

Python Seaborn Tutorial Part-12

Seaborn Pairplot

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

from sklearn.datasets import load_breast_cancer
cancer_dataset = load_breast_cancer()
cancer_dataset

{'data': 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]]),
'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 1, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0,
                0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0,
0,
                1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0,
0,
                1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0,
1,
                1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1,
0,
                0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1,
                1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1,
1,
                1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0,
0,
                0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0,
0,
                1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1,
1,
```

```

0,      1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1,      0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1,
1,      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
1,      1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0,
0,      0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
0,      0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0,
0,      1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
1,      1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
0,      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1,
1,      1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
0,      1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
1,      1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1,
1,      1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1,
1,      1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1)),
'target_names': array(['malignant', 'benign'], dtype='<U9'),
'DESCR': '.. _breast_cancer_dataset:\n\nBreast cancer wisconsin
(diagnostic) dataset\n-----\n\n
**Data Set Characteristics:**\n\n      :Number of Instances: 569\n\n
:Number of Attributes: 30 numeric, predictive attributes and the
class\n\n      :Attribute Information:\n          - radius (mean of
distances from center to points on the perimeter)\n          - texture
(standard deviation of gray-scale values)\n          - perimeter\n
- area\n          - smoothness (local variation in radius lengths)\n
- compactness (perimeter^2 / area - 1.0)\n          - concavity
(severity of concave portions of the contour)\n          - concave
points (number of concave portions of the contour)\n          - symmetry
\n          - fractal dimension ("coastline approximation" - 1)\n\n
The mean, standard error, and "worst" or largest (mean of the three\n
largest values) of these features were computed for each image,\n
resulting in 30 features.  For instance, field 3 is Mean Radius,
field\n          13 is Radius SE, field 23 is Worst Radius.\n\n      -
class:\n          - WDBC-Malignant\n          - WDBC-
Benign\n\n      :Summary Statistics:\n\n
===== \n
Min      Max\n      ===== \n

```

radius (mean):	6.981	28.11	\n	texture
(mean):	9.71	39.28	\n	perimeter (mean):
43.79	188.5	\n	area (mean):	143.5
2501.0	\n	smoothness (mean):	0.053	0.163
\n	compactness (mean):	0.019	0.345	\n
concavity	(mean):	0.0	0.427	\n
concave points (mean):	0.106	0.304	\n	0.0
0.201	\n	symmetry (mean):	0.05	0.097
\n	fractal dimension (mean):	0.112	2.873	\n
radius	(standard error):	0.36	4.885	\n
texture (standard	error):	0.757	21.98	\n
perimeter (standard error):	6.802	542.2	\n	0.757
21.98	\n	area (standard error):	0.002	0.031
\n	smoothness (standard error):	0.002	0.135	\n
compactness	(standard error):	0.0	0.396	\n
concavity (standard	error):	0.0	0.053	\n
concave points (standard error):	0.008	0.079	\n	0.0
0.053	\n	symmetry (standard error):	0.001	0.03
\n	fractal dimension (standard error):	7.93	36.04	\n
texture (worst):	12.02	49.54	\n	7.93
36.04	\n	perimeter (worst):	50.41	251.2
\n	area (worst):	185.2	4254.0	\n
smoothness (worst):	0.071	0.223	\n	185.2
4254.0	\n	compactness (worst):	0.027	1.058
\n	concavity	(worst):	0.0	1.252
0.0	\n	concave points (worst):	0.156	0.664
0.291	\n	symmetry (worst):	0.055	0.208
\n	fractal dimension (worst):	0.055	0.208	\n

=====\n\n :Missing
Attribute Values: None\n\n :Class Distribution: 212 - Malignant,
357 - Benign\n\n :Creator: Dr. William H. Wolberg, W. Nick Street,
Olvi L. Mangasarian\n\n :Donor: Nick Street\n\n :Date: November,
1995\n\n This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic)
datasets.\n\n <https://goo.gl/U2Uwz2>\n\n Features are computed from a
digitized image of a fine needle\n\n aspirate (FNA) of a breast mass.
They describe\n\n characteristics of the cell nuclei present in the
image.\n\n Separating plane described above was obtained using\n\n
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree\n\n
Construction Via Linear Programming." Proceedings of the 4th\n\n
Midwest Artificial Intelligence and Cognitive Science Society,\n\n
pp. 97-101, 1992], a classification method which uses linear\n\n
programming to construct a decision tree. Relevant features\n\n
were selected using an exhaustive search in the space of 1-4\n\n
features and 1-3 separating planes.\n\n
The actual linear program used to obtain the separating
plane\n\n in the 3-dimensional space is that described in:\n\n
[K. P. Bennett and O. L. Mangasarian: "Robust Linear\n\n
Programming Discrimination of Two Linearly Inseparable Sets",\n\n
Optimization Methods and Software 1, 1992, 23-34].\n\n
This database is also available through the UW CS ftp server:\n\n
ftp ftp.cs.wisc.edu\n\n
ncd math-prog/cpo-dataset/machine-learn/WDBC/\n\n
.. topic:: References\n\n
- W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature
extraction \n for breast tumor diagnosis. IS&T/SPIE 1993
International Symposium on \n Electronic Imaging: Science and
Technology, volume 1905, pages 861-870,\n San Jose, CA, 1993.\n
- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer

```

diagnosis and \n      prognosis via linear programming. Operations
Research, 43(4), pages 570-577, \n      July-August 1995.\n      - W.H.
Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning
techniques\n      to diagnose breast cancer from fine-needle aspirates.
Cancer Letters 77 (1994) \n      163-171.',
'feature_names': 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'),
'filename': 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\
sklearn\\datasets\\data\\breast_cancer.csv'}

```

```
# create dataframes
```

```

cancer_df =
pd.DataFrame(np.c_[cancer_dataset['data'],cancer_dataset['target']],
              columns = np.append(cancer_dataset['feature_names'],
['target']))
cancer_df

```

	mean radius	mean texture	mean perimeter	mean area	mean
smoothness \					
0	17.99	10.38	122.80	1001.0	
0.11840					
1	20.57	17.77	132.90	1326.0	
0.08474					
2	19.69	21.25	130.00	1203.0	
0.10960					
3	11.42	20.38	77.58	386.1	
0.14250					
4	20.29	14.34	135.10	1297.0	
0.10030					
..	
...					
564	21.56	22.39	142.00	1479.0	
0.11100					
565	20.13	28.25	131.20	1261.0	
0.09780					
566	16.60	28.08	108.30	858.1	
0.08455					
567	20.60	29.33	140.10	1265.0	
0.11780					

568	7.76	24.54	47.92	181.0
0.05263				

	mean compactness	mean concavity	mean concave points	mean symmetry \
0	0.27760	0.30010	0.14710	0.2419
1	0.07864	0.08690	0.07017	0.1812
2	0.15990	0.19740	0.12790	0.2069
3	0.28390	0.24140	0.10520	0.2597
4	0.13280	0.19800	0.10430	0.1809
...
564	0.11590	0.24390	0.13890	0.1726
565	0.10340	0.14400	0.09791	0.1752
566	0.10230	0.09251	0.05302	0.1590
567	0.27700	0.35140	0.15200	0.2397
568	0.04362	0.00000	0.00000	0.1587

	mean fractal dimension	...	worst texture	worst perimeter
worst area \				
0	0.07871	...	17.33	184.60
1	0.05667	...	23.41	158.80
2	0.05999	...	25.53	152.50
3	0.09744	...	26.50	98.87
4	0.05883	...	16.67	152.20
...
564	0.05623	...	26.40	166.10
565	0.05533	...	38.25	155.00
566	0.05648	...	34.12	126.70
567	0.07016	...	39.42	184.60

```

568          0.05884    ...          30.37          59.16
268.6

```

```

      worst smoothness  worst compactness  worst concavity  \
0          0.16220          0.66560          0.7119
1          0.12380          0.18660          0.2416
2          0.14440          0.42450          0.4504
3          0.20980          0.86630          0.6869
4          0.13740          0.20500          0.4000
..          ...          ...          ...
564         0.14100          0.21130          0.4107
565         0.11660          0.19220          0.3215
566         0.11390          0.30940          0.3403
567         0.16500          0.86810          0.9387
568         0.08996          0.06444          0.0000

```

```

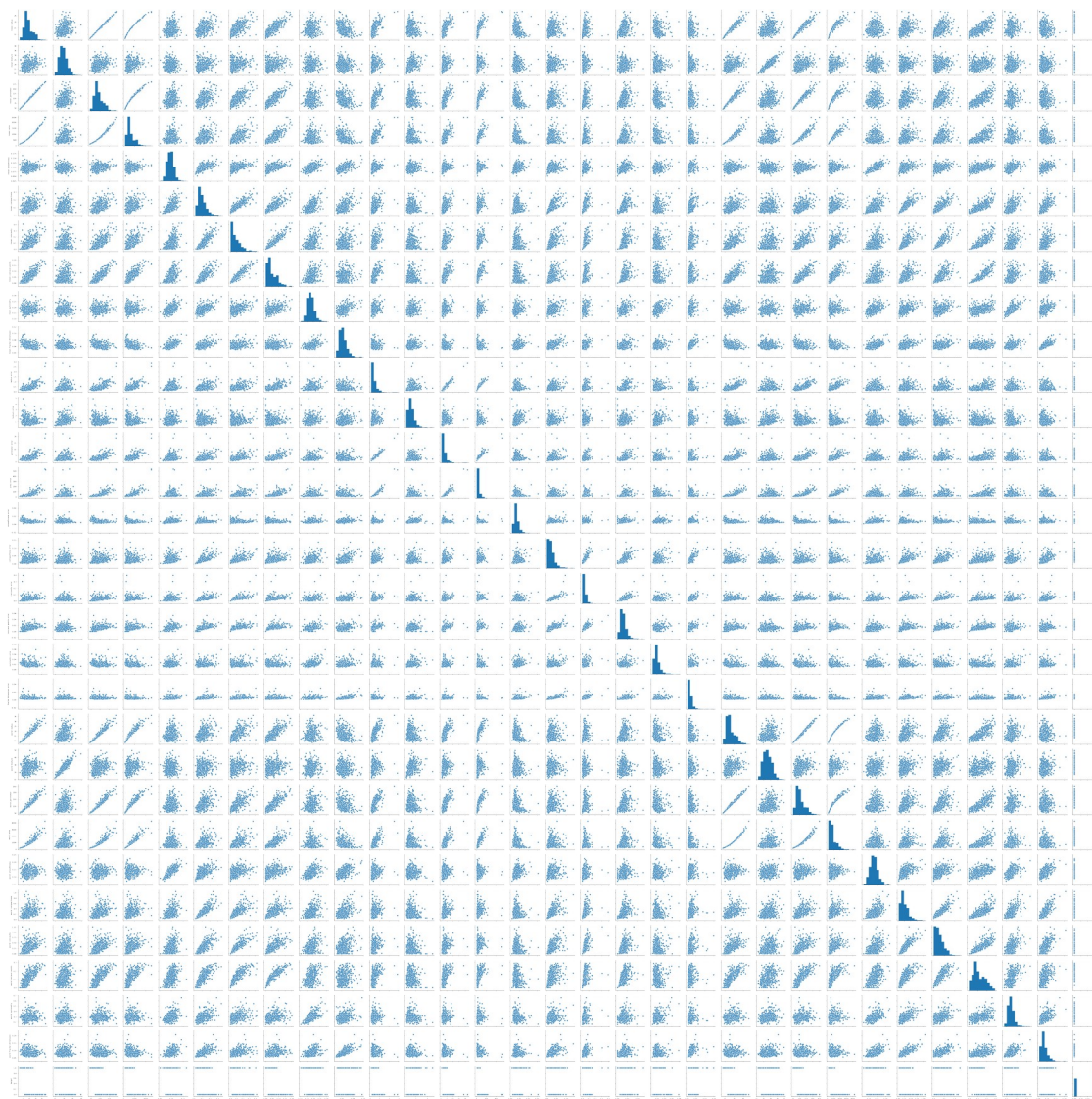
      worst concave points  worst symmetry  worst fractal dimension
target
0          0.2654          0.4601          0.11890
0.0
1          0.1860          0.2750          0.08902
0.0
2          0.2430          0.3613          0.08758
0.0
3          0.2575          0.6638          0.17300
0.0
4          0.1625          0.2364          0.07678
0.0
..          ...          ...          ...
...
564         0.2216          0.2060          0.07115
0.0
565         0.1628          0.2572          0.06637
0.0
566         0.1418          0.2218          0.07820
0.0
567         0.2650          0.4087          0.12400
0.0
568         0.0000          0.2871          0.07039
1.0

```

```
[569 rows x 31 columns]
```

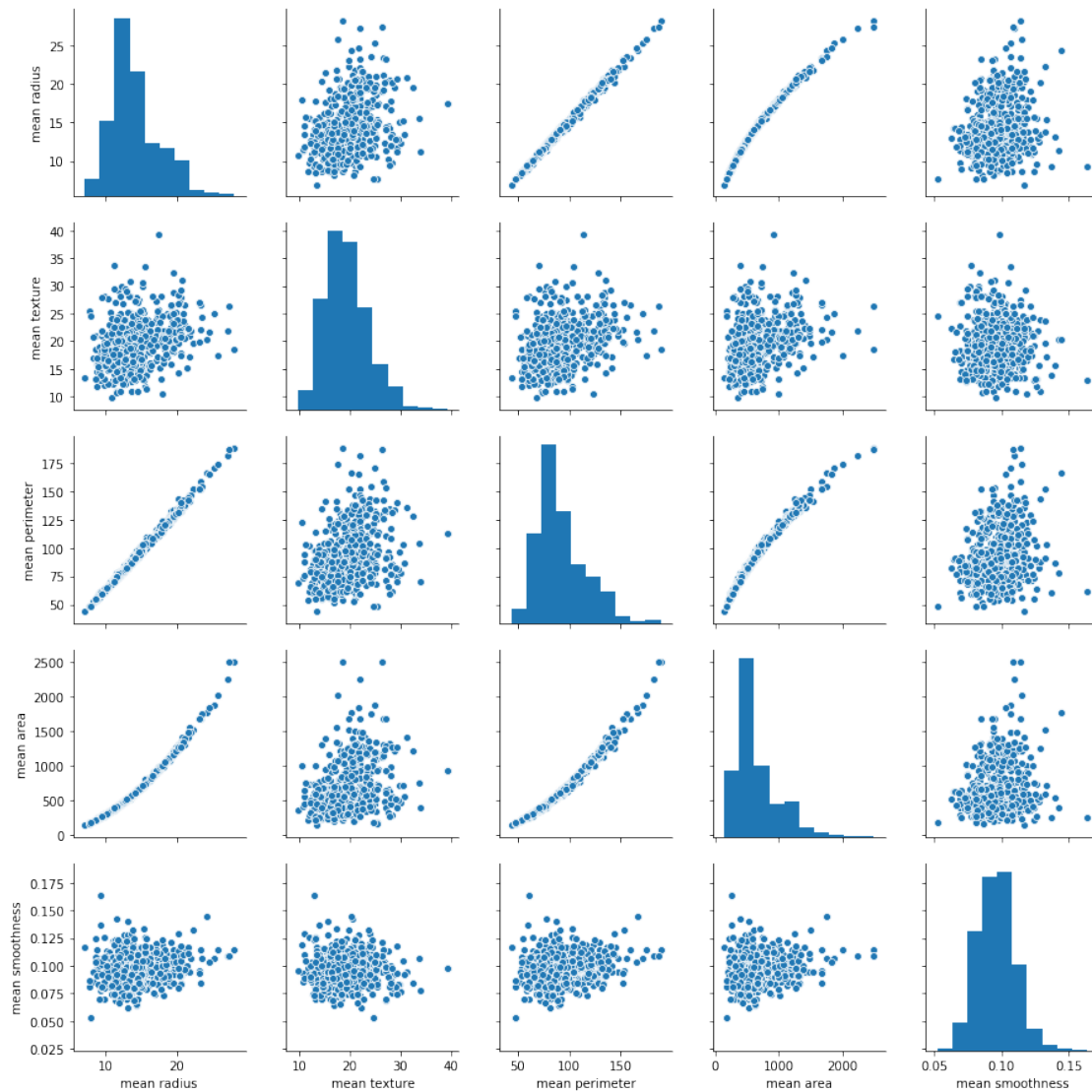
```
sns.pairplot(cancer_df)
```

```
<seaborn.axisgrid.PairGrid at 0x12c8c6cc940>
```



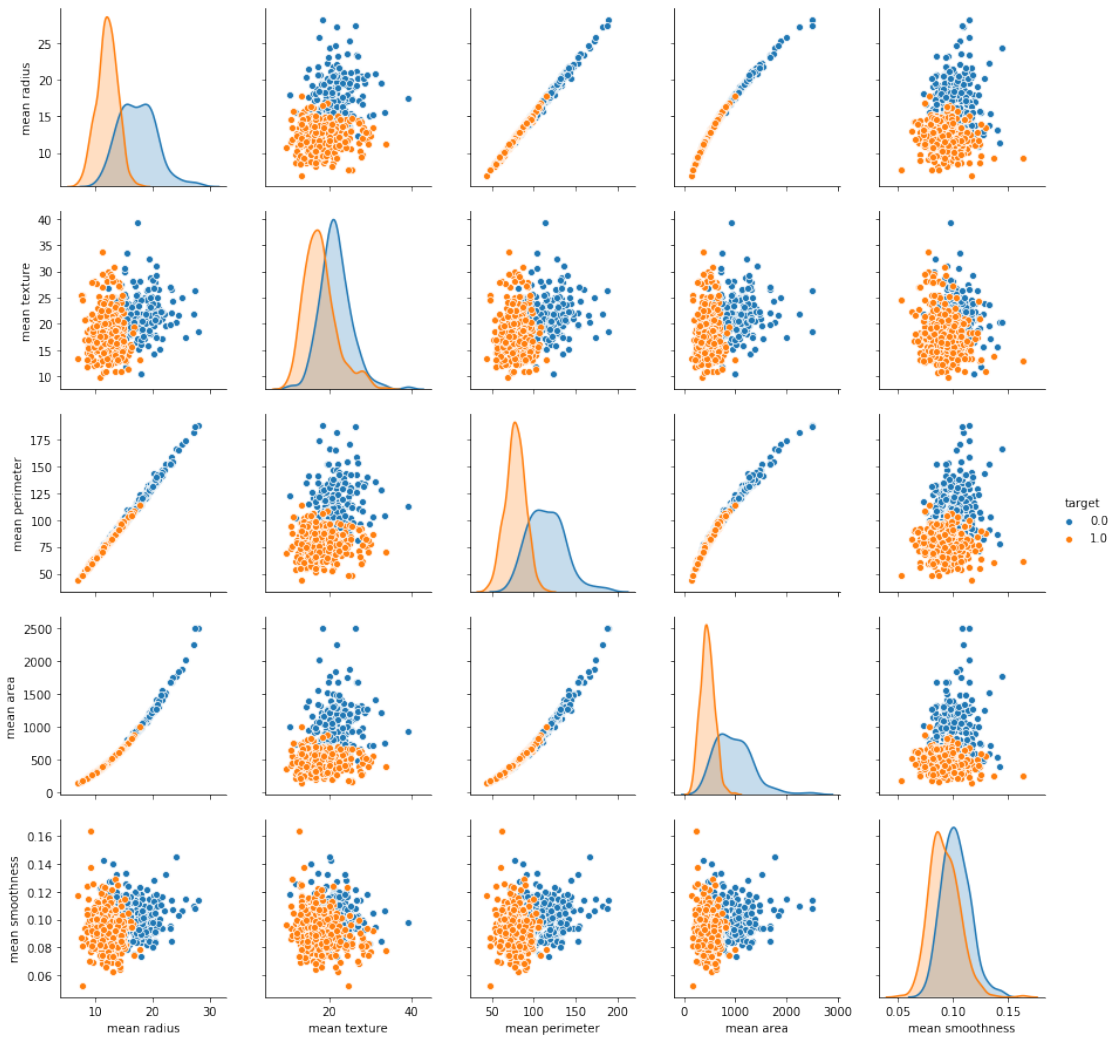
```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean  
perimeter', 'mean area',  
                                'mean smoothness'])
```

```
<seaborn.axisgrid.PairGrid at 0x12cb3a38dd8>
```



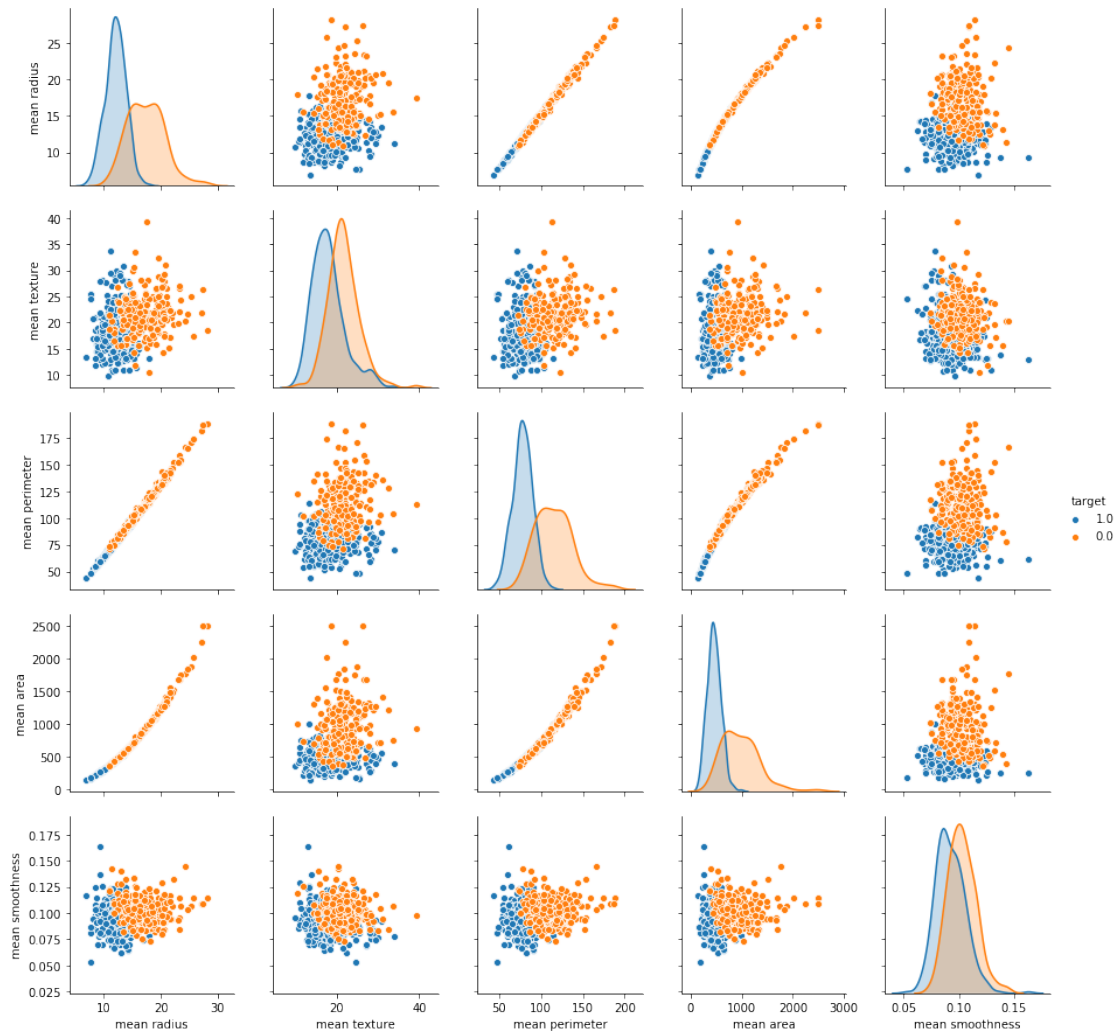
```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean
perimeter', 'mean area',
    'mean smoothness'], hue = 'target')
```

<seaborn.axisgrid.PairGrid at 0x12cbbdc3550>



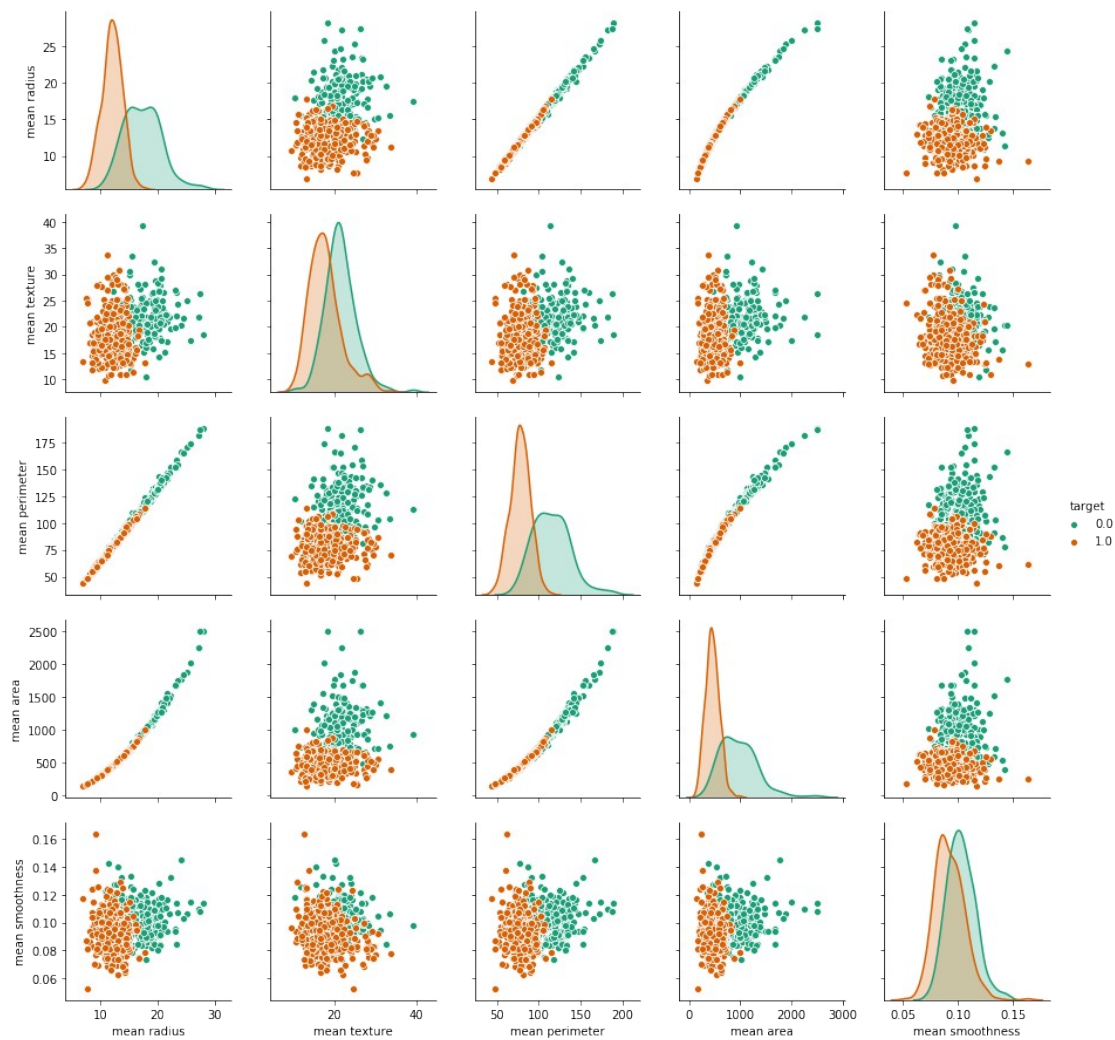
```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean
perimeter', 'mean area',
    'mean smoothness'], hue='target', hue_order = [1.0, 0.0])
```

<seaborn.axisgrid.PairGrid at 0x12cbca53208>



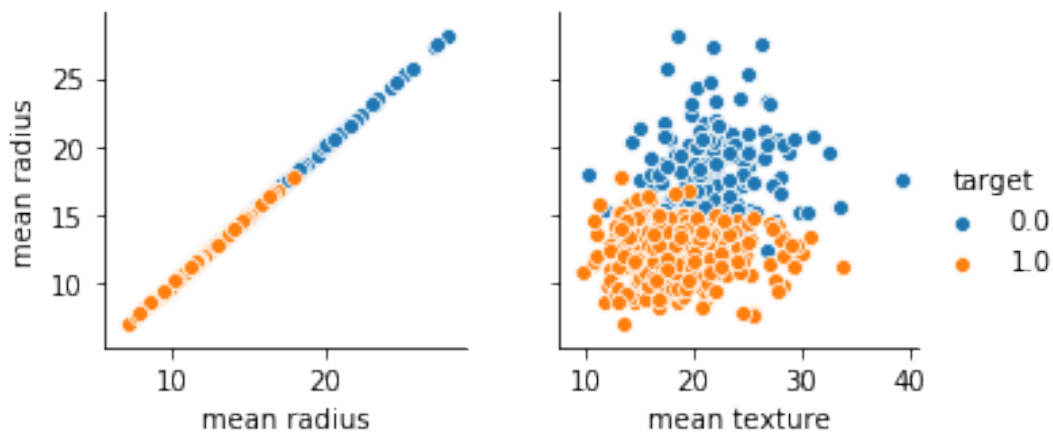
```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean
perimeter', 'mean area',
    'mean smoothness'], hue = 'target', palette='Dark2')
```

<seaborn.axisgrid.PairGrid at 0x12cbf1454e0>



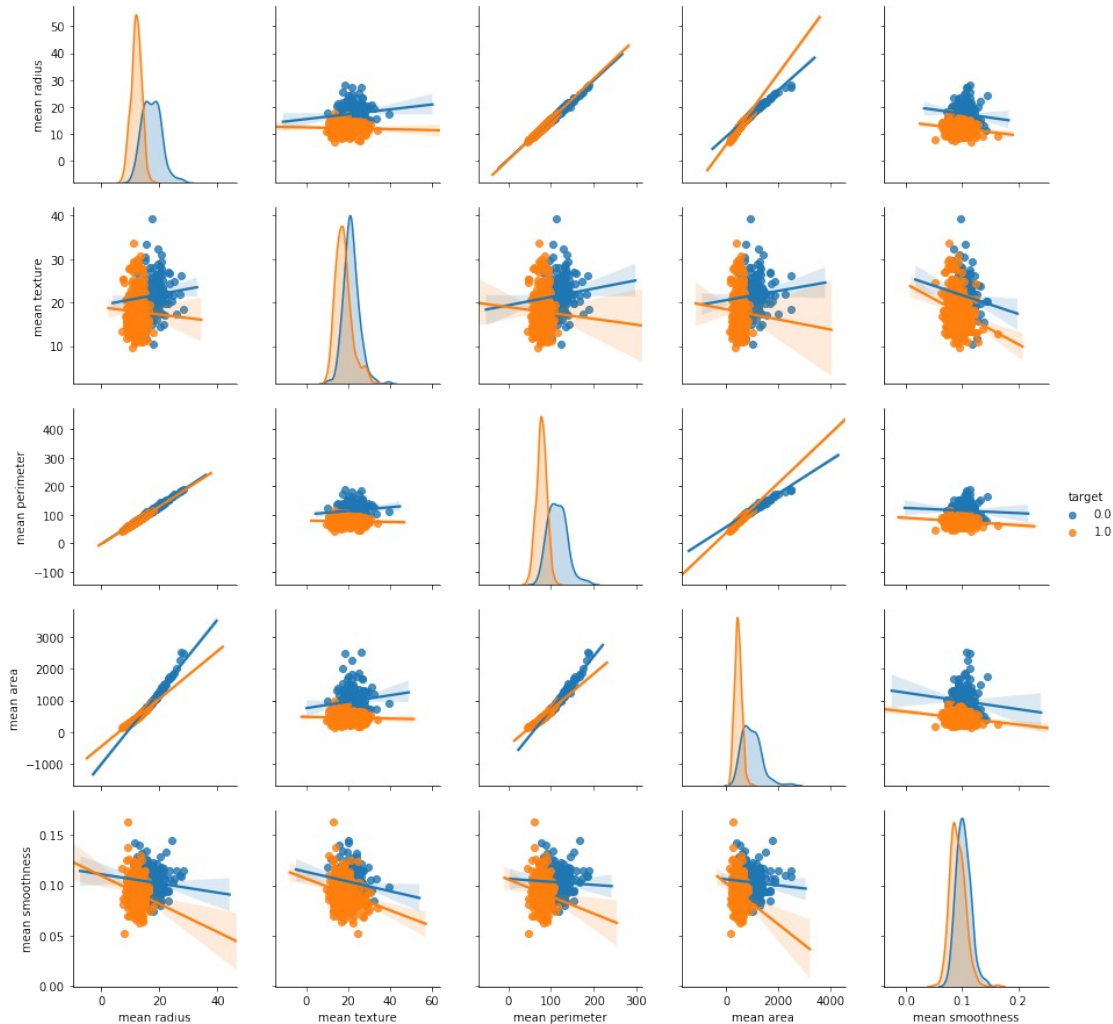
```
sns.pairplot(cancer_df, hue='target', x_vars = ['mean radius', 'mean texture'], y_vars=['mean radius'])
```

<seaborn.axisgrid.PairGrid at 0x12cc019bd30>



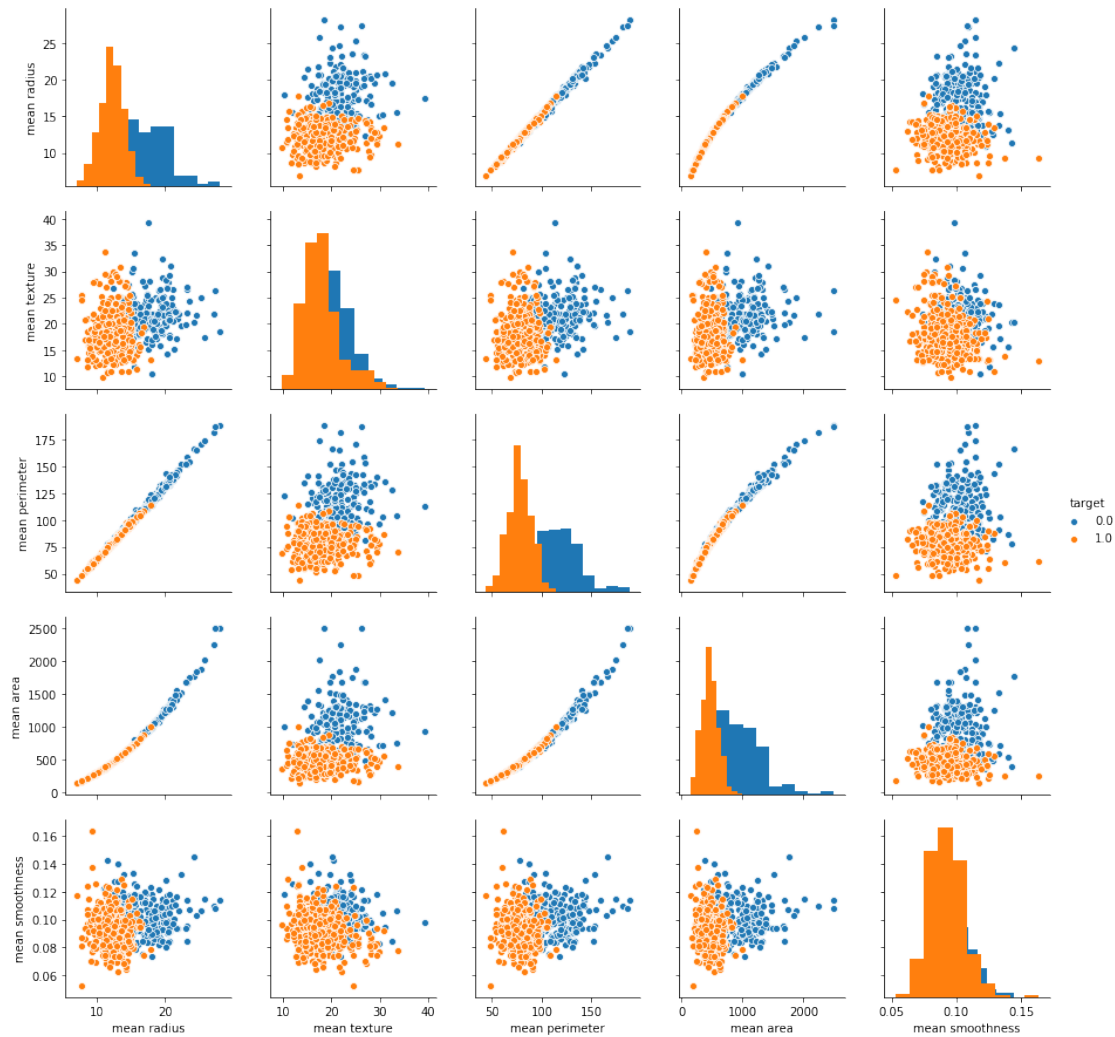
```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean
perimeter', 'mean area',
    'mean smoothness'], hue = 'target', kind = 'reg')
```

<seaborn.axisgrid.PairGrid at 0x12cbfe762b0>



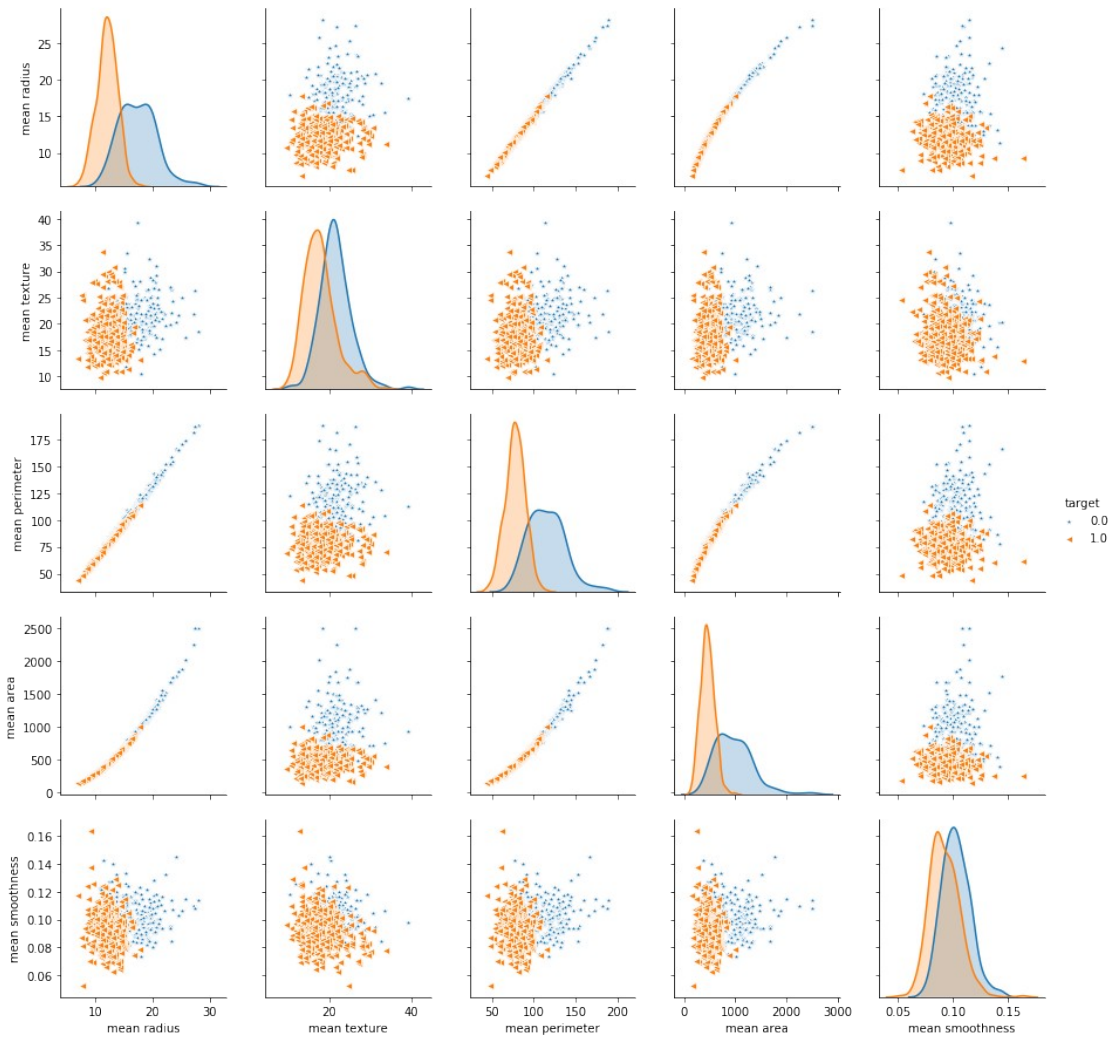
```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean
perimeter', 'mean area',
    'mean smoothness'], hue = 'target', diag_kind = 'hist')
```

<seaborn.axisgrid.PairGrid at 0x12cc1f72c50>



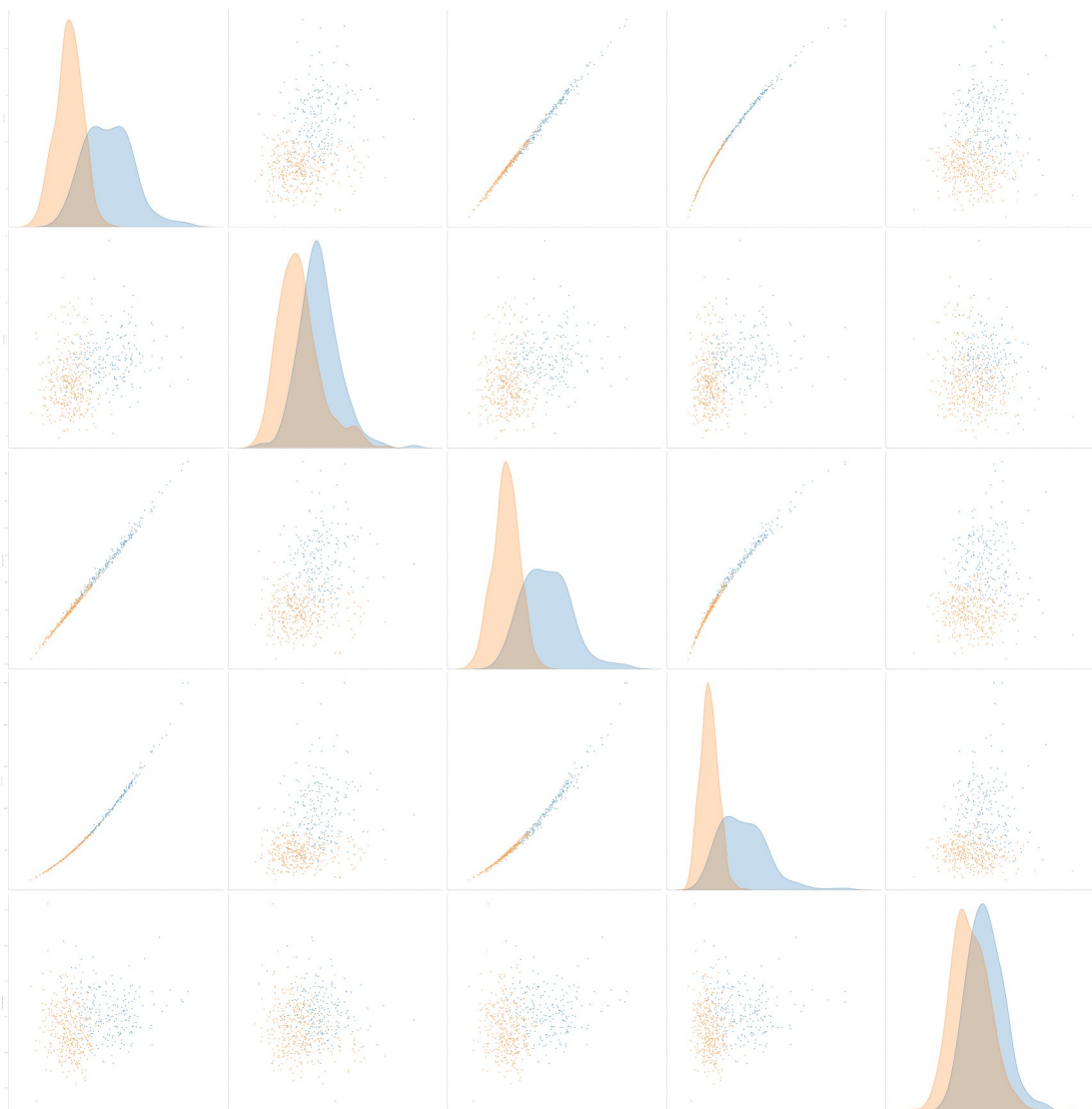
```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean
perimeter', 'mean area',
    'mean smoothness'], hue = 'target', markers = ['*', "<"])
```

<seaborn.axisgrid.PairGrid at 0x12cc3467e10>



```
sns.pairplot(cancer_df, vars = ['mean radius', 'mean texture', 'mean
perimeter', 'mean area',
    'mean smoothness'], hue = 'target', height = 20)
```

<seaborn.axisgrid.PairGrid at 0x12cc406c080>



```
print("Thank you -:)" )
```

Thank you -:)