

Breast Cancer Data Set

To build the model to classify the Breast Cancer of Wisconsin (Original) Data Set

In [31]:

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

In [3]:

```
path = r'C:\Users\AYANMAN\Desktop\Python with anaconda\crampete data science\datas\canc
```

In [7]:

```
df = pd.read_csv(path)
```

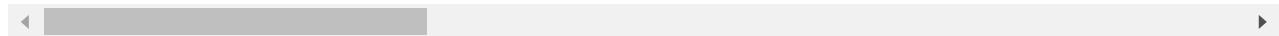
In [9]:

```
df.head()
```

Out[9]:

	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
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.12030	0.08550	0.05690	0.04900	0.03570	1.3890	3.0020	6.0050	1.4600	0.07110	0.02950	0.00850	0.00350	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.09580	0.05370	0.04900	0.03570	0.02770	1.4900	3.1400	6.1900	1.4900	0.07450	0.03000	0.00850	0.00350	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050		
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.11840	0.05370	0.04900	0.03570	0.02770	1.4900	3.1400	6.1900	1.4900	0.07450	0.03000	0.00850	0.00350	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050		
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.14900	0.05370	0.04900	0.03570	0.02770	1.4900	3.1400	6.1900	1.4900	0.07450	0.03000	0.00850	0.00350	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050		
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.10960	0.05370	0.04900	0.03570	0.02770	1.4900	3.1400	6.1900	1.4900	0.07450	0.03000	0.00850	0.00350	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050	0.00050		

5 rows × 32 columns



In [10]:

```
df.isnull().sum()
```

Out[10]:

```
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
```

```
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
dtype: int64
```

In [11]: `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
```

In [12]: `df = df.drop(columns=['id'])`

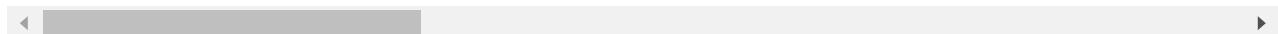
In [13]:

`df.head()`

Out[13]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_
0	M	17.99	10.38	122.80	1001.0	0.11840	0.
1	M	20.57	17.77	132.90	1326.0	0.08474	0.
2	M	19.69	21.25	130.00	1203.0	0.10960	0.
3	M	11.42	20.38	77.58	386.1	0.14250	0.
4	M	20.29	14.34	135.10	1297.0	0.10030	0.

5 rows × 31 columns



In [14]:

`backup = df.copy()`

In [15]:

`df['fractal_dimension_worst'].unique()`

Out[15]:

```
array([0.1189 , 0.08902, 0.08758, 0.173 , 0.07678, 0.1244 , 0.08368,
       0.1151 , 0.1072 , 0.2075 , 0.08452, 0.1048 , 0.1023 , 0.06287,
       0.1431 , 0.1341 , 0.08216, 0.1142 , 0.07615, 0.07259, 0.08183,
       0.07773, 0.09946, 0.07526, 0.09564, 0.1059 , 0.1275 , 0.07421,
       0.09876, 0.07919, 0.09782, 0.1402 , 0.08482, 0.1123 , 0.1233 ,
       0.08633, 0.1014 , 0.06169, 0.05504, 0.1071 , 0.07146, 0.09606,
       0.1038 , 0.1027 , 0.09618, 0.09185, 0.07409, 0.1179 , 0.08301,
       0.06917, 0.06563, 0.08025, 0.07408, 0.07987, 0.07873, 0.07036,
       0.08294, 0.1094 , 0.06289, 0.09026, 0.0802 , 0.07712, 0.1132 ,
       0.0849 , 0.1031 , 0.08911, 0.09211, 0.06641, 0.1175 , 0.0641 ,
       0.06589, 0.1084 , 0.1339 , 0.103 , 0.07609, 0.06387, 0.07191,
       0.1108 , 0.09964, 0.07918, 0.08851, 0.1016 , 0.1051 , 0.09203,
       0.07924, 0.08579, 0.06846, 0.09288, 0.09261, 0.08473, 0.07246,
       0.06828, 0.06206, 0.06603, 0.08234, 0.07376, 0.08988, 0.08756,
       0.09353, 0.07397, 0.09382, 0.06878, 0.07552, 0.1405 , 0.09097,
       0.07185, 0.09789, 0.08832, 0.08468, 0.08486, 0.1082 , 0.1017 ,
       0.08541, 0.07722, 0.1065 , 0.1252 , 0.06111, 0.08523, 0.08456,
       0.08009, 0.08006, 0.07628, 0.07182, 0.079 , 0.06541, 0.07779,
       0.08465, 0.09241, 0.08019, 0.07619, 0.07071, 0.0761 , 0.08067,
       0.07343, 0.06765, 0.07147, 0.06784, 0.08151, 0.08158, 0.08096,
       0.08118, 0.06769, 0.1036 , 0.09218, 0.07683, 0.07014, 0.06435,
       0.1486 , 0.1259 , 0.06772, 0.08132, 0.07738, 0.05972, 0.07898,
       0.07685, 0.06251, 0.09223, 0.09082, 0.09187, 0.06085, 0.07699,
       0.07228, 0.093 , 0.06428, 0.06771, 0.07371, 0.101 , 0.07313,
       0.06164, 0.07848, 0.1162 , 0.09519, 0.05843, 0.07319, 0.08082,
       0.1284 , 0.08631, 0.07427, 0.09772, 0.07697, 0.06938, 0.07097,
       0.06576, 0.06306, 0.1446 , 0.06871, 0.06559, 0.1205 , 0.08701,
       0.06949, 0.09333, 0.06558, 0.09221, 0.1013 , 0.08174, 0.07867,
       0.08762, 0.1086 , 0.0875 , 0.0974 , 0.0738 , 0.06469, 0.1076 ,
       0.07474, 0.05865, 0.07993, 0.05525, 0.06818, 0.1026 , 0.08365,
       0.07809, 0.08255, 0.07568, 0.08718, 0.08177, 0.08797, 0.1064 ,
       0.07623, 0.06072, 0.08269, 0.08362, 0.09585, 0.1243 , 0.09061,
       0.07087, 0.07307, 0.08328, 0.08178, 0.07617, 0.08677, 0.07127,
       0.07796, 0.08496, 0.0651 , 0.06783, 0.1297 , 0.06321, 0.07614,
       0.07748, 0.07198, 0.1178 , 0.08147, 0.07849, 0.06487, 0.08113,
```

```
0.0895 , 0.07957, 0.1005 , 0.1191 , 0.1019 , 0.1204 , 0.07999,
0.06515, 0.07484, 0.06829, 0.0757 , 0.08218, 0.07587, 0.07024,
0.07062, 0.0612 , 0.08022, 0.08858, 0.08175, 0.07948, 0.06033,
0.06386, 0.05737, 0.06263, 0.06912, 0.0972 , 0.06688, 0.07787,
0.1063 , 0.06431, 0.09981, 0.06915, 0.07009, 0.06994, 0.08799,
0.08472, 0.09584, 0.07007, 0.06922, 0.06794, 0.06643, 0.07676,
0.06777, 0.09929, 0.07764, 0.09469, 0.07842, 0.07638, 0.06745,
0.08385, 0.07804, 0.06192, 0.0658 , 0.06958, 0.05695, 0.08253,
0.07434, 0.08116, 0.06174, 0.06037, 0.08198, 0.1055 , 0.05932,
0.09702, 0.05933, 0.08553, 0.1024 , 0.07961, 0.06888, 0.07083,
0.07037, 0.082 , 0.07953, 0.09124, 0.09166, 0.06522, 0.07418,
0.07207, 0.07599, 0.1009 , 0.0987 , 0.07664, 0.08764, 0.09825,
0.0908 , 0.07806, 0.08488, 0.08083, 0.08187, 0.08763, 0.0759 ,
0.06825, 0.105 , 0.08815, 0.09438, 0.07018, 0.07188, 0.08317,
0.07113, 0.07431, 0.08136, 0.05521, 0.06658, 0.07238, 0.07582,
0.06735, 0.07632, 0.0747 , 0.06494, 0.08574, 0.09614, 0.06766,
0.08666, 0.07055, 0.07701, 0.0896 , 0.12 , 0.07061, 0.09638,
0.1403 , 0.09215, 0.07287, 0.09349, 0.1118 , 0.0732 , 0.06836,
0.08824, 0.06623, 0.1043 , 0.07602, 0.08865, 0.1007 , 0.07081,
0.06609, 0.07686, 0.07053, 0.09158, 0.08121, 0.1198 , 0.07262,
0.07247, 0.07834, 0.05974, 0.07732, 0.07012, 0.08503, 0.06896,
0.07745, 0.07881, 0.09206, 0.09251, 0.06165, 0.07351, 0.08304,
0.09464, 0.07123, 0.08284, 0.09208, 0.08839, 0.08061, 0.09646,
0.07662, 0.06025, 0.1155 , 0.09359, 0.08075, 0.08314, 0.06827,
0.07735, 0.07234, 0.06911, 0.0671 , 0.09532, 0.07944, 0.0681 ,
0.06736, 0.08225, 0.08251, 0.09075, 0.07285, 0.07463, 0.07425,
0.09952, 0.06091, 0.08194, 0.0781 , 0.0733 , 0.07675, 0.0722 ,
0.06788, 0.06291, 0.07211, 0.0906 , 0.06464, 0.07863, 0.06925,
0.1249 , 0.07875, 0.1224 , 0.0927 , 0.08524, 0.06639, 0.08273,
0.06743, 0.108 , 0.07802, 0.07858, 0.07698, 0.0918 , 0.09136,
0.07729, 0.07603, 0.09326, 0.0696 , 0.08181, 0.1034 , 0.06596,
0.09009, 0.08024, 0.08203, 0.05871, 0.07625, 0.07028, 0.07429,
0.06599, 0.1033 , 0.07661, 0.09445, 0.08999, 0.08549, 0.1183 ,
0.07538, 0.07277, 0.1364 , 0.1168 , 0.0723 , 0.1067 , 0.09879,
0.06142, 0.1109 , 0.07048, 0.06954, 0.08893, 0.08557, 0.08982,
0.09671, 0.07613, 0.09031, 0.09209, 0.1049 , 0.08665, 0.07592,
0.07253, 0.08052, 0.07757, 0.07782, 0.08278, 0.07569, 0.08351,
0.0997 , 0.09938, 0.1066 , 0.08134, 0.06956, 0.06443, 0.08492,
0.06953, 0.07399, 0.09479, 0.0792 , 0.07626, 0.06592, 0.08032,
0.06484, 0.07393, 0.07242, 0.08283, 0.06742, 0.06969, 0.08004,
0.08732, 0.08321, 0.05905, 0.1409 , 0.09873, 0.07115, 0.06637,
0.0782 , 0.124 , 0.07039])
```

In [17]: `df.shape`

Out[17]: (569, 31)

In [18]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   diagnosis        569 non-null    object  
 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
dtypes: float64(30), object(1)
memory usage: 137.9+ KB

```

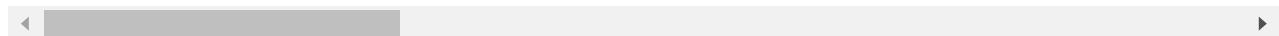
In [19]:

`df.describe()`

Out[19]:

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400

8 rows × 30 columns

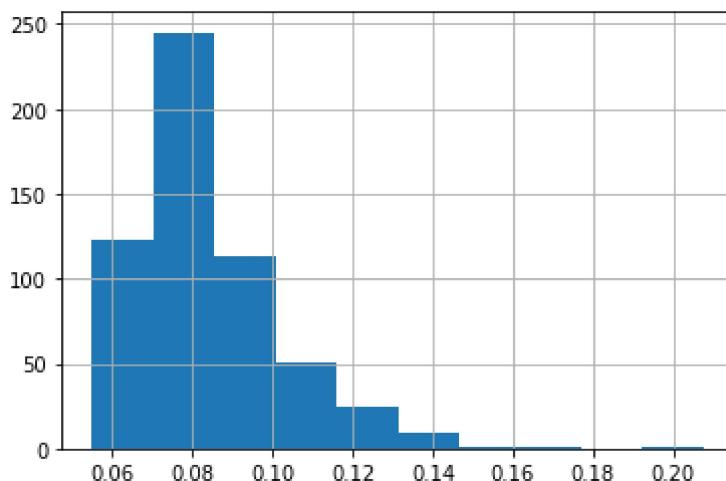


In [21]:

`df['fractal_dimension_worst'].hist()`

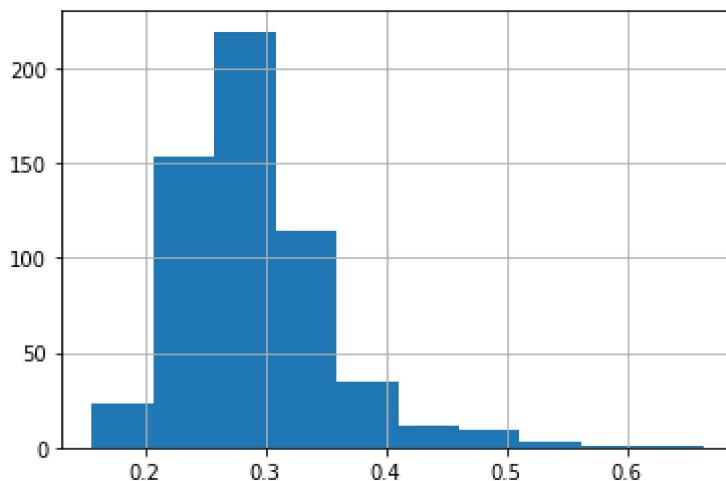
Out[21]:

<AxesSubplot:>



```
In [22]: df['symmetry_worst'].hist()
```

```
Out[22]: <AxesSubplot:>
```



```
In [23]: df['diagnosis']
```

```
Out[23]: 0      M
1      M
2      M
3      M
4      M
..
564    M
565    M
566    M
567    M
568    B
Name: diagnosis, Length: 569, dtype: object
```

```
In [24]: from sklearn.preprocessing import LabelEncoder
```

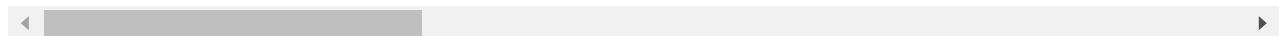
```
In [29]: le = LabelEncoder()
df['diagnosis'] = le.fit_transform(df['diagnosis'])
```

```
In [30]: df.head()
```

Out[30]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_
0	1	17.99	10.38	122.80	1001.0	0.11840	0.
1	1	20.57	17.77	132.90	1326.0	0.08474	0.
2	1	19.69	21.25	130.00	1203.0	0.10960	0.
3	1	11.42	20.38	77.58	386.1	0.14250	0.
4	1	20.29	14.34	135.10	1297.0	0.10030	0.

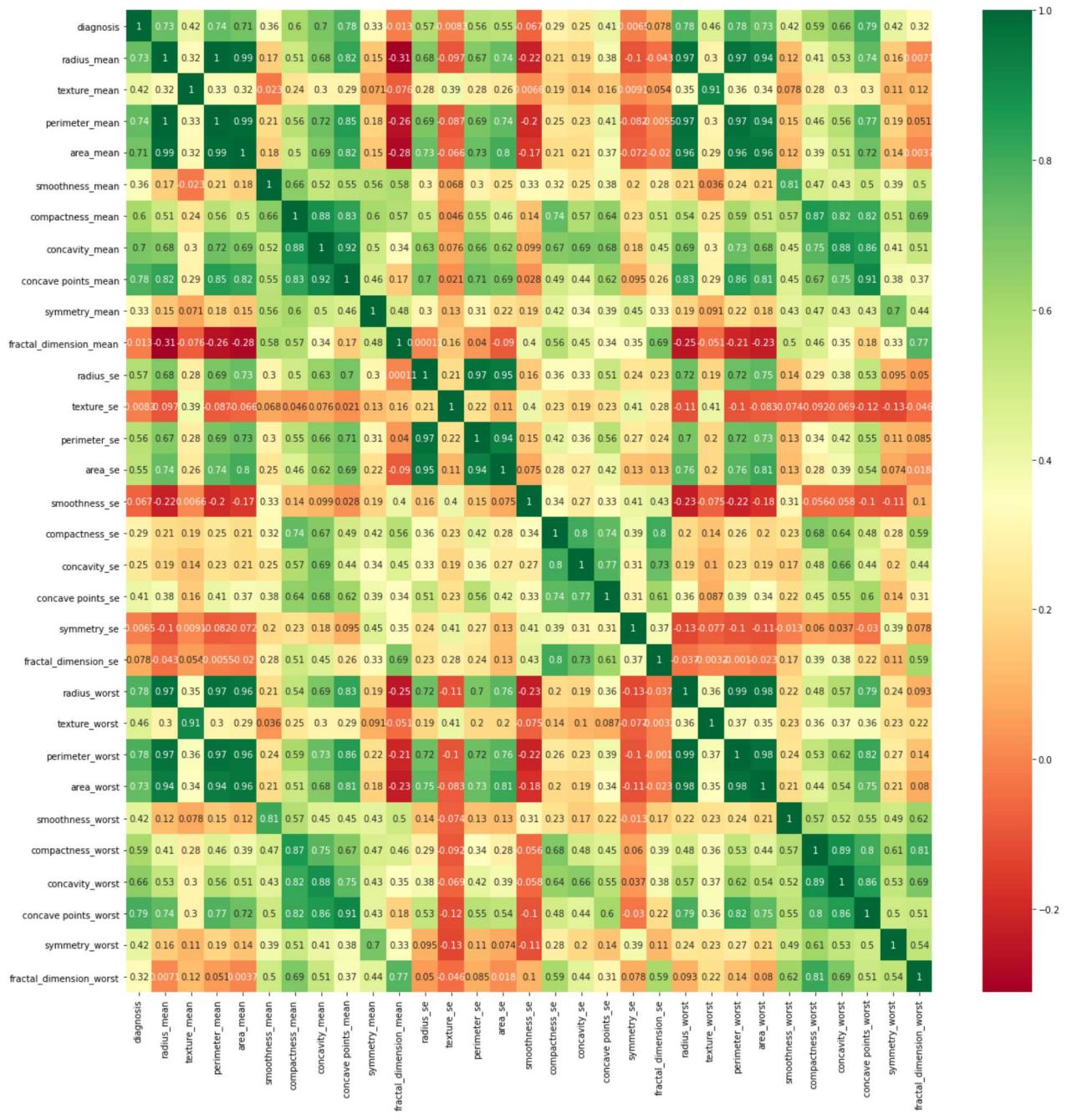
5 rows × 31 columns



In [33]:

```
plt.figure(figsize = (20,20))
sns.heatmap(df.corr(), annot = True, cmap = 'RdYlGn')
```

Out[33]: <AxesSubplot:



In [41]:

```
#Feature scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

In [42]:

```
#Implementing model
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train,y_train)
```

Out[42]: LinearRegression()

In [46]: print('Accuracy with LinearRegression : ', lr.score(x_test,y_test) * 100)

Accuracy with LinearRegression : 92.82574489099486

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 2)
x_poly = poly_reg.fit_transform(x_train)
lin_reg = LinearRegression()
lin_reg.fit(x_poly, y_train)
```

Out[51]: LinearRegression()

In [53]: print('Accuracy: Polynomial deg-2 Regression : ', lin_reg.score(x_poly, y_train) * 100)

Accuracy: Polynomial deg-2 Regression : 100.0