x_train)
x_test=scaler.fit_transform(x_test) #scaling should only done for inputs
```

```
#apply the classifier/regressor
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()

#fitting the model
model.fit(x_train,y_train)#mapping of inputs and outputs
```

```
    LogisticRegression()
```

FINDING ACCURACY SCORCE OF THE MODEL

```
from sklearn.metrics import accuracy_score
x_train_prediction=model.predict(x_train)
accuracy=accuracy_score(y_train,x_train_prediction)
```

```
accuracy*100
```

```
    96.92307692307692
```