

# HEAMNATH

## 20104028

In [8]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [9]:

```
df=pd.read_csv("8_BreastCancerPrediction.csv")
df
```

Out[9]:

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>...</b>
<b>0</b>	842302	M	17.99	10.38	122.80	1001.0	0.11840	
<b>1</b>	842517	M	20.57	17.77	132.90	1326.0	0.08474	
<b>2</b>	84300903	M	19.69	21.25	130.00	1203.0	0.10960	
<b>3</b>	84348301	M	11.42	20.38	77.58	386.1	0.14250	
<b>4</b>	84358402	M	20.29	14.34	135.10	1297.0	0.10030	
...	...	...	...	...	...	...	...	...
<b>564</b>	926424	M	21.56	22.39	142.00	1479.0	0.11100	
<b>565</b>	926682	M	20.13	28.25	131.20	1261.0	0.09780	
<b>566</b>	926954	M	16.60	28.08	108.30	858.1	0.08455	
<b>567</b>	927241	M	20.60	29.33	140.10	1265.0	0.11780	
<b>568</b>	92751	B	7.76	24.54	47.92	181.0	0.05263	

569 rows × 33 columns



In [3]:

```
df.head()
```

Out[3]:

	<b>Cst_Cnt</b>	<b>Btl_Cnt</b>	<b>Sta_ID</b>	<b>Depth_ID</b>	<b>Depthm</b>	<b>T_degC</b>	<b>Salnty</b>	<b>O2ml_L</b>	<b>STheta</b>	<b>O2Sat</b>	<b>...</b>	<b>R_PHAE0</b>
<b>0</b>	1	1	054.0	19- 4903CR- HY-060- 0930-	0	10.50	33.440	NaN	25.649	NaN	...	NaN

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHAEO
1	1	2	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0008A-3	8	10.46	33.440	NaN	25.656	NaN	...	NaN
2	1	3	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0010A-7	10	10.46	33.437	NaN	25.654	NaN	...	NaN
3	1	4	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0019A-3	19	10.45	33.420	NaN	25.643	NaN	...	NaN
4	1	5	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0020A-7	20	10.45	33.421	NaN	25.643	NaN	...	NaN

5 rows × 74 columns

## DATA CLEANING AND DATA PREPROCESSING

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 864863 entries, 0 to 864862
Data columns (total 74 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Cst_Cnt          864863 non-null   int64  
 1   Btl_Cnt          864863 non-null   int64  
 2   Sta_ID           864863 non-null   object 
 3   Depth_ID         864863 non-null   object 
 4   Depthm           864863 non-null   int64  
 5   T_degC           853900 non-null   float64 
 6   Salnty           817509 non-null   float64 
 7   O2ml_L            696201 non-null   float64 
 8   STheta            812174 non-null   float64 
 9   O2Sat             661274 non-null   float64 
 10  Oxy_µmol/Kg      661268 non-null   float64 
 11  BtlNum            118667 non-null   float64 
 12  RecInd            864863 non-null   int64  
 13  T_prec            853900 non-null   float64 
 14  T_qual            23127 non-null    float64
```

```

15 S_prec           817509 non-null float64
16 S_qual          74914 non-null float64
17 P_qual          673755 non-null float64
18 O_qual          184676 non-null float64
19 SThtaq          65823 non-null float64
20 O2Satq          217797 non-null float64
21 ChlorA          225272 non-null float64
22 Chlqua          639166 non-null float64
23 Phaeop          225271 non-null float64
24 Phaqua          639170 non-null float64
25 PO4uM           413317 non-null float64
26 PO4q            451786 non-null float64
27 SiO3uM          354091 non-null float64
28 SiO3qu          510866 non-null float64
29 NO2uM           337576 non-null float64
30 NO2q            529474 non-null float64
31 NO3uM           337403 non-null float64
32 NO3q            529933 non-null float64
33 NH3uM           64962 non-null float64
34 NH3q            808299 non-null float64
35 C14As1          14432 non-null float64
36 C14A1p          12760 non-null float64
37 C14A1q          848605 non-null float64
38 C14As2          14414 non-null float64
39 C14A2p          12742 non-null float64
40 C14A2q          848623 non-null float64
41 DarkAs          22649 non-null float64
42 DarkAp          20457 non-null float64
43 DarkAq          840440 non-null float64
44 MeanAs          22650 non-null float64
45 MeanAp          20457 non-null float64
46 MeanAq          840439 non-null float64
47 IncTim          14437 non-null object
48 LightP           18651 non-null float64
49 R_Depth          864863 non-null float64
50 R_TEMP           853900 non-null float64
51 R_POTEMP         818816 non-null float64
52 R_SALINITY        817509 non-null float64
53 R_SIGMA          812007 non-null float64
54 R_SVA            812092 non-null float64
55 R_DYNHT          818206 non-null float64
56 R_O2              696201 non-null float64
57 R_O2Sat          666448 non-null float64
58 R_SI03           354099 non-null float64
59 R_PO4             413325 non-null float64
60 R_NO3             337411 non-null float64
61 R_NO2             337584 non-null float64
62 R_NH4             64982 non-null float64
63 R_CHLA            225276 non-null float64
64 R_PHAE0            225275 non-null float64
65 R_PRES            864863 non-null int64
66 R_SAMP            122006 non-null float64
67 DIC1              1999 non-null float64
68 DIC2              224 non-null float64
69 TA1                2084 non-null float64
70 TA2                234 non-null float64
71 pH2                10 non-null float64
72 pH1                84 non-null float64
73 DIC Quality Comment 55 non-null object
dtypes: float64(65), int64(5), object(4)
memory usage: 488.3+ MB

```

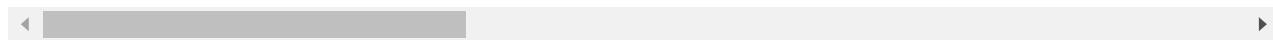
In [5]:

`df.describe()`

Out[5]:

	Cst_Cnt	Btl_Cnt	Depthm	T_degC	Salnty	O2ml_L	
<b>count</b>	864863.000000	864863.000000	864863.000000	853900.000000	817509.000000	696201.000000	81217
<b>mean</b>	17138.790958	432432.000000	226.831951	10.799677	33.840350	3.392468	2
<b>std</b>	10240.949817	249664.587267	316.050259	4.243825	0.461843	2.073256	
<b>min</b>	1.000000	1.000000	0.000000	1.440000	28.431000	-0.010000	2
<b>25%</b>	8269.000000	216216.500000	46.000000	7.680000	33.488000	1.360000	2
<b>50%</b>	16848.000000	432432.000000	125.000000	10.060000	33.863000	3.440000	2
<b>75%</b>	26557.000000	648647.500000	300.000000	13.880000	34.196900	5.500000	2
<b>max</b>	34404.000000	864863.000000	5351.000000	31.140000	37.034000	11.130000	25

8 rows × 70 columns



In [6]:

df.columns

```
Out[6]: Index(['Cst_Cnt', 'Btl_Cnt', 'Sta_ID', 'Depth_ID', 'Depthm', 'T_degC',
       'Salnty', 'O2ml_L', 'STheta', 'O2Sat', 'Oxy_µmol/Kg', 'BtlNum',
       'RecInd', 'T_prec', 'T_qual', 'S_prec', 'S_qual', 'P_qual', 'O_qual',
       'SThtaq', 'O2Satq', 'ChlorA', 'Chlqua', 'Phaeop', 'Phaqua', 'PO4uM',
       'PO4q', 'SiO3uM', 'SiO3qu', 'NO2uM', 'NO2q', 'NO3uM', 'NO3q', 'NH3uM',
       'NH3q', 'C14As1', 'C14A1p', 'C14A1q', 'C14As2', 'C14A2p', 'C14A2q',
       'DarkAs', 'DarkAp', 'DarkAq', 'MeanAs', 'MeanAp', 'MeanAq', 'IncTim',
       'LightP', 'R_Depth', 'R_TEMP', 'R_POTEMP', 'R_SALINITY', 'R_SIGMA',
       'R_SVA', 'R_DYNHT', 'R_O2', 'R_O2Sat', 'R_SI03', 'R_PO4', 'R_NO3',
       'R_NO2', 'R_NH4', 'R_CHLA', 'R_PHAE0', 'R_PRES', 'R_SAMP', 'DIC1',
       'DIC2', 'TA1', 'TA2', 'pH2', 'pH1', 'DIC Quality Comment'],
      dtype='object')
```

In [10]:

df1=df.dropna(axis=1)  
df1

Out[10]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	
<b>0</b>	842302	M	17.99	10.38	122.80	1001.0	0.11840	
<b>1</b>	842517	M	20.57	17.77	132.90	1326.0	0.08474	
<b>2</b>	84300903	M	19.69	21.25	130.00	1203.0	0.10960	
<b>3</b>	84348301	M	11.42	20.38	77.58	386.1	0.14250	
<b>4</b>	84358402	M	20.29	14.34	135.10	1297.0	0.10030	
...	...	...	...	...	...	...	...	...
<b>564</b>	926424	M	21.56	22.39	142.00	1479.0	0.11100	
<b>565</b>	926682	M	20.13	28.25	131.20	1261.0	0.09780	
<b>566</b>	926954	M	16.60	28.08	108.30	858.1	0.08455	
<b>567</b>	927241	M	20.60	29.33	140.10	1265.0	0.11780	

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>compactness_mean</b>	<b>concavity_mean</b>	<b>concave points_mean</b>	<b>symmetry_mean</b>	<b>fractal_dimension_mean</b>	<b>radius_se</b>	<b>texture_se</b>	<b>perimeter_se</b>	<b>area_se</b>	<b>smoothness_se</b>	<b>compactness_se</b>	<b>concavity_se</b>	<b>concave points_se</b>	<b>symmetry_se</b>	<b>fractal_dimension_se</b>	<b>radius_worst</b>	<b>texture_worst</b>	<b>perimeter_worst</b>	<b>area_worst</b>	<b>smoothness_worst</b>	<b>compactness_worst</b>	<b>concavity_worst</b>	<b>concave points_worst</b>	<b>symmetry_worst</b>	<b>fractal_dimension_worst</b>
568	92751	B	7.76	24.54	47.92	181.0	0.05263																									

569 rows × 32 columns

In [11]: df1.columns

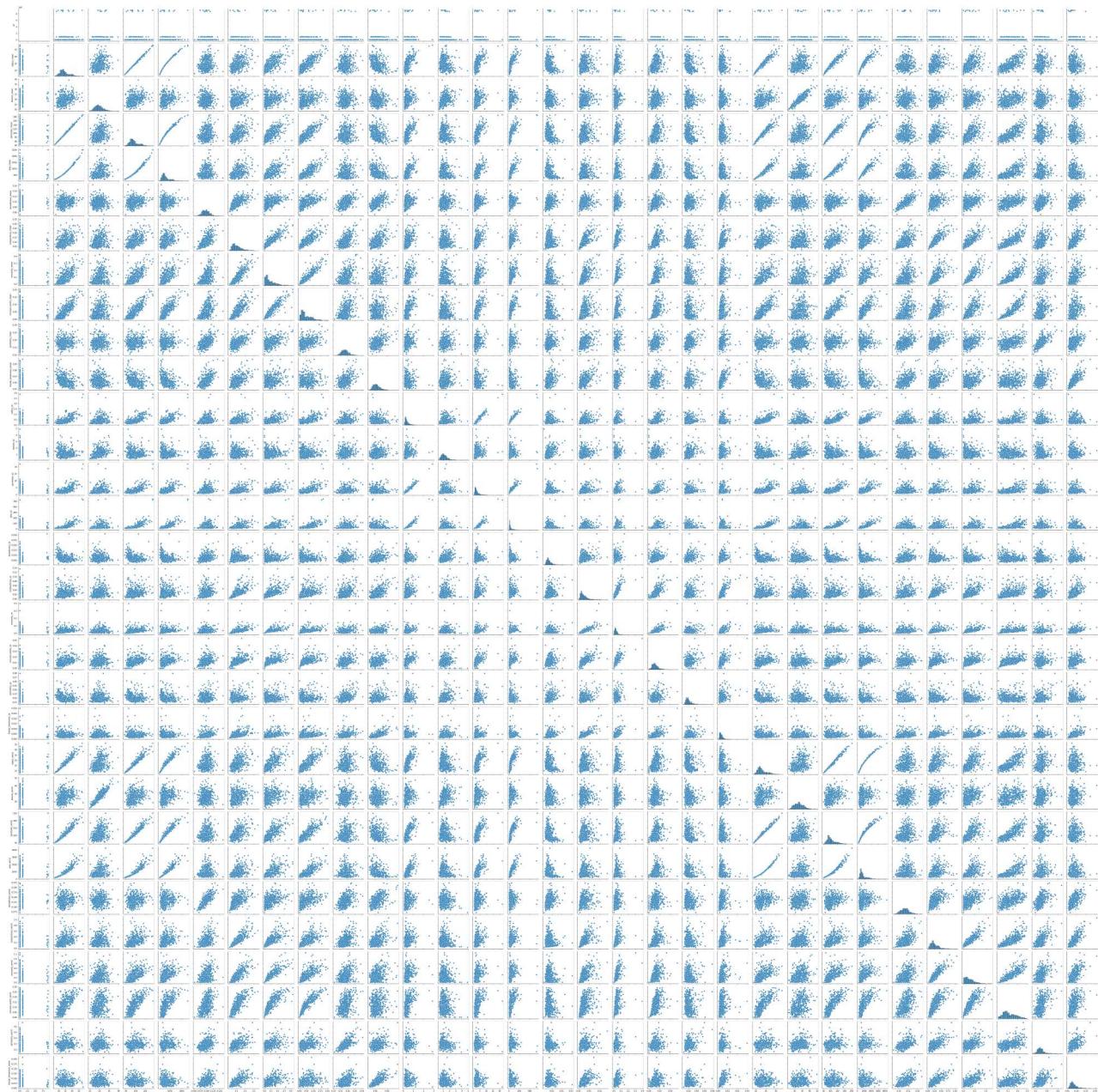
```
Out[11]: Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       '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')
```

```
In [12]: df1=df1[['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']]
```

## EDA AND VISUALIZATION

In [13]: sns.pairplot(df1)

Out[13]: &lt;seaborn.axisgrid.PairGrid at 0x1bf2a25ac10&gt;

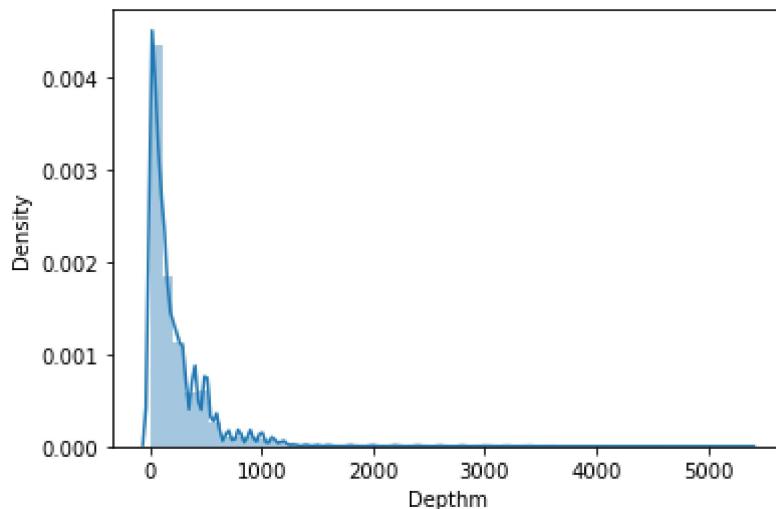


```
In [10]: sns.distplot(df1['Depthm'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:  
`distplot` is a deprecated function and will be removed in a future version. Please adapt  
your code to use either `displot` (a figure-level function with similar flexibility)  
or `histplot` (an axes-level function for histograms).
```

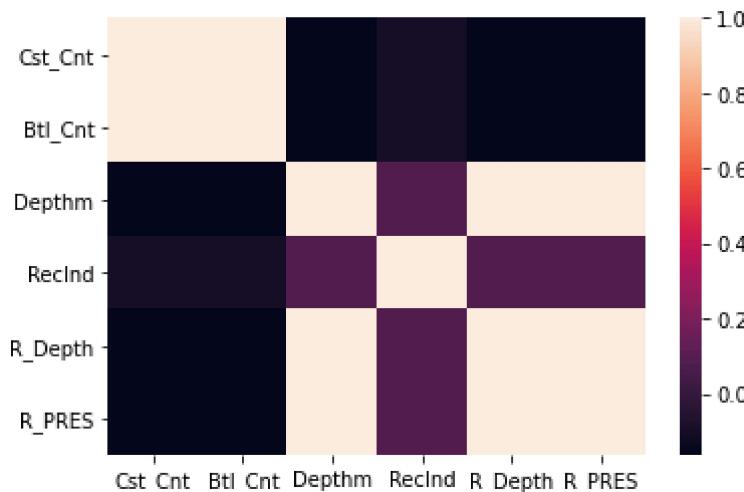
```
    warnings.warn(msg, FutureWarning)
```

```
Out[10]: <AxesSubplot:xlabel='Depthm', ylabel='Density'>
```



```
In [11]: sns.heatmap(df1.corr())
```

```
Out[11]: <AxesSubplot:>
```



## TO TRAIN THE MODEL AND MODEL BUILDING

```
In [12]: x=df[['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']]
y=df['fractal_dimension_worst']
```

```
In [13]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
In [14]: from sklearn.linear_model import LinearRegression  
lr=LinearRegression()  
lr.fit(x_train,y_train)
```

```
Out[14]: LinearRegression()
```

```
In [15]: lr.intercept_
```

```
Out[15]: -1.045477565149156
```

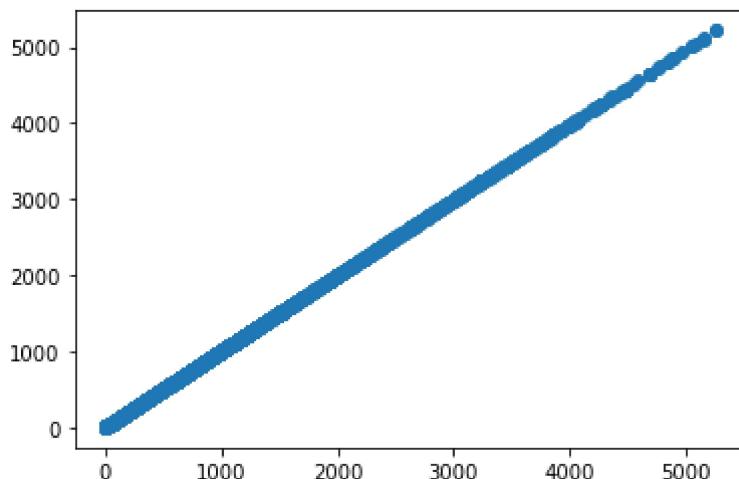
```
In [16]: coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])  
coeff
```

```
Out[16]: Co-efficient
```

Cst_Cnt	-0.000165
Btl_Cnt	0.000007
Depthm	-0.671700
Reclnd	-0.019142
R_Depth	1.682533

```
In [17]: prediction =lr.predict(x_test)  
plt.scatter(y_test,prediction)
```

```
Out[17]: <matplotlib.collections.PathCollection at 0x230070ef100>
```



## ACCURACY

```
In [18]: lr.score(x_test,y_test)
```

```
Out[18]: 0.9999882034055533
```