Importing the Libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
from sklearn.datasets import load_breast_cancer
```

In [7]:
```
cancer = load_breast_cancer()
```

In [57]:
```
cancer.keys()
```

Out[57]:
```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'file
name'])
```

In [58]:
```
cancer['feature_names']
```

Out[58]:
```
array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal dimension',
       'radius error', 'texture error', 'perimeter error', 'area error',
       'smoothness error', 'compactness error', 'concavity error',
       'concave points error', 'symmetry error',
       'fractal dimension error', 'worst radius', 'worst texture',
       'worst perimeter', 'worst area', 'worst smoothness',
       'worst compactness', 'worst concavity', 'worst concave points',
       'worst symmetry', 'worst fractal dimension'], dtype='<U23')
```

In [40]:
```
print(cancer['data'])
```

```
[[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 [41]:
```
cancer['data'].shape
```

Out[41]:
```
(569, 30)
```

In [42]:
```
df_cancer = pd.DataFrame(np.c_[cancer['data'],cancer['target']], columns =np.
```

In [43]:
```
df_cancer.head()
```

Out[43]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	d
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	

5 rows × 31 columns

In [44]:
```
df_cancer.tail()
```

Out[44]:

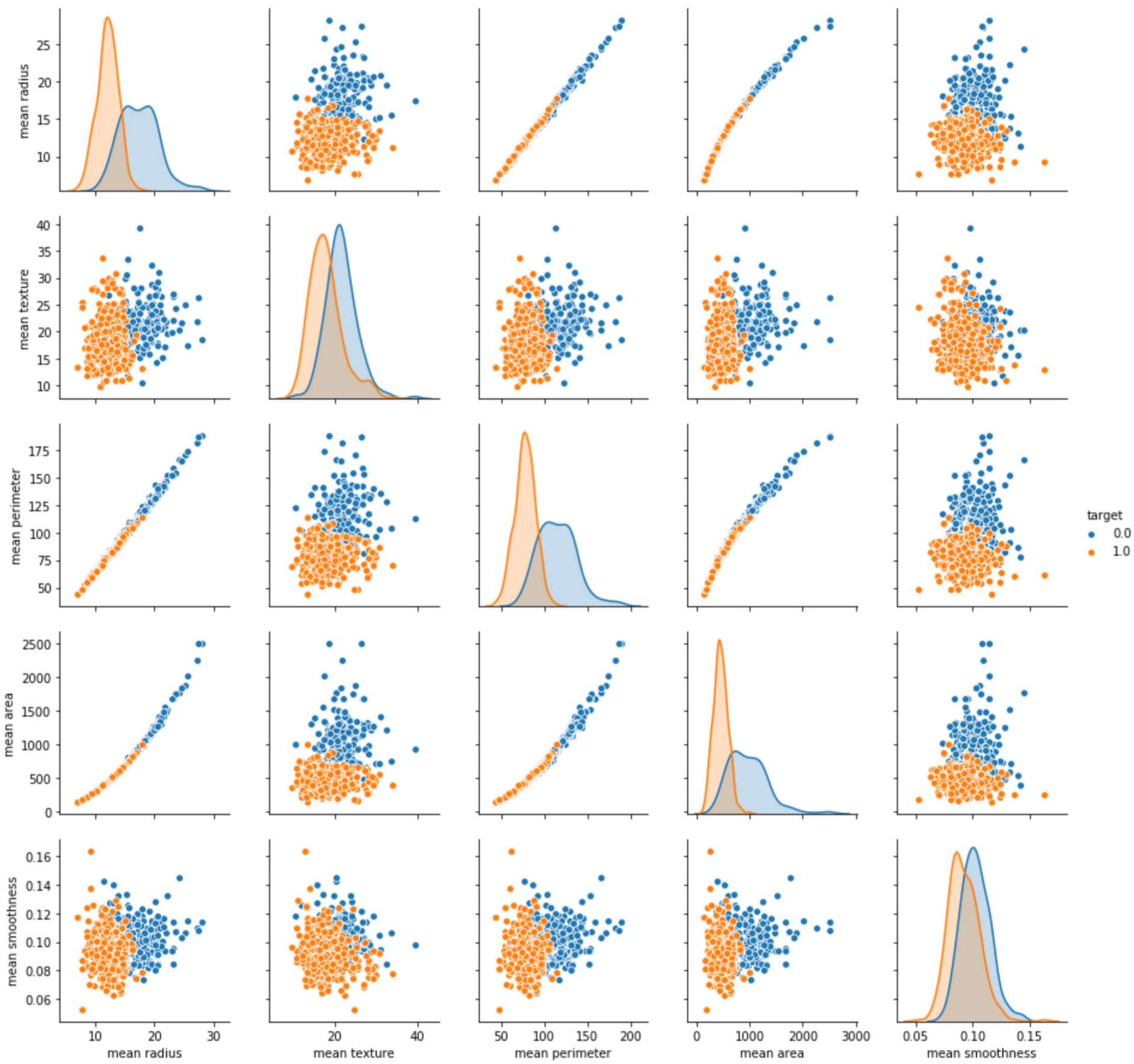
	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587

5 rows × 31 columns

Visualizing The DataSet

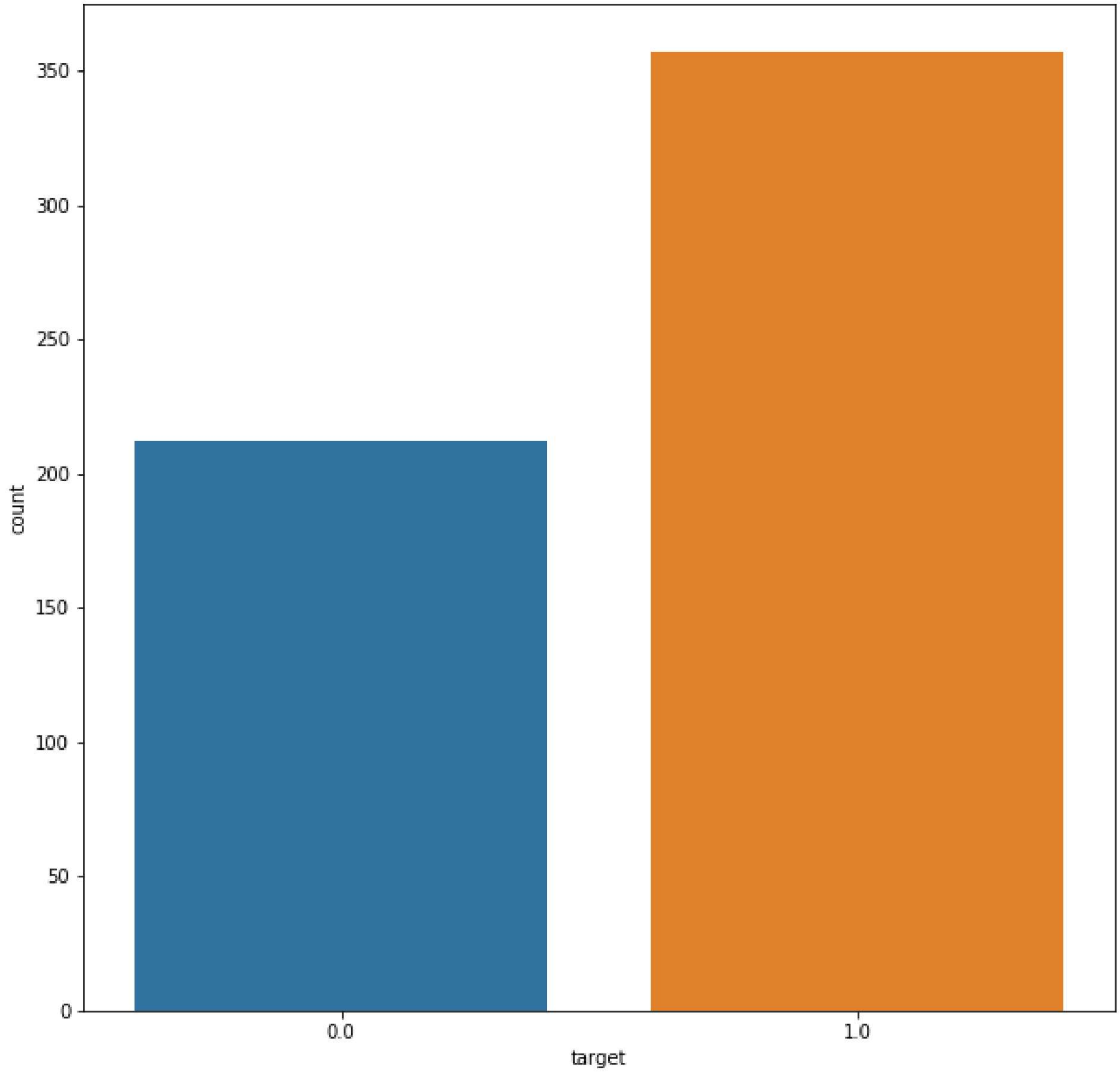
In [64]:
```
sns.pairplot(df_cancer, hue = 'target', vars = ['mean radius', 'mean texture'
```

Out[64]: <seaborn.axisgrid.PairGrid at 0x236a89d5f28>



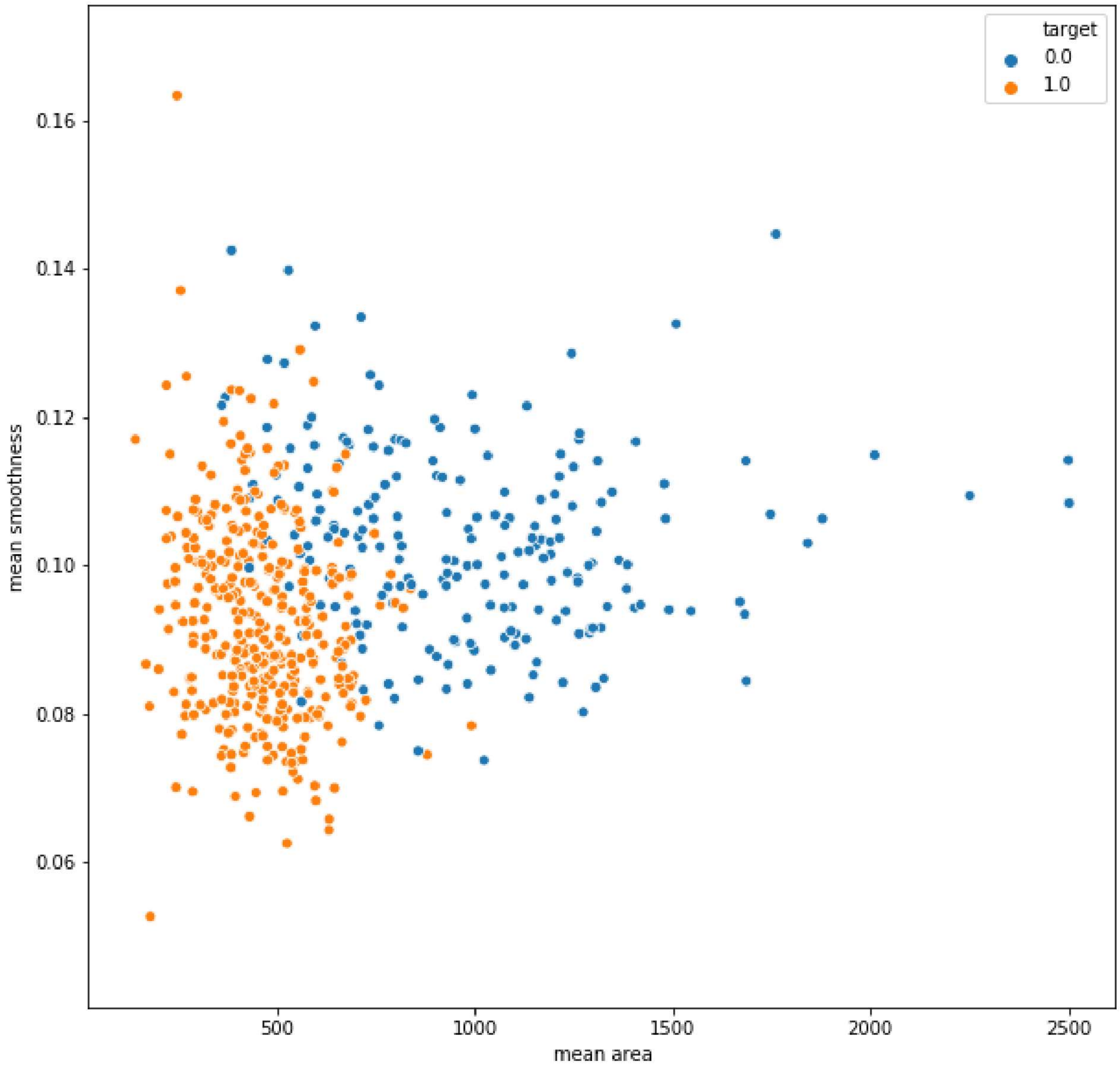
In [77]:
```
plt.figure(figsize = (10,10))

sns.countplot(df_cancer['target'])
```

Out[77]: <matplotlib.axes._subplots.AxesSubplot at 0x236ac089f60>



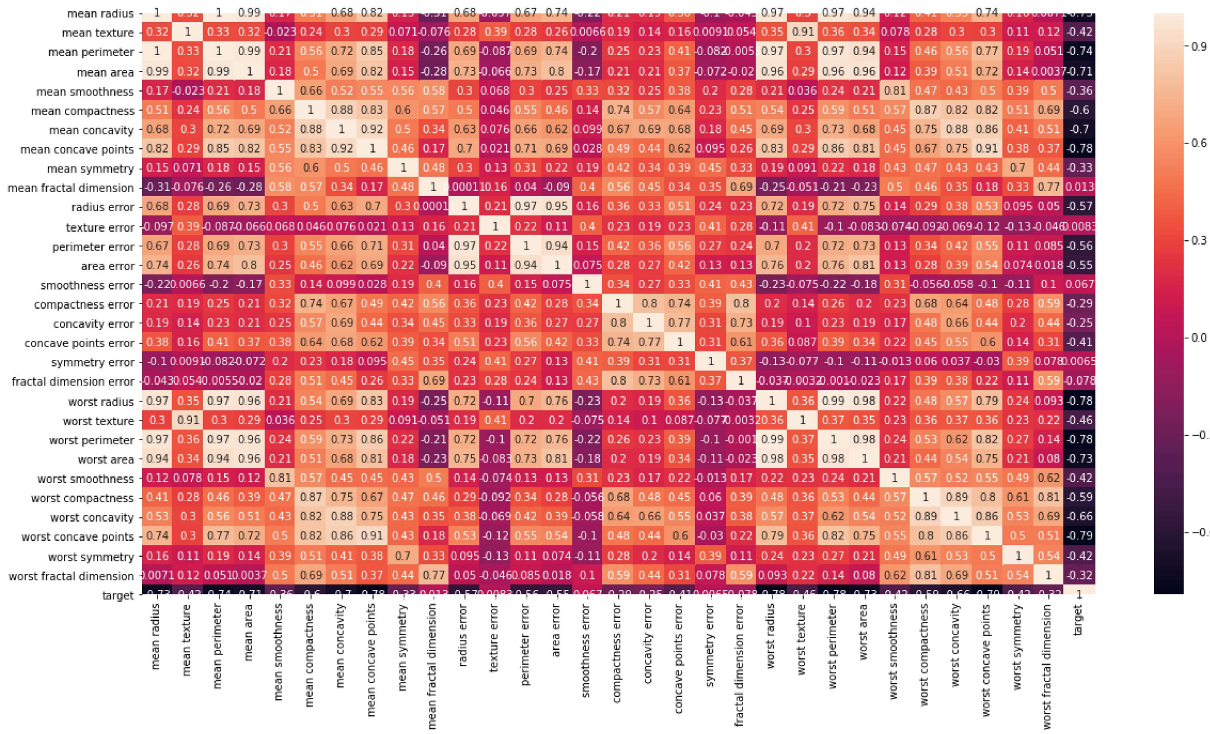
In [78]:
```
plt.figure(figsize = (10,10))
sns.scatterplot(x = 'mean area', y='mean smoothness', hue = 'target', data =
```

Out[78]: <matplotlib.axes._subplots.AxesSubplot at 0x236ac6ec828>



In [73]:
```
plt.figure(figsize = (20,10))
sns.heatmap(df_cancer.corr(), annot = True)
```

Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0x236abb86160>



Model Training

In [103]:
```
X = df_cancer.drop(['target'], axis = 1)
```

In [104]:
```
X
```

Out[104]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	me symme
0	17.990	10.38	122.80	1001.0	0.11840	0.27760	0.300100	0.147100	0.2
1	20.570	17.77	132.90	1326.0	0.08474	0.07864	0.086900	0.070170	0.1
2	19.690	21.25	130.00	1203.0	0.10960	0.15990	0.197400	0.127900	0.2
3	11.420	20.38	77.58	386.1	0.14250	0.28390	0.241400	0.105200	0.2
4	20.290	14.34	135.10	1297.0	0.10030	0.13280	0.198000	0.104300	0.1
5	12.450	15.70	82.57	477.1	0.12780	0.17000	0.157800	0.080890	0.2
6	18.250	19.98	119.60	1040.0	0.09463	0.10900	0.112700	0.074000	0.1
7	13.710	20.83	90.20	577.9	0.11890	0.16450	0.093660	0.059850	0.2
8	13.000	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2

In [105]:
```
y = df_cancer['target']
```

In [106]:
```
y
```

Out[106]:
```
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
      ...
564    0.0
565    0.0
566    0.0
567    0.0
568    1.0
Name: target, Length: 569, dtype: float64
```

In [107]:
```
from sklearn.model_selection import train_test_split
```

In [112]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

In [113]:
```
X_train
```

Out[113]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
306	13.200	15.82	84.07	537.3	0.08511	0.05251	0.001461	0.003261	0.1632
410	11.360	17.57	72.49	399.8	0.08858	0.05313	0.027830	0.021000	0.1601
197	18.080	21.84	117.40	1024.0	0.07371	0.08642	0.110300	0.057780	0.1770
376	10.570	20.22	70.15	338.3	0.09073	0.16600	0.228000	0.059410	0.2188
244	19.400	23.50	129.10	1155.0	0.10270	0.15580	0.204900	0.088860	0.1978
299	10.510	23.09	66.85	334.2	0.10150	0.06797	0.024950	0.018750	0.1695
312	12.760	13.37	82.29	504.1	0.08794	0.07948	0.040520	0.025480	0.1601
331	12.980	19.35	84.52	514.0	0.09579	0.11250	0.071070	0.029500	0.1761
317	18.220	18.87	118.70	1027.0	0.09746	0.11170	0.113000	0.079500	0.1807
341	9.606	16.84	61.64	280.5	0.08481	0.09228	0.084220	0.022920	0.2036
156	17.680	20.74	117.40	963.7	0.11150	0.16650	0.185500	0.105400	0.1971
71	8.888	14.64	58.79	244.0	0.09783	0.15310	0.086060	0.028720	0.1902
218	19.800	21.56	129.70	1230.0	0.09383	0.13060	0.127200	0.086910	0.2094
344	11.710	15.45	75.03	420.3	0.11500	0.07281	0.040060	0.032500	0.2009
247	12.890	14.11	84.95	512.2	0.08760	0.13460	0.137400	0.039800	0.1596
212	28.110	18.47	188.50	2499.0	0.11420	0.15160	0.320100	0.159500	0.1648
559	11.510	23.93	74.52	403.5	0.09261	0.10210	0.111200	0.041050	0.1388
176	9.904	18.06	64.60	302.4	0.09699	0.12940	0.130700	0.037160	0.1669
422	11.610	16.02	75.46	408.2	0.10880	0.11680	0.070970	0.044970	0.1886
248	10.650	25.22	68.01	347.0	0.09657	0.07234	0.023790	0.016150	0.1897
232	11.220	33.81	70.79	386.8	0.07780	0.03574	0.004967	0.006434	0.1845
444	18.030	16.85	117.50	990.0	0.08947	0.12320	0.109000	0.062540	0.1720
383	12.390	17.48	80.64	462.9	0.10420	0.12970	0.058920	0.028800	0.1779
279	13.850	15.18	88.99	587.4	0.09516	0.07688	0.044790	0.037110	0.2110
494	13.160	20.54	84.06	538.7	0.07335	0.05275	0.018000	0.012560	0.1713
316	12.180	14.08	77.25	461.4	0.07734	0.03212	0.011230	0.005051	0.1673
523	13.710	18.68	88.73	571.0	0.09916	0.10700	0.053850	0.037830	0.1714
90	14.620	24.02	94.57	662.7	0.08974	0.08606	0.031020	0.029570	0.1685
469	11.620	18.18	76.38	408.8	0.11750	0.14830	0.102000	0.055640	0.1957
373	20.640	17.35	134.80	1335.0	0.09446	0.10760	0.152700	0.089410	0.1571
...	...	...	...	...	...	...	...	...	...
539	7.691	25.44	48.34	170.4	0.08668	0.11990	0.092520	0.013640	0.2037

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
110	9.777	16.99	62.50	290.2	0.10370	0.08404	0.043340	0.017780	0.1584
5	12.450	15.70	82.57	477.1	0.12780	0.17000	0.157800	0.080890	0.2087
144	10.750	14.97	68.26	355.3	0.07793	0.05139	0.022510	0.007875	0.1399
103	9.876	19.40	63.95	298.3	0.10050	0.09697	0.061540	0.030290	0.1945
210	20.580	22.14	134.70	1290.0	0.09090	0.13480	0.164000	0.095610	0.1765
446	17.750	28.03	117.30	981.6	0.09997	0.13140	0.169800	0.082930	0.1713
41	10.950	21.35	71.90	371.1	0.12270	0.12180	0.104400	0.056690	0.1895
362	12.760	18.84	81.87	496.6	0.09676	0.07952	0.026880	0.017810	0.1759
377	13.460	28.21	85.89	562.1	0.07517	0.04726	0.012710	0.011170	0.1421
254	19.450	19.33	126.50	1169.0	0.10350	0.11880	0.137900	0.085910	0.1776
146	11.800	16.58	78.99	432.0	0.10910	0.17000	0.165900	0.074150	0.2678
86	14.480	21.46	94.25	648.2	0.09444	0.09947	0.120400	0.049380	0.2075
542	14.740	25.42	94.70	668.6	0.08275	0.07214	0.041050	0.030270	0.1840
431	12.400	17.68	81.47	467.8	0.10540	0.13160	0.077410	0.027990	0.1811
65	14.780	23.94	97.40	668.3	0.11720	0.14790	0.126700	0.090290	0.1953
205	15.120	16.68	98.78	716.6	0.08876	0.09588	0.075500	0.040790	0.1594
44	13.170	21.81	85.42	531.5	0.09714	0.10470	0.082590	0.052520	0.1746
27	18.610	20.25	122.10	1094.0	0.09440	0.10660	0.149000	0.077310	0.1697
80	11.450	20.97	73.81	401.5	0.11020	0.09362	0.045910	0.022330	0.1842
437	14.040	15.98	89.78	611.2	0.08458	0.05895	0.035340	0.029440	0.1714
113	10.510	20.19	68.64	334.2	0.11220	0.13030	0.064760	0.030680	0.1922
204	12.470	18.60	81.09	481.9	0.09965	0.10580	0.080050	0.038210	0.1925
519	12.750	16.70	82.51	493.8	0.11250	0.11170	0.038800	0.029950	0.2120
411	11.040	16.83	70.92	373.2	0.10770	0.07804	0.030460	0.024800	0.1714
8	13.000	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2350
73	13.800	15.79	90.43	584.1	0.10070	0.12800	0.077890	0.050690	0.1662
400	17.910	21.02	124.40	994.0	0.12300	0.25760	0.318900	0.119800	0.2113
118	15.780	22.91	105.70	782.6	0.11550	0.17520	0.213300	0.094790	0.2096
206	9.876	17.27	62.92	295.4	0.10890	0.07232	0.017560	0.019520	0.1934

455 rows × 30 columns

In [114]:
```
y_train
```

Out[114]:
```
306    1.0
410    1.0
197    0.0
376    1.0
244    0.0
      ...
8      0.0
73     0.0
400    0.0
118    0.0
206    1.0
Name: target, Length: 455, dtype: float64
```

In [115]:
```
X_test
```

Out[115]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mear symmetry
28	15.300	25.27	102.40	732.4	0.10820	0.16970	0.168300	0.087510	0.1926
163	12.340	22.22	79.85	464.5	0.10120	0.10150	0.053700	0.028220	0.1551
123	14.500	10.89	94.28	640.7	0.11010	0.10990	0.088420	0.057780	0.1856
361	13.300	21.57	85.24	546.1	0.08582	0.06373	0.033440	0.024240	0.1815
549	10.820	24.21	68.89	361.6	0.08192	0.06602	0.015480	0.008160	0.1976
339	23.510	24.27	155.10	1747.0	0.10690	0.12830	0.230800	0.141000	0.1797
286	11.940	20.76	77.87	441.0	0.08605	0.10110	0.065740	0.037910	0.1588
354	11.140	14.07	71.24	384.6	0.07274	0.06064	0.045050	0.014710	0.1690
421	14.690	13.98	98.22	656.1	0.10310	0.18360	0.145000	0.063000	0.2086
124	13.370	16.39	86.10	553.5	0.07115	0.07325	0.080920	0.028000	0.1422
543	13.210	28.06	84.88	538.4	0.08671	0.06877	0.029870	0.032750	0.1628
537	11.690	24.44	76.37	406.4	0.12360	0.15520	0.045150	0.045310	0.2131
567	20.600	29.33	140.10	1265.0	0.11780	0.27700	0.351400	0.152000	0.2397
555	10.290	27.61	65.67	321.4	0.09030	0.07658	0.059990	0.027380	0.1593
511	14.810	14.70	94.66	680.7	0.08472	0.05016	0.034160	0.025410	0.1659
333	11.250	14.78	71.38	390.0	0.08306	0.04458	0.000974	0.002941	0.1773
68	9.029	17.33	58.79	250.5	0.10660	0.14130	0.313000	0.043750	0.2111
189	12.300	15.90	78.83	463.7	0.08080	0.07253	0.038440	0.016540	0.1667
557	9.423	27.88	59.26	271.3	0.08123	0.04971	0.000000	0.000000	0.1742
436	12.870	19.54	82.67	509.2	0.09136	0.07883	0.017970	0.020900	0.1861
479	16.250	19.51	109.80	815.8	0.10260	0.18930	0.223600	0.091940	0.2151
52	11.940	18.24	75.71	437.6	0.08261	0.04751	0.019720	0.013490	0.1868
401	11.930	10.91	76.14	442.7	0.08872	0.05242	0.026060	0.017960	0.1601
355	12.560	19.07	81.92	485.8	0.08760	0.10380	0.103000	0.043910	0.1533
318	9.042	18.90	60.07	244.5	0.09968	0.19720	0.197500	0.049080	0.2330
359	9.436	18.32	59.82	278.6	0.10090	0.05956	0.027100	0.014060	0.1506
40	13.440	21.58	86.18	563.0	0.08162	0.06031	0.031100	0.020310	0.1784
323	20.340	21.51	135.90	1264.0	0.11700	0.18750	0.256500	0.150400	0.2569
495	14.870	20.21	96.12	680.9	0.09587	0.08345	0.068240	0.049510	0.1487
45	18.650	17.60	123.70	1076.0	0.10990	0.16860	0.197400	0.100900	0.1907
...	...	...	...	...	...	...	...	...	..
7	13.710	20.83	90.20	577.9	0.11890	0.16450	0.093660	0.059850	0.2196

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
155	12.250	17.94	78.27	460.3	0.08654	0.06679	0.038850	0.023310	0.1970
56	19.210	18.57	125.50	1152.0	0.10530	0.12670	0.132300	0.089940	0.1917
151	8.219	20.70	53.27	203.9	0.09405	0.13050	0.132100	0.021680	0.2222
203	13.810	23.75	91.56	597.8	0.13230	0.17680	0.155800	0.091760	0.2251
34	16.130	17.88	107.00	807.2	0.10400	0.15590	0.135400	0.077520	0.1998
417	15.500	21.08	102.90	803.1	0.11200	0.15710	0.152200	0.084810	0.2085
42	19.070	24.81	128.30	1104.0	0.09081	0.21900	0.210700	0.099610	0.2310
453	14.530	13.98	93.86	644.2	0.10990	0.09242	0.068950	0.064950	0.1650
500	15.040	16.74	98.73	689.4	0.09883	0.13640	0.077210	0.061420	0.1668
258	15.660	23.20	110.20	773.5	0.11090	0.31140	0.317600	0.137700	0.2495
369	22.010	21.90	147.20	1482.0	0.10630	0.19540	0.244800	0.150100	0.1824
313	11.540	10.72	73.73	409.1	0.08597	0.05969	0.013670	0.008907	0.1833
426	10.480	14.98	67.49	333.6	0.09816	0.10130	0.063350	0.022180	0.1925
140	9.738	11.97	61.24	288.5	0.09250	0.04102	0.000000	0.000000	0.1903
388	11.270	15.50	73.38	392.0	0.08365	0.11140	0.100700	0.027570	0.1810
116	8.950	15.76	58.74	245.2	0.09462	0.12430	0.092630	0.023080	0.1305
198	19.180	22.49	127.50	1148.0	0.08523	0.14280	0.111400	0.067720	0.1767
490	12.250	22.44	78.18	466.5	0.08192	0.05200	0.017140	0.012610	0.1544
50	11.760	21.60	74.72	427.9	0.08637	0.04966	0.016570	0.011150	0.1495
199	14.450	20.22	94.49	642.7	0.09872	0.12060	0.118000	0.059800	0.1950
366	20.200	26.83	133.70	1234.0	0.09905	0.16690	0.164100	0.126500	0.1875
455	13.380	30.72	86.34	557.2	0.09245	0.07426	0.028190	0.032640	0.1375
162	19.590	18.15	130.70	1214.0	0.11200	0.16660	0.250800	0.128600	0.2027
403	12.940	16.17	83.18	507.6	0.09879	0.08836	0.032960	0.023900	0.1735
414	15.130	29.81	96.71	719.5	0.08320	0.04605	0.046860	0.027390	0.1852
515	11.340	18.61	72.76	391.2	0.10490	0.08499	0.043020	0.025940	0.1927
186	18.310	18.58	118.60	1041.0	0.08588	0.08468	0.081690	0.058140	0.1621
3	11.420	20.38	77.58	386.1	0.14250	0.28390	0.241400	0.105200	0.2597
261	17.350	23.06	111.00	933.1	0.08662	0.06290	0.028910	0.028370	0.1564

114 rows × 30 columns

In [116]:
```
y_test
```

Out[116]:
```
28     0.0
163    1.0
123    1.0
361    1.0
549    1.0
      ...
414    0.0
515    1.0
186    0.0
3      0.0
261    0.0
Name: target, Length: 114, dtype: float64
```

In [117]:
```
from sklearn.svm import SVC
```

In [119]:
```
from sklearn.metrics import classification_report, confusion_matrix
```

In [120]:
```
svc_model = SVC()
```

In [121]:
```
svc_model.fit(X_train,y_train)
```

```
C:\Users\Nouman Rasheed\AppData\Local\Continuum\anaconda3\lib\site-packages
\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will cha
nge from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[121]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

Evaluating the Model

In [130]:
```
y_pred = svc_model.predict(X_test)
```

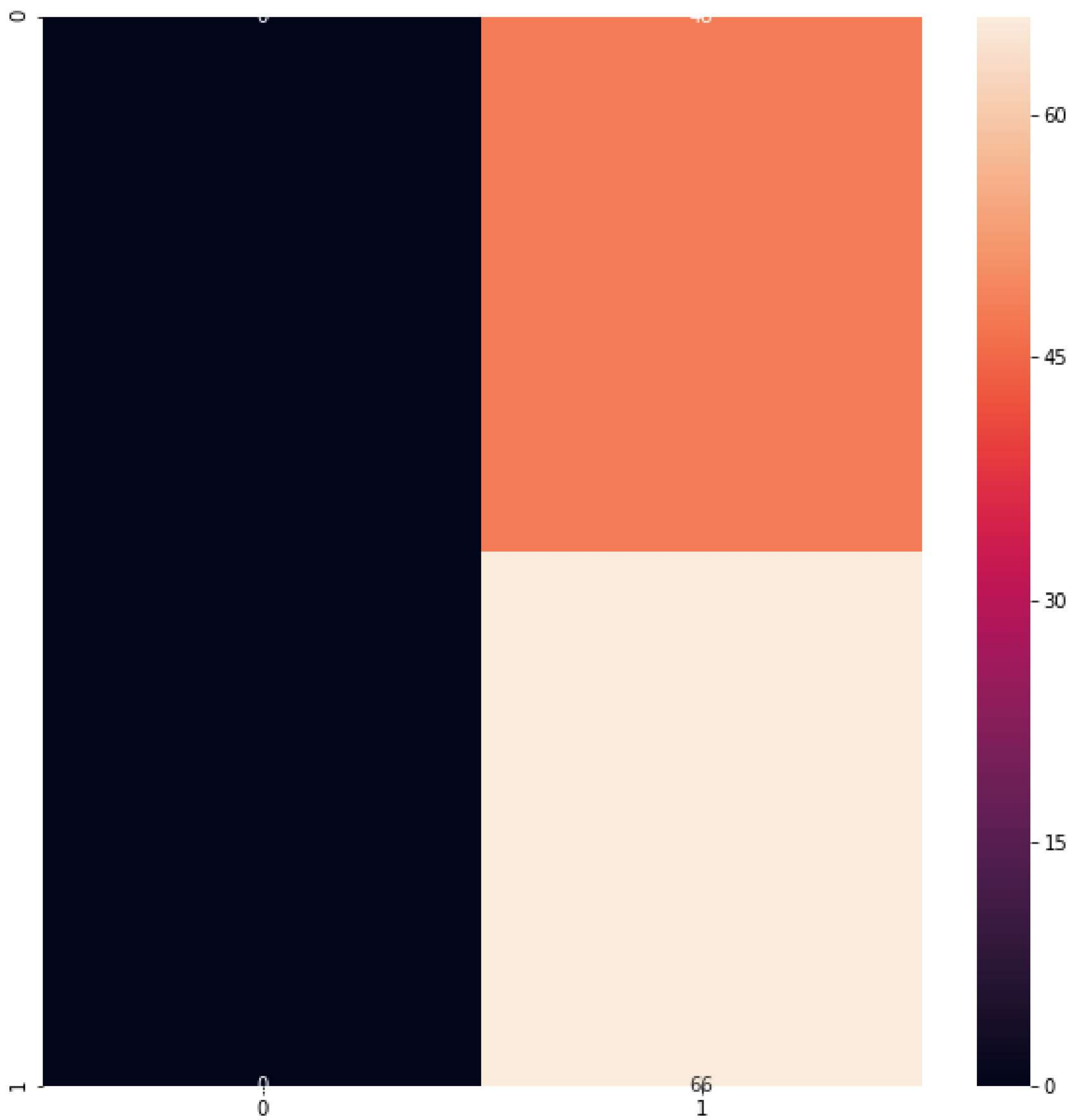
In [131]:
```
y_pred
```

Out[131]:
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

In [134]:
```
cm = confusion_matrix(y_test,y_pred)
```

In [137]:
```
plt.figure(figsize=(10,10))
sns.heatmap(cm, annot=True)
```

Out[137]: <matplotlib.axes._subplots.AxesSubplot at 0x236a056e898>



Improving The Model

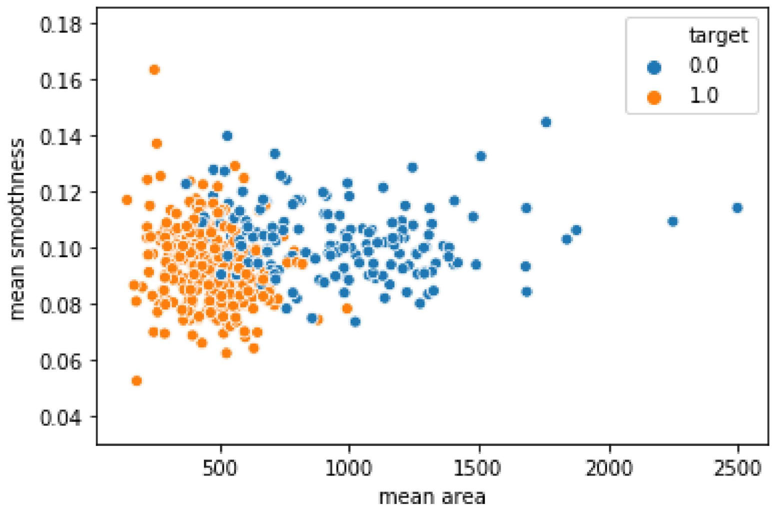
In [139]:
```
min_train = X_train.min()
```

In [141]:
```
range_train = (X_train-min_train).max()
```

In [142]:
```
X_train_scaled = (X_train-min_train) / range_train
```

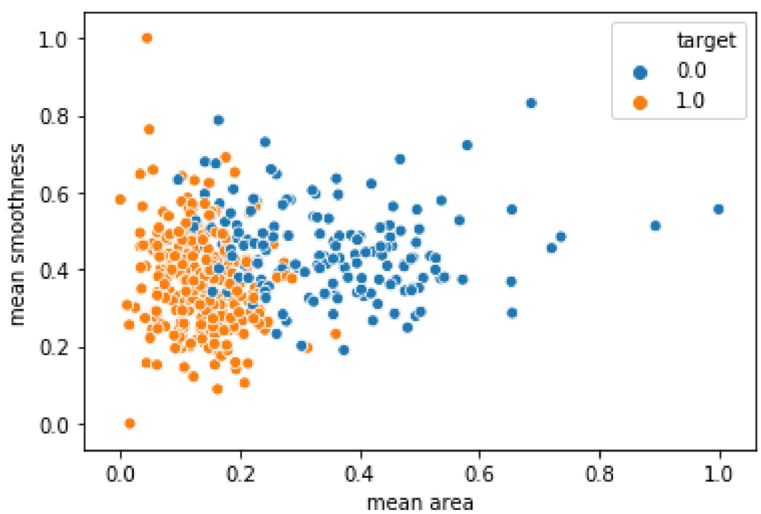
In [143]:
```
sns.scatterplot(x = X_train['mean area'], y = X_train['mean smoothness'], hue
```

Out[143]: <matplotlib.axes._subplots.AxesSubplot at 0x236a48f1080>



In [144]:
```
sns.scatterplot(x = X_train_scaled['mean area'], y = X_train_scaled['mean smo
```

Out[144]: <matplotlib.axes._subplots.AxesSubplot at 0x236a4027d68>



In [149]:
```
min_test = X_test.min()
range_test = (X_test-min_test).max()
X_test_scaled = (X_test-min_test) / range_train
```

In [153]:
```
svc_model.fit(X_train_scaled,y_train)
```

```
C:\Users\Nouman Rasheed\AppData\Local\Continuum\anaconda3\lib\site-packages
\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will cha
nge from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

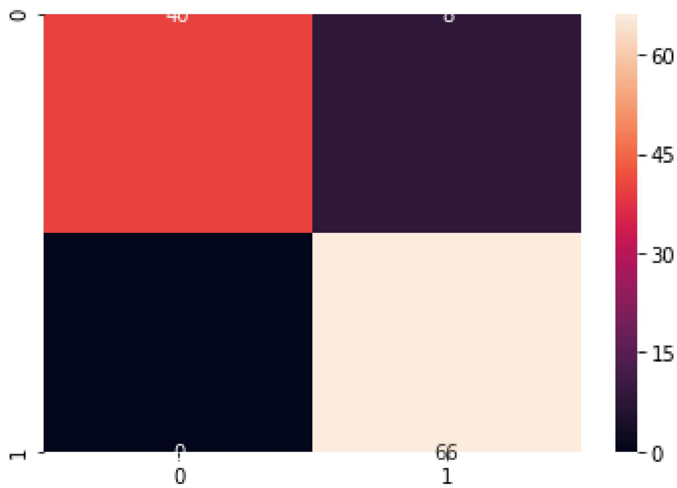
Out[153]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

In [159]:
```
y_predict = svc_model.predict(X_test_scaled)
```

In [161]:
```
cm = confusion_matrix(y_test,y_predict)
```

In [162]:
```
sns.heatmap(cm,annot = True)
```

Out[162]: <matplotlib.axes._subplots.AxesSubplot at 0x236a49d6c50>



In [163]:
```
print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.83      0.91        48
         1.0       0.89      1.00      0.94        66

    accuracy                           0.93       114
   macro avg       0.95      0.92      0.93       114
weighted avg       0.94      0.93      0.93       114
```

Improving the Model Part 2

In [230]:
```
param_grid = {'C':[0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001], 'kernel'
```

In [231]:
```python
from sklearn.model_selection import GridSearchCV
```

In [232]:
```python
grid = GridSearchCV(SVC(), param_grid, refit= True, verbose= 4)
```

In [233]:
```python
grid.fit(X_train_scaled, y_train)
```

```
[CV] C=10, gamma=1, kernel=rbf ........................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.974, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ......................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.993, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ......................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ......................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.974, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf .....................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.974, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf .....................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.921, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf .....................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.940, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ....................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.914, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ....................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.895, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ....................................
```

In [234]:
```python
grid.best_params_
```

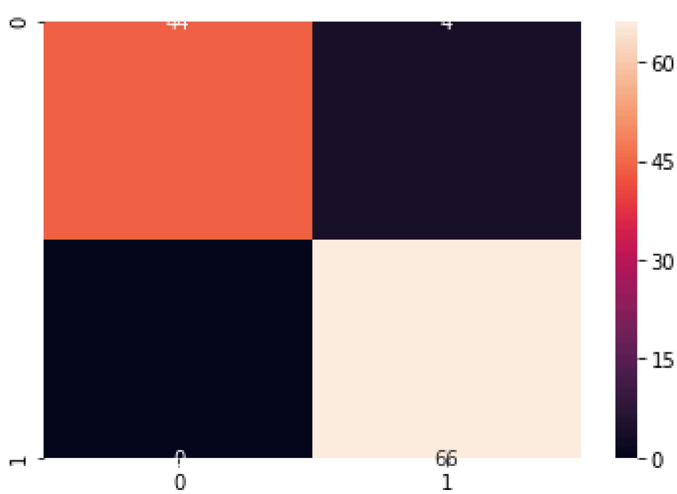
Out[234]:
```
{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
```

In [237]:
```python
grid_predictions = grid.predict(X_test_scaled)
```

In [238]:
```python
cm = confusion_matrix(y_test,grid_predictions)
```

In [239]:
```python
sns.heatmap(cm,annot=True)
```

Out[239]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x236a3fed400>
```



In [240]:
```
print(classification_report(y_test,grid_predictions))
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.92      0.96        48
         1.0       0.94      1.00      0.97        66

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114
```

In [ ]:
```

```