BREAST CANCER CLASSIFICATION

Importing Libraries and Loading Dataset

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
```

In [2]:
```
df=pd.read_csv("/content/BREAST CANCER PREDICTION.csv")
df
```

Out[2]:

	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	0.30010	0.14710
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430
...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000

569 rows × 33 columns

In [3]:
```
df.head()
```

Out[3]:

	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	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430

5 rows × 33 columns

In [4]:
```
df.tail()
```

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000

5 rows × 33 columns

In [5]:
```
df.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   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
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

In [6]:
```
df.isna().sum()
```

Out[6]:

	0
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

	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 [7]:
```
df.dtypes
```

Out[7]:

	0
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

	0
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
Unnamed: 32	float64

dtype: object

Data Preprocessing

In [8]:
```
#Dropping unnecessaary features

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

#Dropping duplicates if there is any

df.drop_duplicates()
```

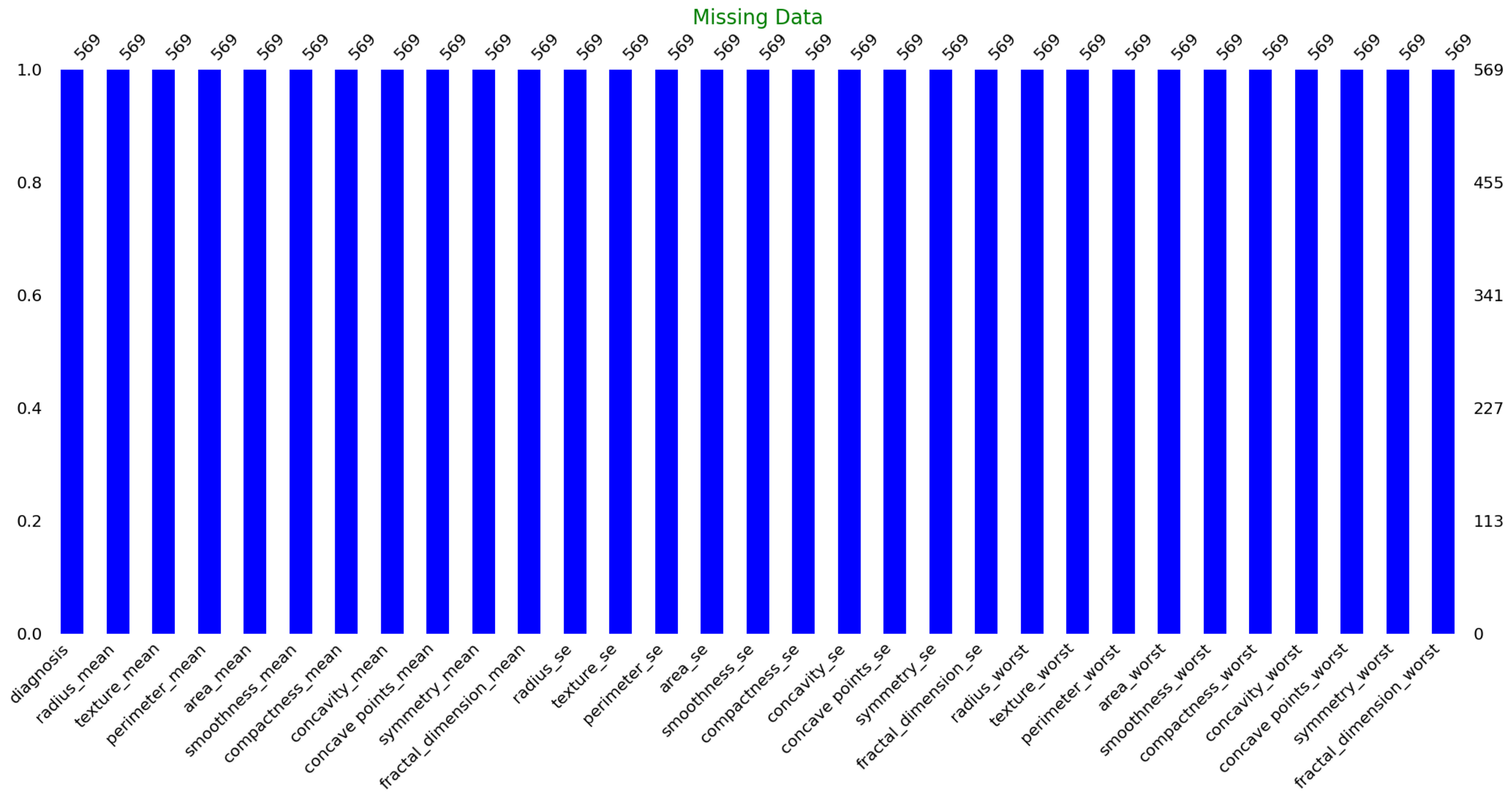
Out[8]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry
0	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	
1	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	
2	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	
3	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	
4	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	
...	...	...	...	...	...	...	...	...	...	
564	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	
565	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	
566	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	
567	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	
568	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	

569 rows × 31 columns

In [9]:

```
msno.bar(df,color='blue')
plt.title("Missing Data",color='green',fontsize=20)
plt.show()
```



In [10]:
```
df['diagnosis']=df['diagnosis'].map({"M":"Malignant","B":"Bengin"})
df
```

Out[10]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry
0	Malignant	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	
1	Malignant	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	
2	Malignant	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	
3	Malignant	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	
4	Malignant	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	
...	...	...	...	...	...	...	...	...	...	
564	Malignant	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	
565	Malignant	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	
566	Malignant	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	
567	Malignant	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	
568	Bengin	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	

569 rows × 31 columns

In [11]:
```
df['diagnosis'].value_counts()
```

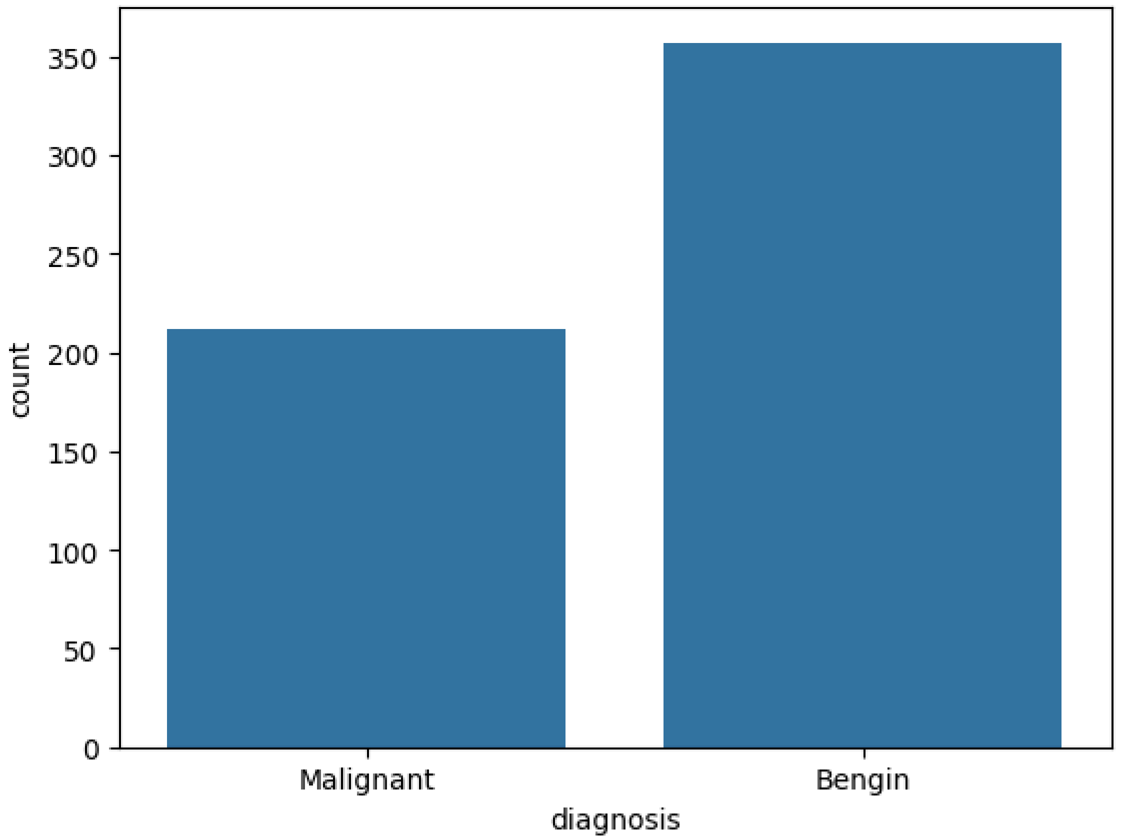
Out[11]:

diagnosis	count
Bengin	357
Malignant	212

dtype: int64

In [12]:
```
sns.countplot(x='diagnosis',data=df)
```

Out[12]:
```
<Axes: xlabel='diagnosis', ylabel='count'>
```



In [13]:
```
df1=df.drop(['diagnosis'],axis=1)
df1.corr()
```

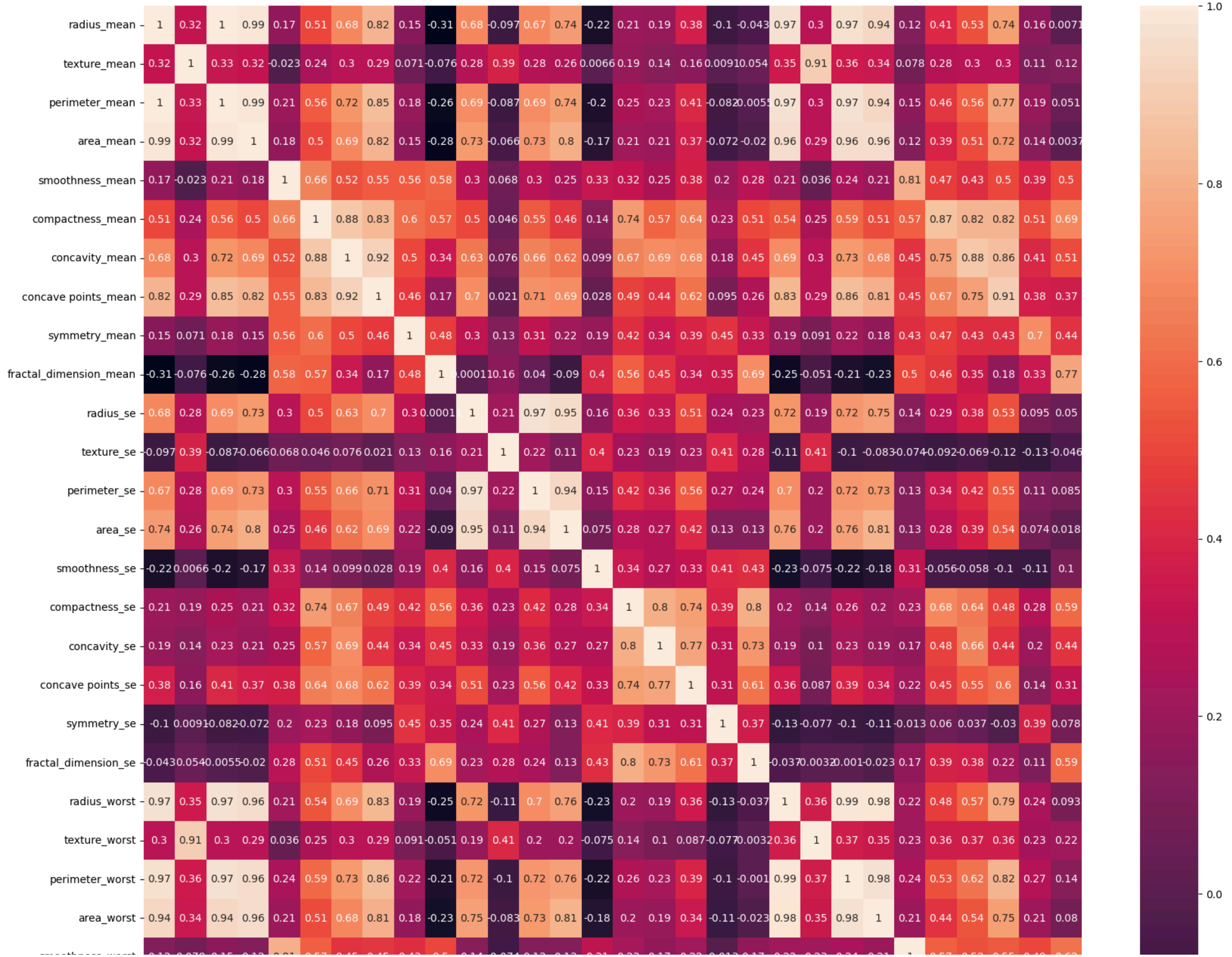
Out[13]:

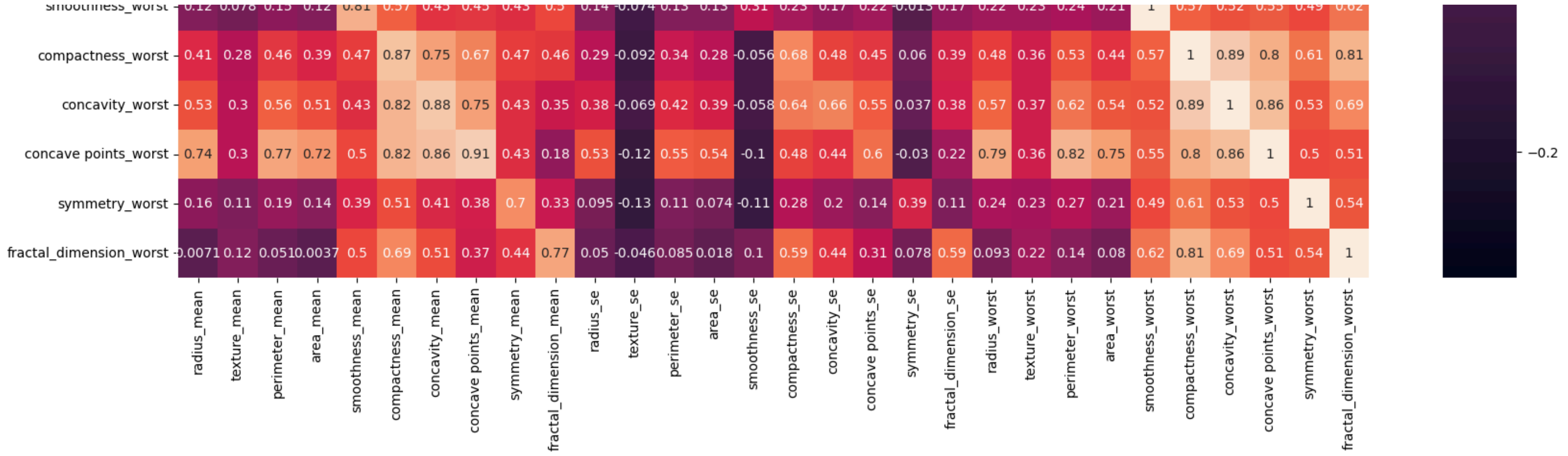
	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
radius_mean	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822529
texture_mean	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.293464
perimeter_mean	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.850977
area_mean	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823269
smoothness_mean	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553695
compactness_mean	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.831135
concavity_mean	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.921391
concave points_mean	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.000000
symmetry_mean	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462497
fractal_dimension_mean	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.166917
radius_se	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925	0.698050
texture_se	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218	0.021480
perimeter_se	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391	0.710650
area_se	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427	0.690299
smoothness_se	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564	0.027653
compactness_se	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	0.670279	0.490424
concavity_se	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	0.691270	0.439167
concave points_se	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	0.683260	0.615634
symmetry_se	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	0.178009	0.095351
fractal_dimension_se	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	0.449301	0.257584
radius_worst	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	0.688236	0.830318
texture_worst	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	0.299879	0.292752
perimeter_worst	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	0.729565	0.855923
area_worst	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	0.675987	0.809630

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
smoothness_worst	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	0.448822	0.452753
compactness_worst	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	0.754968	0.667454
concavity_worst	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	0.884103	0.752399
concave points_worst	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	0.861323	0.910155
symmetry_worst	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	0.409464	0.375744
fractal_dimension_worst	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	0.514930	0.368661

30 rows × 30 columns

```
In [14]: plt.figure(figsize=(20,20))
         sns.heatmap(df1.corr(),annot=True)
         plt.show()
```


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
1.0
0.8
0.6
0.4
0.2
0.0



In [15]:
```
x=df.drop('diagnosis',axis=1).values
x
```

Out[15]:
```
array([[1.799e+01, 1.038e+01, 1.228e+02, ..., 2.654e-01, 4.601e-01,
        1.189e-01],
       [2.057e+01, 1.777e+01, 1.329e+02, ..., 1.860e-01, 2.750e-01,
        8.902e-02],
       [1.969e+01, 2.125e+01, 1.300e+02, ..., 2.430e-01, 3.613e-01,
        8.758e-02],
       ...,
       [1.660e+01, 2.808e+01, 1.083e+02, ..., 1.418e-01, 2.218e-01,
        7.820e-02],
       [2.060e+01, 2.933e+01, 1.401e+02, ..., 2.650e-01, 4.087e-01,
        1.240e-01],
       [7.760e+00, 2.454e+01, 4.792e+01, ..., 0.000e+00, 2.871e-01,
        7.039e-02]])
```

In [16]:
```
y=df['diagnosis'].values
y
```

Out[16]:
```
array(['Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Bengin'], dtype=object)
```

In [17]:
```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

Out[17]:
```
array([[1.374e+01, 1.791e+01, 8.812e+01, ..., 6.019e-02, 2.350e-01,
        7.014e-02],
       [1.337e+01, 1.639e+01, 8.610e+01, ..., 8.978e-02, 2.048e-01,
        7.628e-02],
       [1.469e+01, 1.398e+01, 9.822e+01, ..., 1.108e-01, 2.827e-01,
        9.208e-02],
       ...,
       [1.429e+01, 1.682e+01, 9.030e+01, ..., 3.333e-02, 2.458e-01,
        6.120e-02],
       [1.398e+01, 1.962e+01, 9.112e+01, ..., 1.827e-01, 3.179e-01,
        1.055e-01],
       [1.218e+01, 2.052e+01, 7.722e+01, ..., 7.431e-02, 2.694e-01,
        6.878e-02]])
```

In [18]:
```python
x_test
```

Out[18]:
```
array([[1.247e+01, 1.860e+01, 8.109e+01, ..., 1.015e-01, 3.014e-01,
         8.750e-02],
        [1.894e+01, 2.131e+01, 1.236e+02, ..., 1.789e-01, 2.551e-01,
         6.589e-02],
        [1.546e+01, 1.948e+01, 1.017e+02, ..., 1.514e-01, 2.837e-01,
         8.019e-02],
        ...,
        [9.904e+00, 1.806e+01, 6.460e+01, ..., 9.910e-02, 2.614e-01,
         1.162e-01],
        [1.382e+01, 2.449e+01, 9.233e+01, ..., 1.521e-01, 3.651e-01,
         1.183e-01],
        [1.289e+01, 1.411e+01, 8.495e+01, ..., 1.561e-01, 2.639e-01,
         1.178e-01]])
```

In [19]:
```
y_train
```

Out[19]:
```
array(['Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Malignant', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Malignant', 'Malignant', 'Malignant', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Malignant', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
       'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
       'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
       'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
       'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Malignant', 'Malignant', 'Malignant', 'Bengin',
       'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Bengin',
       'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin'], dtype=object)
```

In [20]:

```
y_test
```

```
Out[20]: array(['Bengin', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
                'Malignant', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
                'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Bengin',
                'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant',
                'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
                'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
                'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
                'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
                'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
                'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
                'Bengin', 'Malignant', 'Malignant', 'Bengin', 'Bengin',
                'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
                'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Malignant',
                'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
                'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
                'Malignant', 'Malignant', 'Malignant', 'Malignant', 'Malignant',
                'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Bengin',
                'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Malignant', 'Bengin',
                'Malignant', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
                'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Bengin',
                'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
                'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
                'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant',
                'Malignant', 'Bengin', 'Bengin', 'Malignant', 'Malignant',
                'Malignant', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
                'Bengin', 'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Bengin',
                'Bengin', 'Malignant', 'Bengin', 'Malignant', 'Malignant',
                'Malignant', 'Bengin', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
                'Bengin', 'Malignant', 'Malignant', 'Bengin', 'Bengin', 'Bengin',
                'Bengin', 'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin',
                'Bengin', 'Bengin', 'Bengin', 'Malignant', 'Bengin'], dtype=object)
```

Normalization

```
In [21]: from sklearn.preprocessing import StandardScaler
         scaler=StandardScaler()
         scaler.fit(x_train)
         x_train=scaler.transform(x_train)
         x_test=scaler.transform(x_test)
```

Model Creation and Performance Evaluation

In [22]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,classification_report
knn=KNeighborsClassifier(n_neighbors=7)
dectree=DecisionTreeClassifier(random_state=42)
ranfor=RandomForestClassifier(n_estimators=100,random_state=42)
logreg=LogisticRegression()
lst=[knn,dectree,ranfor,logreg]
```

In [23]:
```
for i in lst:
 i.fit(x_train,y_train)
 y_pred=i.predict(x_test)
 print("Test Accuracy of",i,"model is",accuracy_score(y_test,y_pred))
 print(classification_report(y_test,y_pred))
```

```
Test Accuracy of KNeighborsClassifier(n_neighbors=7) model is 0.9590643274853801
              precision    recall  f1-score   support

      Bengin       0.96      0.97      0.97       108
   Malignant       0.95      0.94      0.94        63

    accuracy                           0.96       171
   macro avg       0.96      0.95      0.96       171
weighted avg       0.96      0.96      0.96       171

Test Accuracy of DecisionTreeClassifier(random_state=42) model is 0.9415204678362573
              precision    recall  f1-score   support

      Bengin       0.97      0.94      0.95       108
   Malignant       0.90      0.95      0.92        63

    accuracy                           0.94       171
   macro avg       0.93      0.94      0.94       171
weighted avg       0.94      0.94      0.94       171

Test Accuracy of RandomForestClassifier(random_state=42) model is 0.9707602339181286
              precision    recall  f1-score   support

      Bengin       0.96      0.99      0.98       108
   Malignant       0.98      0.94      0.96        63

    accuracy                           0.97       171
   macro avg       0.97      0.96      0.97       171
weighted avg       0.97      0.97      0.97       171

Test Accuracy of LogisticRegression() model is 0.9824561403508771
              precision    recall  f1-score   support

      Bengin       0.99      0.98      0.99       108
   Malignant       0.97      0.98      0.98        63

    accuracy                           0.98       171
   macro avg       0.98      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```