```
In [1]:
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
          %matplotlib inline
         import seaborn as sns
         from sklearn.model selection import train test split
         from sklearn.linear_model import LogisticRegression
          from sklearn.tree import DecisionTreeClassifier
         from sklearn.ensemble import BaggingClassifier
         from sklearn.ensemble import AdaBoostClassifier
          from sklearn.ensemble import GradientBoostingClassifier
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy score
In [2]:
         data=pd.read_csv("wisc_bc_data (1).csv")
In [3]:
         data.head()
                 id diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean compactness_mean concavity_mean poin
         0 87139402
                           В
                                    12.32
                                                 12.39
                                                               78.85
                                                                         464.1
                                                                                        0.10280
                                                                                                          0.06981
                                                                                                                         0.03987
            8910251
                           В
                                    10.60
                                                 18.95
                                                               69.28
                                                                         346.4
                                                                                        0.09688
                                                                                                          0.11470
                                                                                                                         0.06387
             905520
                           В
                                    11.04
                                                 16.83
                                                               70.92
                                                                         373.2
                                                                                        0.10770
                                                                                                          0.07804
                                                                                                                         0.03046
             868871
                           В
                                    11.28
                                                 13.39
                                                               73.00
                                                                         384.8
                                                                                        0.11640
                                                                                                          0.11360
                                                                                                                         0.04635
            9012568
                           В
                                    15.19
                                                 13.21
                                                               97.65
                                                                         711.8
                                                                                        0.07963
                                                                                                          0.06934
                                                                                                                         0.03393
        5 rows × 32 columns
In [4]:
         data.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
                                   569 non-null
              diagnosis
                                                    obiect
          1
          2
              radius mean
                                   569 non-null
                                                    float64
                                   569 non-null
                                                    float64
          3
              texture mean
          4
                                   569 non-null
                                                    float64
              perimeter mean
          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 points mean 569 non-null float64 10 symmetry mean 569 non-null float64 dimension mean 569 non-null float64 11 569 non-null 12 radius\_se float64 569 non-null float64 13 texture se 14 perimeter se 569 non-null float64 569 non-null 15 area se float64 16 smoothness\_se 569 non-null float64 569 non-null float64 17 compactness\_se 569 non-null float64 18 concavity\_se 19 points\_se 569 non-null float64 symmetry\_se 20 569 non-null float64 21 dimension se 569 non-null float64 radius\_worst 569 non-null float64 22 23 texture worst 569 non-null float64 569 non-null perimeter worst float64 float64 25 569 non-null area worst 26 smoothness worst 569 non-null float64 27 compactness worst 569 non-null float64 28 concavity worst 569 non-null float64 29 569 non-null float64 points worst 30 symmetry\_worst 569 non-null float64 569 non-null float64 31 dimension worst dtypes:  $float6\overline{4}(30)$ , int64(1), object(1)

memory usage: 142.4+ KB

```
In [5]: data.describe()
```

```
mean 3.037183e+07
                                   14.127292
                                                 19.289649
                                                                 91.969033
                                                                            654.889104
                                                                                                 0.096360
                                                                                                                     0.104341
                                                                                                                                     0.088799
                                                                                                                                                   0.0
             std
                 1.250206e+08
                                   3.524049
                                                 4.301036
                                                                 24.298981
                                                                            351.914129
                                                                                                 0.014064
                                                                                                                     0.052813
                                                                                                                                     0.079720
                                                                                                                                                   0.0
             min 8.670000e+03
                                    6.981000
                                                  9.710000
                                                                 43.790000
                                                                             143.500000
                                                                                                 0.052630
                                                                                                                     0.019380
                                                                                                                                     0.000000
                                                                                                                                                   0.0
            25% 8.692180e+05
                                   11.700000
                                                 16.170000
                                                                            420.300000
                                                                                                 0.086370
                                                                                                                     0.064920
                                                                                                                                     0.029560
                                                                 75.170000
                                                                                                                                                   0.0
            50%
                 9.060240e+05
                                   13.370000
                                                 18.840000
                                                                 86.240000
                                                                            551.100000
                                                                                                 0.095870
                                                                                                                     0.092630
                                                                                                                                     0.061540
                                                                                                                                                   0.0
                 8.813129e+06
                                                                104.100000
                                   15.780000
                                                 21.800000
                                                                            782.700000
                                                                                                 0.105300
                                                                                                                     0.130400
                                                                                                                                     0.130700
                                                                                                                                                   0.0
                                                                                                                                     0.426800
            max 9.113205e+08
                                   28.110000
                                                 39.280000
                                                                188.500000 2501.000000
                                                                                                 0.163400
                                                                                                                     0.345400
                                                                                                                                                   0.2
          8 rows × 31 columns
 In [6]:
            data.columns
           Out[6]:
                   'compactness_se', 'concavity_se', 'points_se', 'symmetry_se',
'dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst',
'area_worst', 'smoothness_worst', 'compactness_worst',
'concavity_worst', 'points_worst', 'symmetry_worst', 'dimension_worst'],
                  dtype='object')
 In [7]:
            data.shape
           (569, 32)
 Out[7]:
 In [8]:
            data.isnull().sum()
                                    0
           id
 Out[8]:
           diagnosis
                                    0
           radius mean
                                    0
                                    0
           texture mean
           perimeter_mean
                                    0
           area mean
                                    0
           smoothness mean
           compactness_mean
                                    0
           concavity_mean
                                    0
                                    0
           points mean
           symmetry_mean
                                    0
           dimension_mean
                                    0
           radius_se
                                    0
                                    0
           texture se
           perimeter_se
                                    0
                                    0
           area se
           smoothness se
                                    0
                                    0
           compactness se
           concavity_se
                                    0
           points_se
                                    0
           symmetry se
           dimension_se
                                    0
           radius worst
                                    0
           texture worst
                                    0
           perimeter worst
                                    0
           area worst
                                    0
                                    0
           smoothness_worst
           compactness worst
                                    0
           concavity_worst
                                    0
           points worst
                                    0
           symmetry worst
                                    0
           dimension_worst
                                    0
           dtype: int64
 In [9]:
            df=pd.get dummies(data,columns=['diagnosis'])
In [10]:
            df.sample()
                       id radius_mean texture_mean perimeter_mean area_mean smoothness_mean compactness_mean concavity_mean points_mean
Out[10]:
```

count 5.690000e+02

569.000000

569.000000

569.000000

569.000000

569.000000

569.000000

569.000000

569.0

**344** 86730502 16.16 21.54 106.2 809.8 0.1008 0.1284 0.1043 0.05613

1 rows × 33 columns

## In [11]: df.median()

Out[11]: id 906024.000000 radius mean 13.370000 18.840000 texture mean 86.240000 perimeter\_mean area mean 551.100000 0.095870 smoothness mean 0.092630 compactness\_mean concavity\_mean 0.061540 0.033500 points mean 0.179200 symmetry\_mean dimension\_mean 0.061540 radius se 0.324200 texture se 1.108000 2.287000 perimeter\_se area\_se 24.530000 smoothness se 0.006380 compactness se 0.020450 0.025890 concavity\_se points se 0.010930 symmetry\_se 0.018730  ${\tt dimension\_se}$ 0.003187 radius\_worst 14.970000 texture worst 25.410000 97.660000 perimeter worst 686.500000 area\_worst smoothness worst 0.131300 compactness worst 0.211900 0.226700 concavity\_worst points\_worst 0.099930 symmetry worst 0.282200 0.080040 dimension\_worst

1.000000

0.000000

In [12]: corr=data.corr()

diagnosis\_B

diagnosis M

dtype: float64

In [13]: co

CO

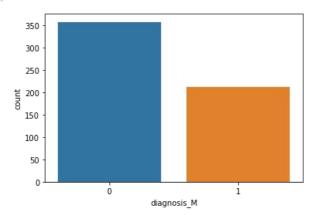
radius_m	id 1.000000		texture_mean 0.099770	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mear
_		0.074626	0 099770					
_	ean 0.074626		0.000110	0.073159	0.096893	-0.012968	0.000096	0.050080
texture m		1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764
_	ean 0.099770	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418
perimeter_m	ean 0.073159	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136
area_m	ean 0.096893	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983
smoothness_m	ean -0.012968	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984
compactness_m	ean 0.000096	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.88312
concavity_m	ean 0.050080	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000
points_m	ean 0.044158	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.92139 <sup>-</sup>
symmetry_m	ean -0.022114	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667
dimension_m	ean -0.052511	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783
radius	_ <b>se</b> 0.143048	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631928
texture	_ <b>se</b> -0.007526	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218
perimeter	_ <b>se</b> 0.137331	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.66039
area	_ <b>se</b> 0.177742	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427
smoothness	_ <b>se</b> 0.096781	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564
compactness	_ <b>se</b> 0.033961	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	0.670279
concavity	_ <b>se</b> 0.055239	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	0.691270
points	_ <b>se</b> 0.078768	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	0.683260
symmetry	_ <b>se</b> -0.017306	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	0.178009
dimension	_ <b>se</b> 0.025725	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	0.449301

radius_worst	0.082405	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	0.688236
texture_worst	0.064720	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	0.299879
perimeter_worst	0.079986	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	0.729565
area_worst	0.107187	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	0.675987
smoothness_worst	0.010338	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	0.448822
compactness_worst	-0.002968	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	0.754968
concavity_worst	0.023203	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	0.884103
points_worst	0.035174	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	0.861323
symmetry_worst	-0.044224	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	0.409464
dimension_worst	-0.029866	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	0.514930

31 rows × 31 columns

In [15]: sns.countplot(x='diagnosis\_M',data=df)

Out[15]: <AxesSubplot:xlabel='diagnosis\_M', ylabel='count'>

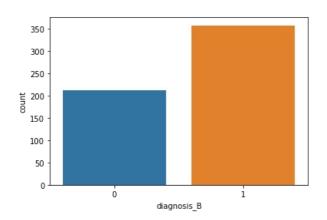


## In [18]: sns.countplot(df['diagnosis\_B'])

C:\Users\Sathvika Reddy.M\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the followin g variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

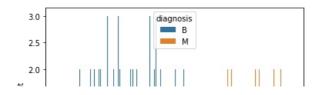
warnings.warn(

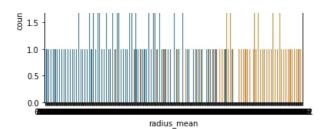
Out[18]: <AxesSubplot:xlabel='diagnosis\_B', ylabel='count'>



```
In [19]:
    sns.countplot(x='radius_mean', hue='diagnosis', data=data)
```

Out[19]: <AxesSubplot:xlabel='radius\_mean', ylabel='count'>

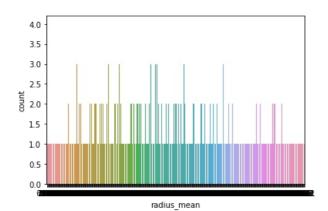




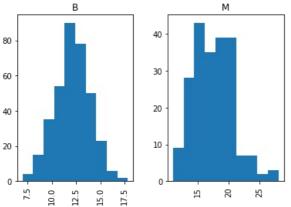
In [20]: sns.countplot(data['radius\_mean'])

C:\Users\Sathvika Reddy.M\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the followin
g variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

<AxesSubplot:xlabel='radius mean', ylabel='count'>

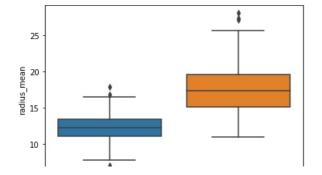


```
In [21]: data.hist(by='diagnosis',column='radius_mean')
```



```
In [22]:
    sns.boxplot(x='diagnosis',y='radius_mean',data=data)
```

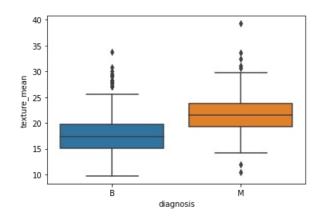
Out[22]: <AxesSubplot:xlabel='diagnosis', ylabel='radius\_mean'>



```
B diagnosis
```

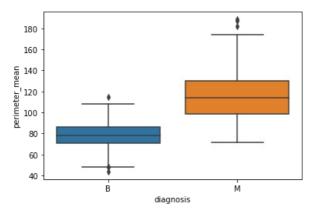
```
In [23]: sns.boxplot(x='diagnosis',y='texture_mean',data=data)
```

Out[23]: <AxesSubplot:xlabel='diagnosis', ylabel='texture\_mean'>



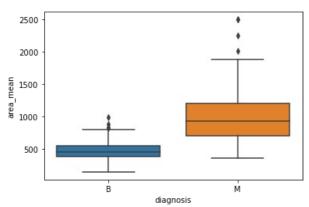
```
In [24]: sns.boxplot(x='diagnosis',y='perimeter_mean',data=data)
```

Out[24]: <AxesSubplot:xlabel='diagnosis', ylabel='perimeter\_mean'>



```
In [25]: sns.boxplot(x='diagnosis',y='area_mean',data=data)
```

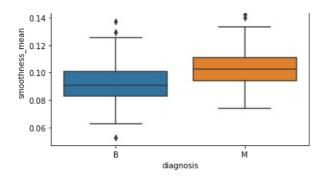
Out[25]: <AxesSubplot:xlabel='diagnosis', ylabel='area\_mean'>



```
In [26]: sns.boxplot(x='diagnosis',y='smoothness_mean',data=data)
```

Out[26]: <AxesSubplot:xlabel='diagnosis', ylabel='smoothness\_mean'>

0.16



In [27]:

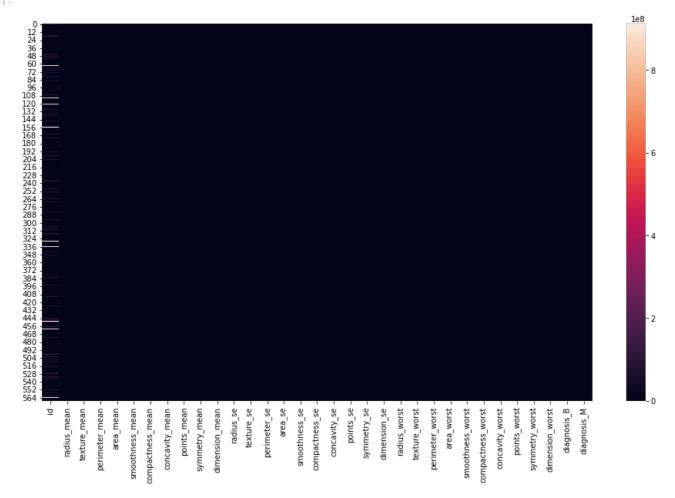
plt.figure(figsize=(30, 30))
sns.heatmap(df.corr(), annot=True)

<AxesSubplot:> Out[27]:

id	- 1	0.075	0.1	0.073	0.097	-0.013	9.6e-05	0.05	0.044	-0.022	-0.053	0.14	-0.0075	0.14	0.18	0.097	0.034	0.055	0.079	-0.017	0.026	0.082	0.065	0.08	0.11	0.01	-0.003	0.023	0.035	-0.044	-0.03	-0.04	0.04
radius_mean	0.075	1	0.32	1	0.99	0.17	0.51	0.68	0.82	0.15		0.68		0.67	0.74							0.97		0.97	0.94		0.41	0.53	0.74			-0.73	0.73
texture_mean	0.1	0.32	1	0.33	0.32	-0.023			0.29													0.35	0.91	0.36	0.34				0.3			-0.42	0.42
perimeter_mean	0.073	1	0.33	1	0.99	0.21	0.56	0.72	0.85	0.18		0.69		0.69	0.74							0.97		0.97	0.94			0.56	0.77			-0.74	0.74
area_mean	0.097	0.99		0.99	1	0.18		0.69	0.82	0.15		0.73		0.73	0.8							0.96		0.96	0.96				0.72			-0.71	0.71
smoothness_mean	-0.013					1	0.66		0.55	0.56	0.58															0.81						-0.36	0.36
compactness_mean	9.6e-05	0.51		0.56		0.66	1	0.88	0.83	0.6	0.57			0.55		0.14	0.74	0.57	0.64					0.59		0.57	0.87	0.82	0.82		0.69	-0.6	0.6
concavity_mean	0.05	0.68		0.72	0.69		0.88	1	0.92			0.63		0.66	0.62	0.099	0.67	0.69	0.68			0.69		0.73	0.68		0.75	0.88	0.86		0.51	-0.7	0.7
points_mean	0.044	0.82		0.85	0.82	0.55	0.83	0.92	1	0.46	0.17	0.7		0.71	0.69	0.028			0.62			0.83		0.86	0.81		0.67	0.75	0.91		0.37	-0.78	0.78
symmetry_mean	-0.022					0.56	0.6		0.46	1	0.48	0.3				0.19						0.19							0.43	0.7	0.44	-0.33	0.33
dimension_mean	0.053	-0.31	-0.076	-0.26	-0.28	0.58	0.57	0.34	0.17		1	0.00011		0.04	-0.09	0.4	0.56				0.69	-0.25		-0.21	-0.23	0.5			0.18		0.77	0.013	-0.013
radius_se	0.14	0.68		0.69	0.73	0.3	0.5	0.63	0.7			1	0.21	0.97	0.95							0.72		0.72	0.75		0.29	0.38	0.53			-0.57	0.57
texture_se	-0.0075	-0.097		-0.087	-0.066	0.068	0.046	0.076	0.021	0.13		0.21	1	0.22	0.11				0.23			-0.11		-0.1	-0.083		-0.092	-0.069	-0.12			0.0083	0.0083
perimeter_se	0.14	0.67		0.69	0.73	0.3	0.55	0.66	0.71			0.97		1	0.94				0.56			0.7		0.72	0.73				0.55			-0.56	0.56
area_se	0.18	0.74	0.26	0.74	0.8	0.25	0.46	0.62	0.69		-0.09	0.95	0.11	0.94	1	0.075						0.76		0.76	0.81			0.39	0.54			-0.55	0.55
smoothness_se	0.097					0.33	0.14	0.099	0.028	0.19						1	0.34	0.27	0.33		0.43	-0.23				0.31	-0.056	-0.058	-0.1		0.1	0.067	-0.067
compactness_se	0.034						0.74	0.67			0.56					0.34	1	0.8	0.74		0.8					0.23	0.68	0.64			0.59	-0.29	0.29
concavity_se	0.055						0.57	0.69								0.27	0.8	1	0.77		0.73							0.66	0.44			-0.25	0.25
points_se	0.079	0.38		0.41	0.37	0.38	0.64	0.68	0.62					0.56			0.74	0.77	1	0.31	0.61	0.36		0.39	0.34		0.45	0.55	0.6		0.31	-0.41	0.41
symmetry_se	-0.017										0.35						0.39	0.31		1	0.37						0.06	0.037	-0.03		0.078		0.0065
dimension_se	0.026	-0.043	0.054	-0.0055	-0.02	0.28			0.26		0.69	0.23	0.28	0.24	0.13	0.43	0.8	0.73	0.61	0.37	1	-0.037	-0.0032	-0.001	-0.023				0.22		0.59	-0.078	0.078
radius_worst	0.082	0.97	0.35	0.97	0.96	0.21	0.54	0.69	0.83			0.72	-0.11	0.7	0.76							1	0.36	0.99	0.98			0.57	0.79			-0.78	0.78
texture_worst	0.065	0.3	0.91	0.3	0.29	0.036	0.25	0.3	0.29			0.19	0.41	0.2	0.2							0.36	1	0.37	0.35				0.36			-0.46	0.46
perimeter_worst	0.08	0.97		0.97	0.96	0.24	0.59	0.73	0.86			0.72		0.72	0.76	-0.22						0.99		1	0.98			0.62	0.82			-0.78	0.78
area_worst	0.11	0.94	0.34	0.94	0.96	0.21	0.51	0.68	0.81	0.18	-0.23	0.75		0.73	0.81	-0.18						0.98		0.98	1	0.21			0.75		0.08	-0.73	0.73
smoothness_worst	0.01	0.12		0.15		0.81	0.57	0.45								0.31	0.23	0.17							0.21	1	0.57	0.52	0.55		0.62	-0.42	0.42
compactness_worst	-0.003	0.41					0.87	0.75	0.67							-0.056	0.68									0.57	1	0.89	0.8	0.61	0.81	-0.59	0.59
concavity_worst	0.023	0.53		0.56					0.75							-0.058	0.64	0.66	0.55					0.62			0.89	1	0.86		0.69	-0.66	0.66
					0.72			0.86				0.53		0.55	0.54	-0.1									0.75			0.86		0.5	0.51	-0.79	0.79
symmetry_worst																																-0.42	
dimension_worst																																	0.32
diagnosis_B				_																													-1
diagnosis_M	0.04	ė	,	0.74	0.71	0.36	0.6	0.7	ė	0.33	-0.013	0.57	-0.0083	0.56	-	-0.067	0.29		0.41	-0.0065	0.078	1	0.46	0.78	Į.	0.42	0.59	0.66	į.	0.42	ļ.	-1 - B	1
	.9	radius_mean	texture_mean	perimeter_mean	area_mear	noothness_mear	npactness_mear	concavity_mear	points_mean	symmetry_mear	dimension_mear	radius_se	texture_s.	perimeter_st	area_se	smoothness so	compactness_st	concavity se	points_s6	symmetry se	dimension_s&	radius_worst	texture_wors	perimeter_worst	area_worst	noothness wors	npactness_wors	concavity_wors	points_worst	symmetry_worst	ilmension_worst	diagnosis	diagnosis_M

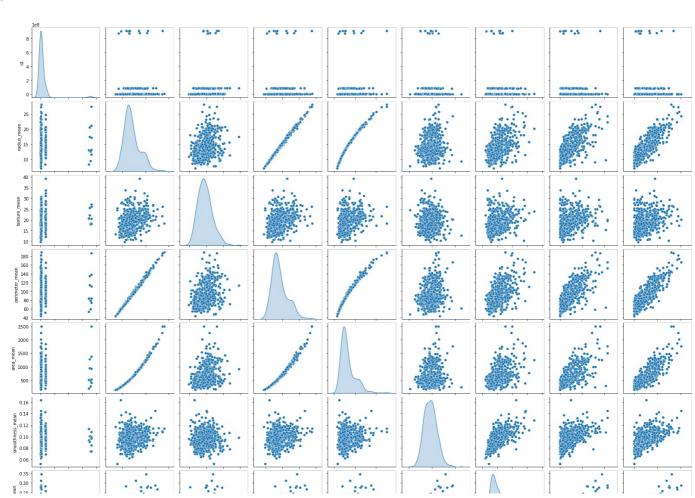
plt.figure(figsize=(16,9))
sns.heatmap(df)

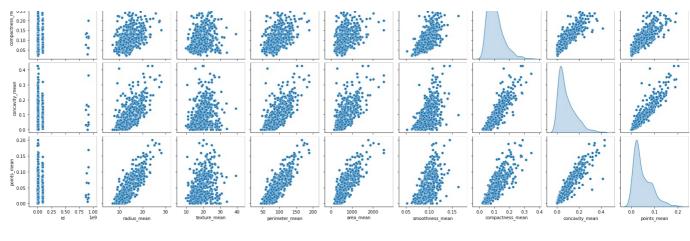
In [28]:



```
In [29]:
    df_attr = df.iloc[:, 0:9]
    sns.pairplot(df_attr, diag_kind='kde')
```

Out[29]: <seaborn.axisgrid.PairGrid at 0x1dce5d98a90>





```
In [30]:
          x=df.drop(['diagnosis_B','diagnosis_M'],axis=1)
          y=df['diagnosis_M']
          x\_train, x\_test, y\_train, y\_test=train\_test\_split(x, y, test\_size=0.35, random\_state=5)
In [31]:
          model=LogisticRegression()
          model.fit(x_train,y_train)
         LogisticRegression()
Out[31]:
In [32]:
          dF=model.predict(x_test)
In [33]:
          accuracy_score(y_test,dF)
         0.645
Out[33]:
In [34]:
          accuracy_score(y_test,dF)
         0.645
Out[34]:
In [35]:
          dtree=DecisionTreeClassifier(criterion='gini', random_state=5)
In [36]:
          dtree.fit(x_train,y_train)
         DecisionTreeClassifier(random_state=5)
Out[36]:
In [37]:
          DecisionTreeClassifier(random_state=5)
         DecisionTreeClassifier(random_state=5)
In [38]:
          print(dtree.score(x_train,y_train))
          print(dtree.score(x_test,y_test))
         1.0
         0.945
In [39]:
          dtree=DecisionTreeClassifier(criterion='entropy', random_state=5)
          dtree.fit(x_train,y_train)
         DecisionTreeClassifier(criterion='entropy', random_state=5)
Out[39]:
```

In [40]:

print(dtree.score(x\_train,y\_train))

```
print(dtree.score(x_test,y_test))
         1.0
         0.95
In [41]:
          dtree=DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=5)
          dtree.fit(x_train,y_train)
          print(dtree.score(x_train,y_train))
          print(dtree.score(x_test,y_test))
         0.975609756097561
         0.945
In [42]:
          bg=BaggingClassifier(n_estimators=50,base_estimator=dtree,random_state=5)
          bg=bg.fit(x train,y train)
          y_predict=bg.predict(x_test)
          print(bg.score(x_test,y_test))
         0.96
In [43]:
          ad=AdaBoostClassifier(n_estimators=50,random_state=5)
          ad=ad.fit(x_train,y_train)
          y_predict=ad.predict(x_test)
          print(ad.score(x_test,y_test))
         0.975
In [44]:
          gd=GradientBoostingClassifier(n_estimators=50, random_state=5)
          gd=gd.fit(x_train,y_train)
          y_predict=gd.predict(x_test)
          print(gd.score(x_test,y_test))
         0.965
In [45]:
          rf=RandomForestClassifier(n_estimators=75, random_state=5, max_features=3)
          rf=rf.fit(x_train,y_train)
          y predict=rf.predict(x test)
          print(rf.score(x_test,y_test))
         0.97
 In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js