breast-cancer-classification

January 24, 2024

Importing the Dependencies

[127]:
```
import numpy as np
import pandas as pd
import sklearn.datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

Data Collection & Processing

[128]:
```
# loading the data from sklearn
breast_cancer_dataset = sklearn.datasets.load_breast_cancer()
```

[129]:
```
print(breast_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, 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,
       1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       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, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
       1, 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, 1, 0, 0,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 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, 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, 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, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1]), 'frame': None,
'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        worst/largest values) of
these features were computed for each image,\n        resulting in 30 features.
For instance, field 0 is Mean Radius, field\n        10 is Radius SE, field 20
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):
0.0    0.201\n    symmetry (mean):                       0.106  0.304\n
fractal dimension (mean):             0.05   0.097\n    radius (standard error):
0.112  2.873\n    texture (standard error):              0.36   4.885\n
perimeter (standard error):           0.757  21.98\n    area (standard error):
6.802  542.2\n    smoothness (standard error):           0.002  0.031\n
compactness (standard error):         0.002  0.135\n    concavity (standard
error):           0.0    0.396\n    concave points (standard error):      0.0
0.053\n    symmetry (standard error):             0.008  0.079\n    fractal
```

dimension (standard error):   0.001 0.03\n   radius (worst):
7.93   36.04\n   texture (worst):   12.02 49.54\n
perimeter (worst):   50.41 251.2\n   area (worst):
185.2 4254.0\n   smoothness (worst):   0.071 0.223\n
compactness (worst):   0.027 1.058\n   concavity (worst):
0.0   1.252\n   concave points (worst):   0.0   0.291\n
symmetry (worst):   0.156 0.664\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\nThis is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic)
datasets.\nhttps://goo.gl/U2Uwz2\n\nFeatures are computed from a digitized image
of a fine needle\naspirate (FNA) of a breast mass. They
describe\ncharacteristics of the cell nuclei present in the image.\n\nSeparating
plane described above was obtained using\nMultisurface Method-Tree (MSM-T) [K.
P. Bennett, "Decision Tree\nConstruction Via Linear Programming." Proceedings of
the 4th\nMidwest Artificial Intelligence and Cognitive Science Society,\npp.
97-101, 1992], a classification method which uses linear\nprogramming to
construct a decision tree. Relevant features\nwere selected using an exhaustive
search in the space of 1-4\nfeatures and 1-3 separating planes.\n\nThe actual
linear program used to obtain the separating plane\nin the 3-dimensional space
is that described in:\n[K. P. Bennett and O. L. Mangasarian: "Robust
Linear\nProgramming Discrimination of Two Linearly Inseparable
Sets",\nOptimization Methods and Software 1, 1992, 23-34].\n\nThis database is
also available through the UW CS ftp server:\n\nftp ftp.cs.wisc.edu\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':
'breast_cancer.csv', 'data_module': 'sklearn.datasets.data'}

[130]:
```
# loading the data to a data frame
data_frame = pd.DataFrame(breast_cancer_dataset.data, columns =␣
 ↪breast_cancer_dataset.feature_names)
```

[131]:
```
# print the first 5 rows of the dataframe
data_frame.head()
```

[131]:

	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	

	mean compactness	mean concavity	mean concave points	mean symmetry	\
0	0.27760	0.3001	0.14710	0.2419	
1	0.07864	0.0869	0.07017	0.1812	
2	0.15990	0.1974	0.12790	0.2069	
3	0.28390	0.2414	0.10520	0.2597	
4	0.13280	0.1980	0.10430	0.1809	

	mean fractal dimension	…	worst radius	worst texture	worst perimeter	\
0	0.07871	…	25.38	17.33	184.60	
1	0.05667	…	24.99	23.41	158.80	
2	0.05999	…	23.57	25.53	152.50	
3	0.09744	…	14.91	26.50	98.87	
4	0.05883	…	22.54	16.67	152.20	

	worst area	worst smoothness	worst compactness	worst concavity	\
0	2019.0	0.1622	0.6656	0.7119	
1	1956.0	0.1238	0.1866	0.2416	
2	1709.0	0.1444	0.4245	0.4504	
3	567.7	0.2098	0.8663	0.6869	
4	1575.0	0.1374	0.2050	0.4000	

	worst concave points	worst symmetry	worst fractal dimension
0	0.2654	0.4601	0.11890
1	0.1860	0.2750	0.08902
2	0.2430	0.3613	0.08758
3	0.2575	0.6638	0.17300
4	0.1625	0.2364	0.07678

[5 rows x 30 columns]

[132]:
```
# adding the 'target' column to the data frame
data_frame['label'] = breast_cancer_dataset.target
```

[133]:
```
# print last 5 rows of the dataframe
data_frame.tail()
```

[133]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	\
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	\
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	\
564	0.05623	…	26.40	166.10	2027.0	
565	0.05533	…	38.25	155.00	1731.0	
566	0.05648	…	34.12	126.70	1124.0	
567	0.07016	…	39.42	184.60	1821.0	
568	0.05884	…	30.37	59.16	268.6	

	worst smoothness	worst compactness	worst concavity	\
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	label
564	0.2216	0.2060	0.07115	0
565	0.1628	0.2572	0.06637	0
566	0.1418	0.2218	0.07820	0
567	0.2650	0.4087	0.12400	0
568	0.0000	0.2871	0.07039	1

[5 rows x 31 columns]

[134]:
```
# number of rows and columns in the dataset
data_frame.shape
```

[134]: (569, 31)

[135]:
```
# getting some information about the data
data_frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   mean radius              569 non-null    float64
 1   mean texture             569 non-null    float64
 2   mean perimeter           569 non-null    float64
 3   mean area                569 non-null    float64
 4   mean smoothness          569 non-null    float64
 5   mean compactness         569 non-null    float64
 6   mean concavity           569 non-null    float64
 7   mean concave points      569 non-null    float64
 8   mean symmetry            569 non-null    float64
 9   mean fractal dimension   569 non-null    float64
 10  radius error             569 non-null    float64
 11  texture error            569 non-null    float64
 12  perimeter error          569 non-null    float64
 13  area error               569 non-null    float64
 14  smoothness error         569 non-null    float64
 15  compactness error        569 non-null    float64
 16  concavity error          569 non-null    float64
 17  concave points error     569 non-null    float64
 18  symmetry error           569 non-null    float64
 19  fractal dimension error  569 non-null    float64
 20  worst radius             569 non-null    float64
 21  worst texture            569 non-null    float64
 22  worst perimeter          569 non-null    float64
 23  worst area               569 non-null    float64
 24  worst smoothness         569 non-null    float64
 25  worst compactness        569 non-null    float64
 26  worst concavity          569 non-null    float64
 27  worst concave points     569 non-null    float64
 28  worst symmetry           569 non-null    float64
 29  worst fractal dimension  569 non-null    float64
 30  label                    569 non-null    int64
dtypes: float64(30), int64(1)
memory usage: 137.9 KB
```

[136]:
```
# checking for missing values
data_frame.isnull().sum()
```

[136]:
```
mean radius                0
mean texture               0
mean perimeter             0
mean area                  0
mean smoothness            0
```

```
mean compactness           0
mean concavity             0
mean concave points        0
mean symmetry              0
mean fractal dimension     0
radius error               0
texture error              0
perimeter error            0
area error                 0
smoothness error           0
compactness error          0
concavity error            0
concave points error       0
symmetry error             0
fractal dimension error    0
worst radius               0
worst texture              0
worst perimeter            0
worst area                 0
worst smoothness           0
worst compactness          0
worst concavity            0
worst concave points       0
worst symmetry             0
worst fractal dimension    0
label                      0
dtype: int64
```

[137]:
```
# statistical measures about the data
data_frame.describe()
```

[137]:

	mean radius	mean texture	mean perimeter	mean area	\
count	569.000000	569.000000	569.000000	569.000000	
mean	14.127292	19.289649	91.969033	654.889104	
std	3.524049	4.301036	24.298981	351.914129	
min	6.981000	9.710000	43.790000	143.500000	
25%	11.700000	16.170000	75.170000	420.300000	
50%	13.370000	18.840000	86.240000	551.100000	
75%	15.780000	21.800000	104.100000	782.700000	
max	28.110000	39.280000	188.500000	2501.000000	

	mean smoothness	mean compactness	mean concavity	mean concave points	\
count	569.000000	569.000000	569.000000	569.000000	
mean	0.096360	0.104341	0.088799	0.048919	
std	0.014064	0.052813	0.079720	0.038803	
min	0.052630	0.019380	0.000000	0.000000	
25%	0.086370	0.064920	0.029560	0.020310	

```
50%          0.095870          0.092630        0.061540             0.033500
75%          0.105300          0.130400        0.130700             0.074000
max          0.163400          0.345400        0.426800             0.201200

       mean symmetry  mean fractal dimension  …  worst texture  \
count     569.000000              569.000000  …     569.000000
mean        0.181162                0.062798  …      25.677223
std         0.027414                0.007060  …       6.146258
min         0.106000                0.049960  …      12.020000
25%         0.161900                0.057700  …      21.080000
50%         0.179200                0.061540  …      25.410000
75%         0.195700                0.066120  …      29.720000
max         0.304000                0.097440  …      49.540000

       worst perimeter   worst area  worst smoothness  worst compactness  \
count       569.000000   569.000000        569.000000         569.000000
mean        107.261213   880.583128          0.132369           0.254265
std          33.602542   569.356993          0.022832           0.157336
min          50.410000   185.200000          0.071170           0.027290
25%          84.110000   515.300000          0.116600           0.147200
50%          97.660000   686.500000          0.131300           0.211900
75%         125.400000  1084.000000          0.146000           0.339100
max         251.200000  4254.000000          0.222600           1.058000

       worst concavity  worst concave points  worst symmetry  \
count       569.000000            569.000000      569.000000
mean          0.272188              0.114606        0.290076
std           0.208624              0.065732        0.061867
min           0.000000              0.000000        0.156500
25%           0.114500              0.064930        0.250400
50%           0.226700              0.099930        0.282200
75%           0.382900              0.161400        0.317900
max           1.252000              0.291000        0.663800

       worst fractal dimension       label
count               569.000000  569.000000
mean                  0.083946    0.627417
std                   0.018061    0.483918
min                   0.055040    0.000000
25%                   0.071460    0.000000
50%                   0.080040    1.000000
75%                   0.092080    1.000000
max                   0.207500    1.000000

[8 rows x 31 columns]
```

[138]:
```
# checking the distribution of Target Varibale
data_frame['label'].value_counts()
```

[138]:
```
1    357
0    212
Name: label, dtype: int64
```

1 –> Benign

0 –> Malignant

[139]:
```
data_frame.groupby('label').mean()
```

[139]:
```
       mean radius  mean texture  mean perimeter   mean area  mean smoothness  \
label
0        17.462830     21.604906      115.365377  978.376415         0.102898
1        12.146524     17.914762       78.075406  462.790196         0.092478

       mean compactness  mean concavity  mean concave points  mean symmetry  \
label
0              0.145188        0.160775             0.087990       0.192909
1              0.080085        0.046058             0.025717       0.174186

       mean fractal dimension  …  worst radius  worst texture  \
label                          …
0                    0.062680  …     21.134811      29.318208
1                    0.062867  …     13.379801      23.515070

       worst perimeter   worst area  worst smoothness  worst compactness  \
label
0           141.370330  1422.286321          0.144845           0.374824
1            87.005938   558.899440          0.124959           0.182673

       worst concavity  worst concave points  worst symmetry  \
label
0             0.450606              0.182237        0.323468
1             0.166238              0.074444        0.270246

       worst fractal dimension
label
0                     0.091530
1                     0.079442

[2 rows x 30 columns]
```

1 Statistical Analysis of the Data

[140]:
```
description=data_frame.describe()
print(f"description of the dataset: ")
print(description)
```

```
description of the dataset:
       mean radius  mean texture  mean perimeter    mean area  \
count   569.000000    569.000000      569.000000   569.000000
mean     14.127292     19.289649       91.969033   654.889104
std       3.524049      4.301036       24.298981   351.914129
min       6.981000      9.710000       43.790000   143.500000
25%      11.700000     16.170000       75.170000   420.300000
50%      13.370000     18.840000       86.240000   551.100000
75%      15.780000     21.800000      104.100000   782.700000
max      28.110000     39.280000      188.500000  2501.000000

       mean smoothness  mean compactness  mean concavity  mean concave points  \
count       569.000000        569.000000      569.000000           569.000000
mean          0.096360          0.104341        0.088799             0.048919
std           0.014064          0.052813        0.079720             0.038803
min           0.052630          0.019380        0.000000             0.000000
25%           0.086370          0.064920        0.029560             0.020310
50%           0.095870          0.092630        0.061540             0.033500
75%           0.105300          0.130400        0.130700             0.074000
max           0.163400          0.345400        0.426800             0.201200

       mean symmetry  mean fractal dimension  …  worst texture  \
count     569.000000              569.000000  …     569.000000
mean        0.181162                0.062798  …      25.677223
std         0.027414                0.007060  …       6.146258
min         0.106000                0.049960  …      12.020000
25%         0.161900                0.057700  …      21.080000
50%         0.179200                0.061540  …      25.410000
75%         0.195700                0.066120  …      29.720000
max         0.304000                0.097440  …      49.540000

       worst perimeter   worst area  worst smoothness  worst compactness  \
count       569.000000   569.000000        569.000000         569.000000
mean        107.261213   880.583128          0.132369           0.254265
std          33.602542   569.356993          0.022832           0.157336
min          50.410000   185.200000          0.071170           0.027290
25%          84.110000   515.300000          0.116600           0.147200
50%          97.660000   686.500000          0.131300           0.211900
75%         125.400000  1084.000000          0.146000           0.339100
max         251.200000  4254.000000          0.222600           1.058000
```

```
       worst concavity  worst concave points  worst symmetry  \
count       569.000000            569.000000      569.000000
mean          0.272188              0.114606        0.290076
std           0.208624              0.065732        0.061867
min           0.000000              0.000000        0.156500
25%           0.114500              0.064930        0.250400
50%           0.226700              0.099930        0.282200
75%           0.382900              0.161400        0.317900
max           1.252000              0.291000        0.663800

       worst fractal dimension       label
count               569.000000  569.000000
mean                  0.083946    0.627417
std                   0.018061    0.483918
min                   0.055040    0.000000
25%                   0.071460    0.000000
50%                   0.080040    1.000000
75%                   0.092080    1.000000
max                   0.207500    1.000000

[8 rows x 31 columns]
```

[141]:
```
corr=data_frame.corr()
corr
```

[141]:

	mean radius	mean texture	mean perimeter	mean area	\
mean radius	1.000000	0.323782	0.997855	0.987357	
mean texture	0.323782	1.000000	0.329533	0.321086	
mean perimeter	0.997855	0.329533	1.000000	0.986507	
mean area	0.987357	0.321086	0.986507	1.000000	
mean smoothness	0.170581	-0.023389	0.207278	0.177028	
mean compactness	0.506124	0.236702	0.556936	0.498502	
mean concavity	0.676764	0.302418	0.716136	0.685983	
mean concave points	0.822529	0.293464	0.850977	0.823269	
mean symmetry	0.147741	0.071401	0.183027	0.151293	
mean fractal dimension	-0.311631	-0.076437	-0.261477	-0.283110	
radius error	0.679090	0.275869	0.691765	0.732562	
texture error	-0.097317	0.386358	-0.086761	-0.066280	
perimeter error	0.674172	0.281673	0.693135	0.726628	
area error	0.735864	0.259845	0.744983	0.800086	
smoothness error	-0.222600	0.006614	-0.202694	-0.166777	
compactness error	0.206000	0.191975	0.250744	0.212583	
concavity error	0.194204	0.143293	0.228082	0.207660	
concave points error	0.376169	0.163851	0.407217	0.372320	
symmetry error	-0.104321	0.009127	-0.081629	-0.072497	
fractal dimension error	-0.042641	0.054458	-0.005523	-0.019887	
worst radius	0.969539	0.352573	0.969476	0.962746	

worst texture	0.297008	0.912045	0.303038	0.287489
worst perimeter	0.965137	0.358040	0.970387	0.959120
worst area	0.941082	0.343546	0.941550	0.959213
worst smoothness	0.119616	0.077503	0.150549	0.123523
worst compactness	0.413463	0.277830	0.455774	0.390410
worst concavity	0.526911	0.301025	0.563879	0.512606
worst concave points	0.744214	0.295316	0.771241	0.722017
worst symmetry	0.163953	0.105008	0.189115	0.143570
worst fractal dimension	0.007066	0.119205	0.051019	0.003738
label	-0.730029	-0.415185	-0.742636	-0.708984

	mean smoothness	mean compactness	mean concavity	\
mean radius	0.170581	0.506124	0.676764	
mean texture	-0.023389	0.236702	0.302418	
mean perimeter	0.207278	0.556936	0.716136	
mean area	0.177028	0.498502	0.685983	
mean smoothness	1.000000	0.659123	0.521984	
mean compactness	0.659123	1.000000	0.883121	
mean concavity	0.521984	0.883121	1.000000	
mean concave points	0.553695	0.831135	0.921391	
mean symmetry	0.557775	0.602641	0.500667	
mean fractal dimension	0.584792	0.565369	0.336783	
radius error	0.301467	0.497473	0.631925	
texture error	0.068406	0.046205	0.076218	
perimeter error	0.296092	0.548905	0.660391	
area error	0.246552	0.455653	0.617427	
smoothness error	0.332375	0.135299	0.098564	
compactness error	0.318943	0.738722	0.670279	
concavity error	0.248396	0.570517	0.691270	
concave points error	0.380676	0.642262	0.683260	
symmetry error	0.200774	0.229977	0.178009	
fractal dimension error	0.283607	0.507318	0.449301	
worst radius	0.213120	0.535315	0.688236	
worst texture	0.036072	0.248133	0.299879	
worst perimeter	0.238853	0.590210	0.729565	
worst area	0.206718	0.509604	0.675987	
worst smoothness	0.805324	0.565541	0.448822	
worst compactness	0.472468	0.865809	0.754968	
worst concavity	0.434926	0.816275	0.884103	
worst concave points	0.503053	0.815573	0.861323	
worst symmetry	0.394309	0.510223	0.409464	
worst fractal dimension	0.499316	0.687382	0.514930	
label	-0.358560	-0.596534	-0.696360	

	mean concave points	mean symmetry	\
mean radius	0.822529	0.147741	
mean texture	0.293464	0.071401	

```
mean perimeter                  0.850977       0.183027
mean area                       0.823269       0.151293
mean smoothness                 0.553695       0.557775
mean compactness                0.831135       0.602641
mean concavity                  0.921391       0.500667
mean concave points             1.000000       0.462497
mean symmetry                   0.462497       1.000000
mean fractal dimension          0.166917       0.479921
radius error                    0.698050       0.303379
texture error                   0.021480       0.128053
perimeter error                 0.710650       0.313893
area error                      0.690299       0.223970
smoothness error                0.027653       0.187321
compactness error               0.490424       0.421659
concavity error                 0.439167       0.342627
concave points error            0.615634       0.393298
symmetry error                  0.095351       0.449137
fractal dimension error         0.257584       0.331786
worst radius                    0.830318       0.185728
worst texture                   0.292752       0.090651
worst perimeter                 0.855923       0.219169
worst area                      0.809630       0.177193
worst smoothness                0.452753       0.426675
worst compactness               0.667454       0.473200
worst concavity                 0.752399       0.433721
worst concave points            0.910155       0.430297
worst symmetry                  0.375744       0.699826
worst fractal dimension         0.368661       0.438413
label                          -0.776614      -0.330499

                        mean fractal dimension  …  worst texture  \
mean radius                          -0.311631  …       0.297008
mean texture                         -0.076437  …       0.912045
mean perimeter                       -0.261477  …       0.303038
mean area                            -0.283110  …       0.287489
mean smoothness                       0.584792  …       0.036072
mean compactness                      0.565369  …       0.248133
mean concavity                        0.336783  …       0.299879
mean concave points                   0.166917  …       0.292752
mean symmetry                         0.479921  …       0.090651
mean fractal dimension                1.000000  …      -0.051269
radius error                          0.000111  …       0.194799
texture error                         0.164174  …       0.409003
perimeter error                       0.039830  …       0.200371
area error                           -0.090170  …       0.196497
smoothness error                      0.401964  …      -0.074743
compactness error                     0.559837  …       0.143003
```

concavity error	0.446630	…	0.100241
concave points error	0.341198	…	0.086741
symmetry error	0.345007	…	-0.077473
fractal dimension error	0.688132	…	-0.003195
worst radius	-0.253691	…	0.359921
worst texture	-0.051269	…	1.000000
worst perimeter	-0.205151	…	0.365098
worst area	-0.231854	…	0.345842
worst smoothness	0.504942	…	0.225429
worst compactness	0.458798	…	0.360832
worst concavity	0.346234	…	0.368366
worst concave points	0.175325	…	0.359755
worst symmetry	0.334019	…	0.233027
worst fractal dimension	0.767297	…	0.219122
label	0.012838	…	-0.456903

	worst perimeter	worst area	worst smoothness	\
mean radius	0.965137	0.941082	0.119616	
mean texture	0.358040	0.343546	0.077503	
mean perimeter	0.970387	0.941550	0.150549	
mean area	0.959120	0.959213	0.123523	
mean smoothness	0.238853	0.206718	0.805324	
mean compactness	0.590210	0.509604	0.565541	
mean concavity	0.729565	0.675987	0.448822	
mean concave points	0.855923	0.809630	0.452753	
mean symmetry	0.219169	0.177193	0.426675	
mean fractal dimension	-0.205151	-0.231854	0.504942	
radius error	0.719684	0.751548	0.141919	
texture error	-0.102242	-0.083195	-0.073658	
perimeter error	0.721031	0.730713	0.130054	
area error	0.761213	0.811408	0.125389	
smoothness error	-0.217304	-0.182195	0.314457	
compactness error	0.260516	0.199371	0.227394	
concavity error	0.226680	0.188353	0.168481	
concave points error	0.394999	0.342271	0.215351	
symmetry error	-0.103753	-0.110343	-0.012662	
fractal dimension error	-0.001000	-0.022736	0.170568	
worst radius	0.993708	0.984015	0.216574	
worst texture	0.365098	0.345842	0.225429	
worst perimeter	1.000000	0.977578	0.236775	
worst area	0.977578	1.000000	0.209145	
worst smoothness	0.236775	0.209145	1.000000	
worst compactness	0.529408	0.438296	0.568187	
worst concavity	0.618344	0.543331	0.518523	
worst concave points	0.816322	0.747419	0.547691	
worst symmetry	0.269493	0.209146	0.493838	
worst fractal dimension	0.138957	0.079647	0.617624	

```
label                          -0.782914   -0.733825         -0.421465

                         worst compactness  worst concavity  \
mean radius                       0.413463         0.526911
mean texture                      0.277830         0.301025
mean perimeter                    0.455774         0.563879
mean area                         0.390410         0.512606
mean smoothness                   0.472468         0.434926
mean compactness                  0.865809         0.816275
mean concavity                    0.754968         0.884103
mean concave points               0.667454         0.752399
mean symmetry                     0.473200         0.433721
mean fractal dimension            0.458798         0.346234
radius error                      0.287103         0.380585
texture error                    -0.092439        -0.068956
perimeter error                   0.341919         0.418899
area error                        0.283257         0.385100
smoothness error                 -0.055558        -0.058298
compactness error                 0.678780         0.639147
concavity error                   0.484858         0.662564
concave points error              0.452888         0.549592
symmetry error                    0.060255         0.037119
fractal dimension error           0.390159         0.379975
worst radius                      0.475820         0.573975
worst texture                     0.360832         0.368366
worst perimeter                   0.529408         0.618344
worst area                        0.438296         0.543331
worst smoothness                  0.568187         0.518523
worst compactness                 1.000000         0.892261
worst concavity                   0.892261         1.000000
worst concave points              0.801080         0.855434
worst symmetry                    0.614441         0.532520
worst fractal dimension           0.810455         0.686511
label                            -0.590998        -0.659610

                         worst concave points  worst symmetry  \
mean radius                          0.744214        0.163953
mean texture                         0.295316        0.105008
mean perimeter                       0.771241        0.189115
mean area                            0.722017        0.143570
mean smoothness                      0.503053        0.394309
mean compactness                     0.815573        0.510223
mean concavity                       0.861323        0.409464
mean concave points                  0.910155        0.375744
mean symmetry                        0.430297        0.699826
mean fractal dimension               0.175325        0.334019
radius error                         0.531062        0.094543
```

texture error	-0.119638	-0.128215
perimeter error	0.554897	0.109930
area error	0.538166	0.074126
smoothness error	-0.102007	-0.107342
compactness error	0.483208	0.277878
concavity error	0.440472	0.197788
concave points error	0.602450	0.143116
symmetry error	-0.030413	0.389402
fractal dimension error	0.215204	0.111094
worst radius	0.787424	0.243529
worst texture	0.359755	0.233027
worst perimeter	0.816322	0.269493
worst area	0.747419	0.209146
worst smoothness	0.547691	0.493838
worst compactness	0.801080	0.614441
worst concavity	0.855434	0.532520
worst concave points	1.000000	0.502528
worst symmetry	0.502528	1.000000
worst fractal dimension	0.511114	0.537848
label	-0.793566	-0.416294

	worst fractal dimension	label
mean radius	0.007066	-0.730029
mean texture	0.119205	-0.415185
mean perimeter	0.051019	-0.742636
mean area	0.003738	-0.708984
mean smoothness	0.499316	-0.358560
mean compactness	0.687382	-0.596534
mean concavity	0.514930	-0.696360
mean concave points	0.368661	-0.776614
mean symmetry	0.438413	-0.330499
mean fractal dimension	0.767297	0.012838
radius error	0.049559	-0.567134
texture error	-0.045655	0.008303
perimeter error	0.085433	-0.556141
area error	0.017539	-0.548236
smoothness error	0.101480	0.067016
compactness error	0.590973	-0.292999
concavity error	0.439329	-0.253730
concave points error	0.310655	-0.408042
symmetry error	0.078079	0.006522
fractal dimension error	0.591328	-0.077972
worst radius	0.093492	-0.776454
worst texture	0.219122	-0.456903
worst perimeter	0.138957	-0.782914
worst area	0.079647	-0.733825
worst smoothness	0.617624	-0.421465

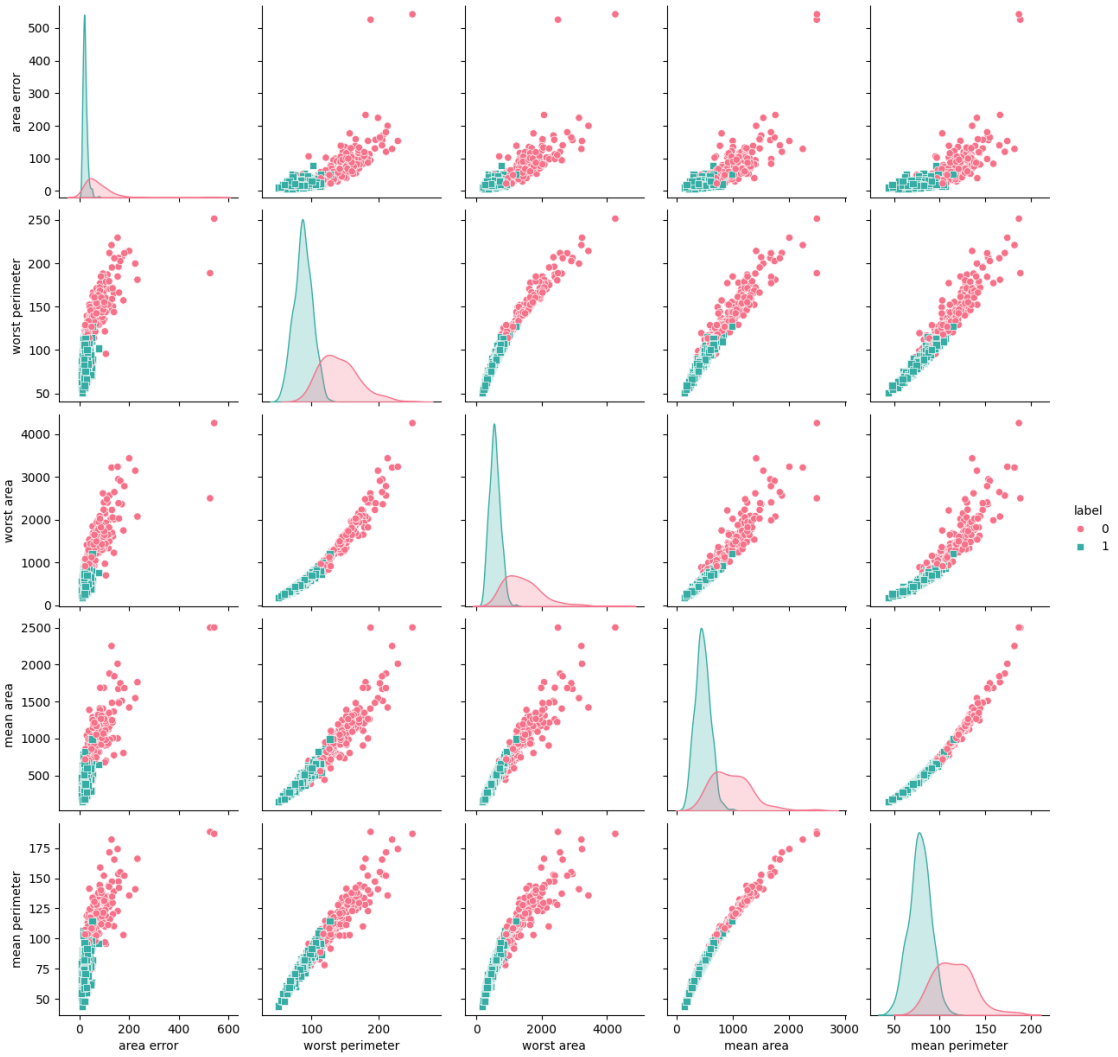
```
worst compactness                      0.810455 -0.590998
worst concavity                        0.686511 -0.659610
worst concave points                   0.511114 -0.793566
worst symmetry                         0.537848 -0.416294
worst fractal dimension                1.000000 -0.323872
label                                 -0.323872  1.000000

[31 rows x 31 columns]
```

[142]:
```
std=data_frame.std()
print(f"std: {std}")
```

```
std: mean radius                3.524049
mean texture                    4.301036
mean perimeter                 24.298981
mean area                     351.914129
mean smoothness                 0.014064
mean compactness                0.052813
mean concavity                  0.079720
mean concave points             0.038803
mean symmetry                   0.027414
mean fractal dimension          0.007060
radius error                    0.277313
texture error                   0.551648
perimeter error                 2.021855
area error                     45.491006
smoothness error                0.003003
compactness error               0.017908
concavity error                 0.030186
concave points error            0.006170
symmetry error                  0.008266
fractal dimension error         0.002646
worst radius                    4.833242
worst texture                   6.146258
worst perimeter                33.602542
worst area                    569.356993
worst smoothness                0.022832
worst compactness               0.157336
worst concavity                 0.208624
worst concave points            0.065732
worst symmetry                  0.061867
worst fractal dimension         0.018061
label                           0.483918
dtype: float64
```

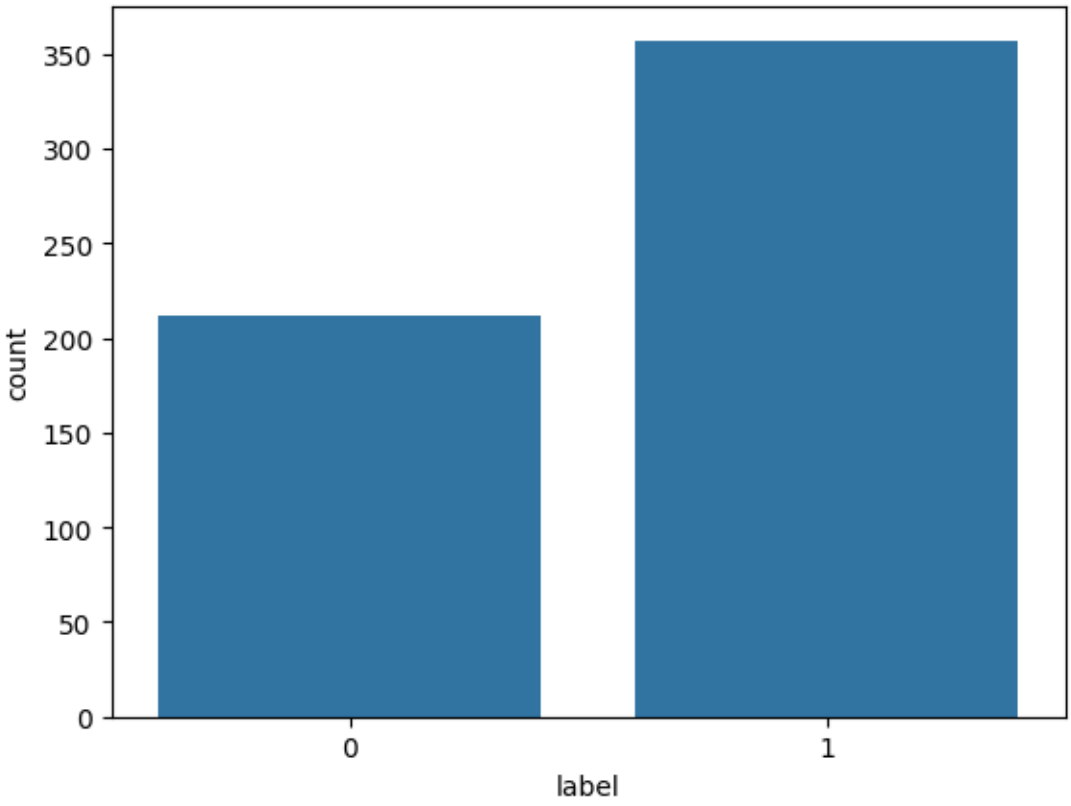
2 Exploratory Data Analysis

[143]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# visualize the data

# Features with higher standard deviations
selected_features = ['area error', 'worst perimeter', 'worst area', 'mean␣
 ↪area', 'mean perimeter']

# Adding 'label' as hue to differentiate classes
sns.pairplot(data_frame, vars=selected_features, hue='label', markers=["o",␣
 ↪"s"], palette="husl")
plt.show()
```



[144]:
```
import seaborn as sns
sns.countplot(x='label',data=data_frame)
```

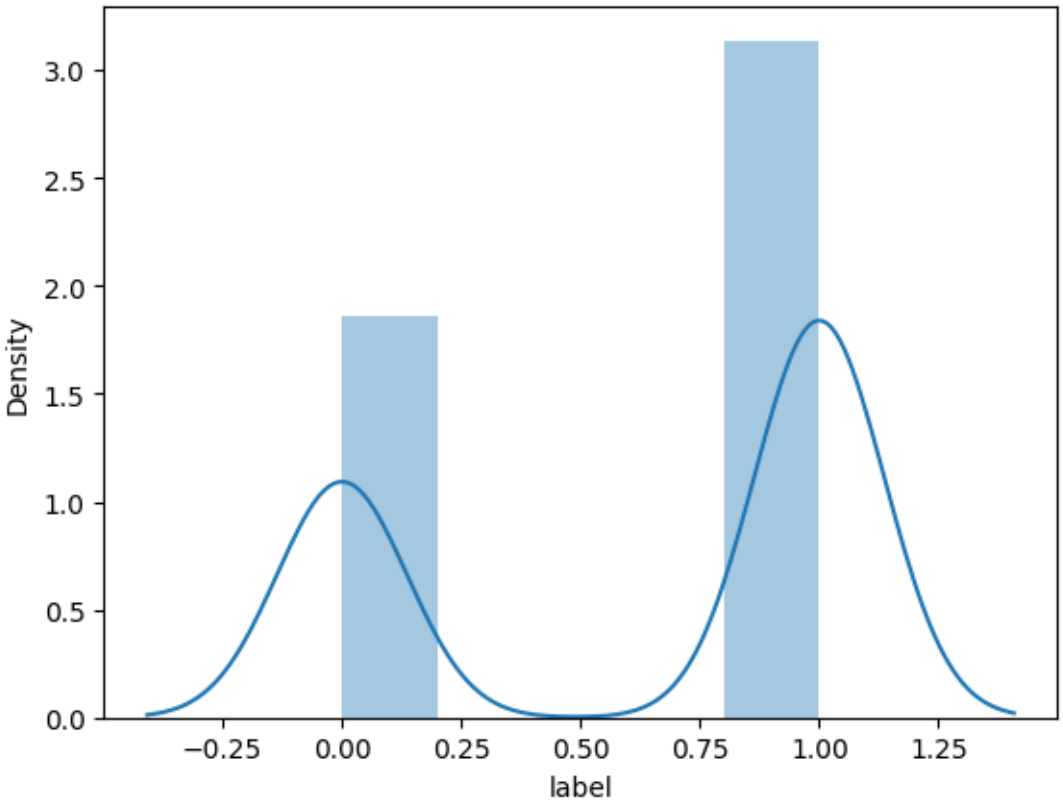
[144]: <Axes: xlabel='label', ylabel='count'>



[145]:
```
sns.distplot(data_frame['label'])
```

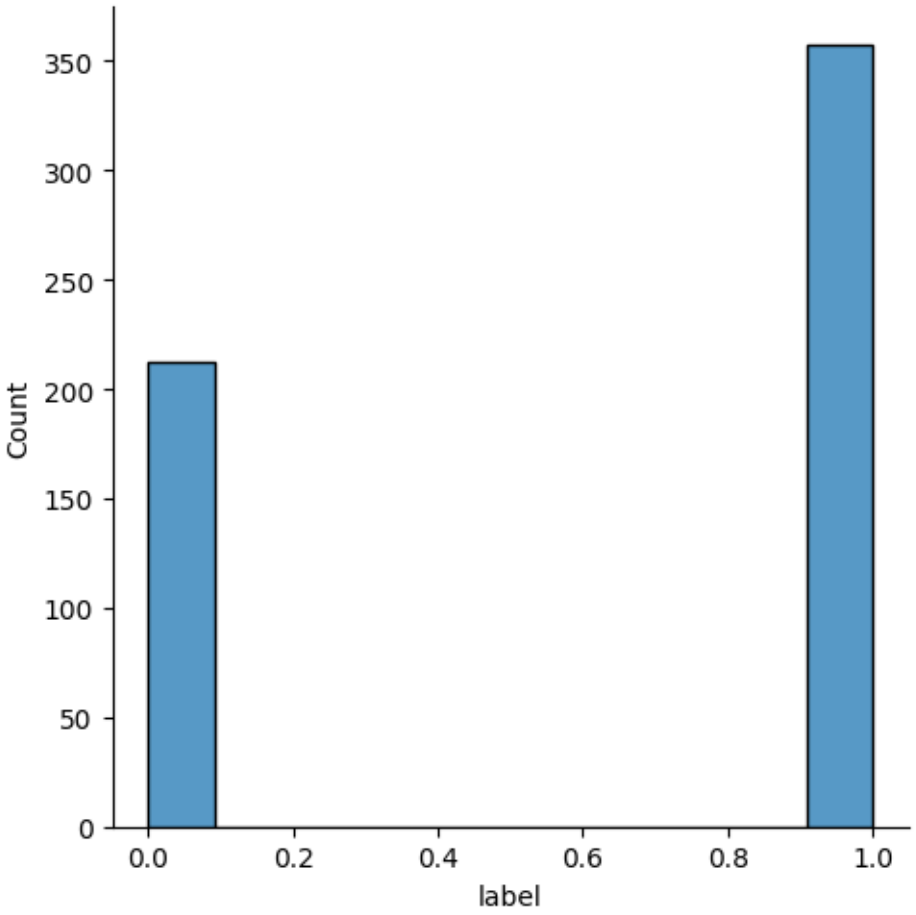
```
<ipython-input-145-fc0043849d4b>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data_frame['label'])
```

[145]: <Axes: xlabel='label', ylabel='Density'>



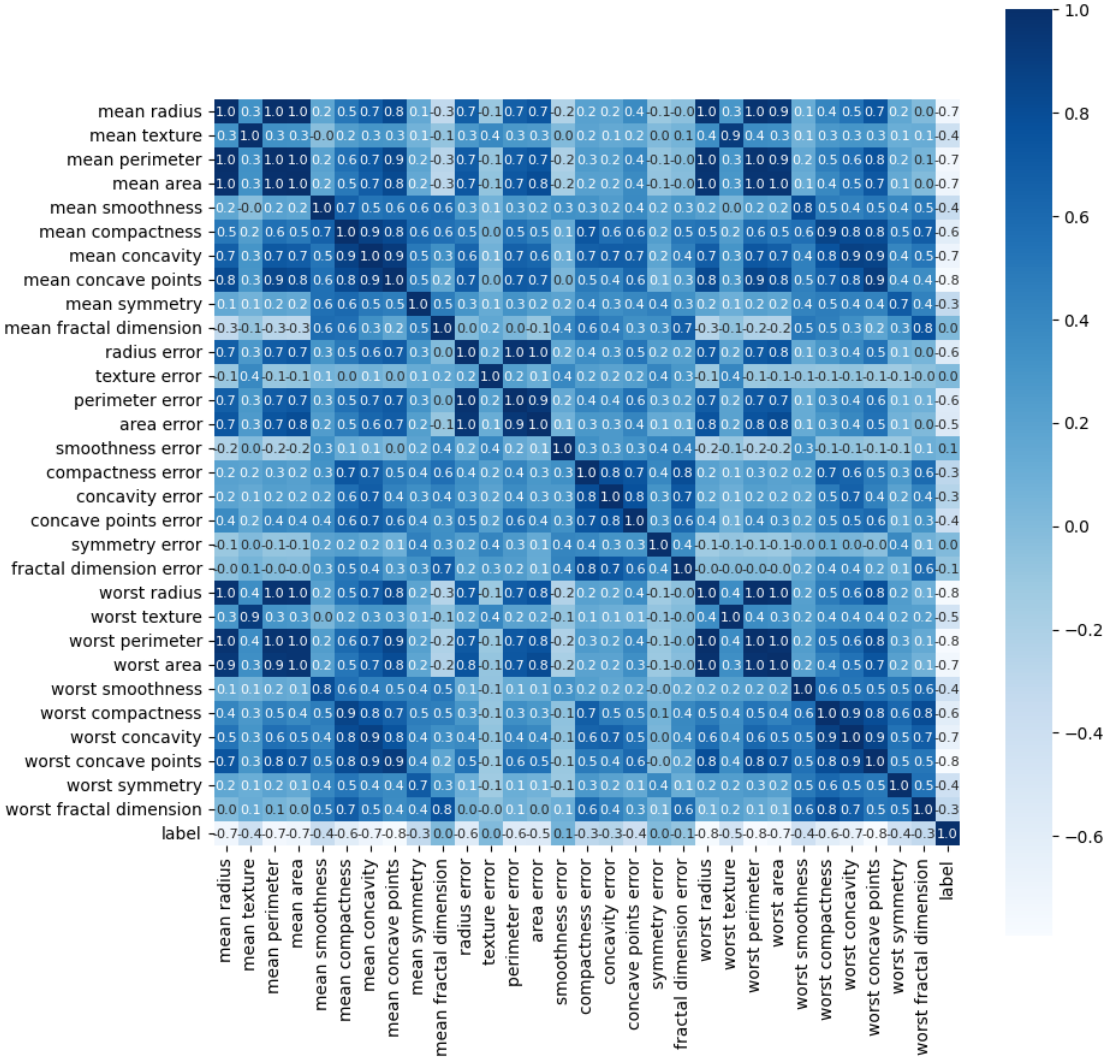
[146]:
```
sns.displot(data_frame['label'])
```

[146]: <seaborn.axisgrid.FacetGrid at 0x7f2cb1b7a740>



[147]:
```
correlation = data_frame.corr()

# constructing a Heat Map
plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True,␣
 ↪annot_kws={'size':8}, cmap='Blues')
```

[147]: <Axes: >



Separating the features and target

[148]:
```
X = data_frame.drop(columns='label', axis=1)
Y = data_frame['label']
```

[149]:
```
print(X)
```

```
     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 radius  worst texture  \
0                   0.07871  …        25.380          17.33
1                   0.05667  …        24.990          23.41
2                   0.05999  …        23.570          25.53
3                   0.09744  …        14.910          26.50
4                   0.05883  …        22.540          16.67
..                      …  …           …              …
564                 0.05623  …        25.450          26.40
565                 0.05533  …        23.690          38.25
566                 0.05648  …        18.980          34.12
567                 0.07016  …        25.740          39.42
568                 0.05884  …         9.456          30.37

     worst perimeter  worst area  worst smoothness  worst compactness  \
0             184.60      2019.0           0.16220            0.66560
1             158.80      1956.0           0.12380            0.18660
2             152.50      1709.0           0.14440            0.42450
3              98.87       567.7           0.20980            0.86630
4             152.20      1575.0           0.13740            0.20500
..               …          …                 …                …
564           166.10      2027.0           0.14100            0.21130
565           155.00      1731.0           0.11660            0.19220
566           126.70      1124.0           0.11390            0.30940
567           184.60      1821.0           0.16500            0.86810
568            59.16       268.6           0.08996            0.06444

     worst concavity  worst concave points  worst symmetry  \
0             0.7119                0.2654          0.4601
1             0.2416                0.1860          0.2750
2             0.4504                0.2430          0.3613
```

```
3              0.6869                0.2575          0.6638
4              0.4000                0.1625          0.2364
..                ...                   ...             ...
564            0.4107                0.2216          0.2060
565            0.3215                0.1628          0.2572
566            0.3403                0.1418          0.2218
567            0.9387                0.2650          0.4087
568            0.0000                0.0000          0.2871

     worst fractal dimension
0                    0.11890
1                    0.08902
2                    0.08758
3                    0.17300
4                    0.07678
..                       ...
564                  0.07115
565                  0.06637
566                  0.07820
567                  0.12400
568                  0.07039

[569 rows x 30 columns]
```

[150]:
```
print(Y)
```

```
0      0
1      0
2      0
3      0
4      0
      ..
564    0
565    0
566    0
567    0
568    1
Name: label, Length: 569, dtype: int64
```

Splitting the data into training data & Testing data

[151]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,␣
 ↪random_state=2)
```

To check if the number of rows in x_train matches the number of elements in y_train. You can use x_train.shape[0] to get the number of rows in x_train and compare it with len(y_train).

[152]:
```
print(X_train.shape)
print(Y_train.shape)
print(X_train.shape[0] == Y_train.shape[0])
print(X_train.shape == Y_train.shape)

print(X_test.shape)          # print(len(X_test))
print(Y_test.shape)           #print(len(Y_test))
print(X_test.shape[0] == Y_test.shape[0])
print(X_test.shape == Y_test.shape)
```

```
(455, 30)
(455,)
True
False
(114, 30)
(114,)
True
False
```

[153]:
```
print("Missing values in y_train:")
print(Y_train.isnull().sum())
print("\nMissing values in y_test:")
print(Y_test.isnull().sum())
print("\nUnique values in y_test:")
print(Y_train.unique())
```

```
Missing values in y_train:
0

Missing values in y_test:
0

Unique values in y_test:
[1 0]
```

[154]:
```
# Check indices of x_train and y_train
print(X_train.index)
print(Y_train.index)
```

```
Int64Index([560, 428, 198, 203,  41, 266, 309, 352,  89, 240,
            …
            433, 263, 360,  75, 466, 299, 534, 493, 527, 168],
           dtype='int64', length=455)
Int64Index([560, 428, 198, 203,  41, 266, 309, 352,  89, 240,
            …
            433, 263, 360,  75, 466, 299, 534, 493, 527, 168],
           dtype='int64', length=455)
```

3 Feature Scaling

[155]:
```
from sklearn.preprocessing import StandardScaler

# Scale the training data
scaler = StandardScaler().fit(X_train)
x_train = scaler.transform(X_train)
```

4 Model Training

Implementing Logistic Regression Model

[156]:
```
model = LogisticRegression()
```

[157]:
```
# training the Logistic Regression model using Training data

model.fit(X_train, Y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[157]:
```
LogisticRegression()
```

5 Model Evaluation

6 Accuracy Score

[158]:
```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
```

[159]:
```
print('Accuracy on training data = ', training_data_accuracy)
```

```
Accuracy on training data =  0.9472527472527472
```

[160]:
```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
```

[161]:
```
print('Accuracy on test data = ', test_data_accuracy)
```

```
Accuracy on test data =  0.9298245614035088
```

[162]:
```
from sklearn.metrics import mean_squared_error,r2_score

# The mean squared error
print("Mean squared error: %.2f" % mean_squared_error(Y_test,␣
 ↪X_test_prediction))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination: %.2f" % r2_score(Y_test,␣
 ↪X_test_prediction))
```

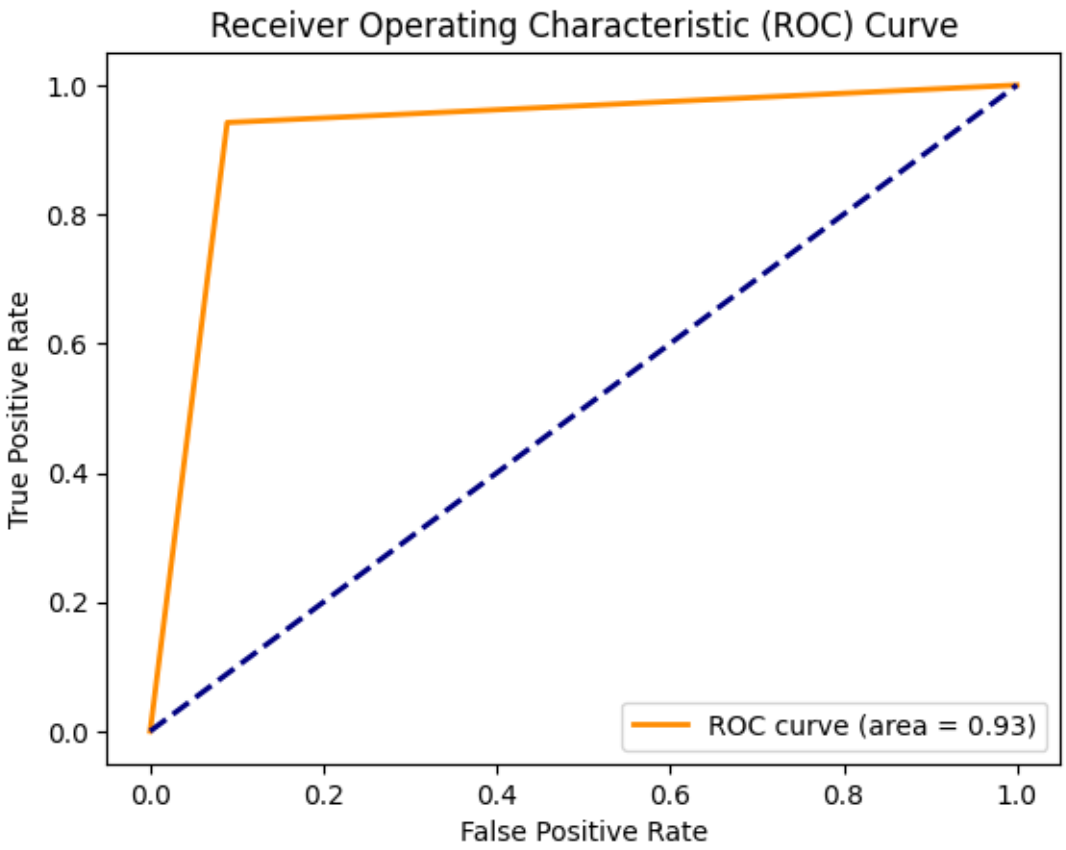
```
Mean squared error: 0.07
Coefficient of determination: 0.71
```

7 Reciever Operating Curve

[163]:
```
from sklearn.metrics import roc_curve, auc

fpr, tpr, thresholds = roc_curve(Y_test, X_test_prediction)
roc_auc = auc(fpr, tpr)
roc_auc
```

[163]:
```
0.9265700483091787
```

[164]:
```
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = {:.2f})'.
 ↪format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```



8 Building a Predictive System

[ ]:
```
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

# Replace the original feature names with the provided list
feature_names = [
    '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'
```

```
]

def preprocess_input(user_input):
    """
    Preprocess the user input before making predictions.
    """

    scaler = StandardScaler()
    scaled_input = scaler.fit_transform(user_input.reshape(1, -1))
    return scaled_input

def predict_cancer_type(model, user_input):
    """
    Make predictions using the trained machine learning model.
    """
    preprocessed_input = preprocess_input(user_input)
    prediction = model.predict(preprocessed_input)
    return prediction

def get_user_input():
    """
    Get input from the user for breast cancer features with validation.
    """
    user_input = np.zeros(len(feature_names) - 1)  # Assuming you have 30␣
 ↪features excluding the label

    # Get input from the user for each feature with validation
    for i, feature_name in enumerate(feature_names[:-1]):  # Exclude the label
        while True:
            try:
                user_input[i] = float(input(f"Enter value for {feature_name}:␣
 ↪"))
                break  # Break the loop if the input is valid
            except ValueError:
                print("Invalid input. Please enter a valid numerical value.")

    return user_input

def main():
    # Get input from the user
    user_input = get_user_input()

    # Make a prediction
    prediction = predict_cancer_type(model, user_input)

    # Display the prediction label
    if prediction == 1:
```

```
        print("The predicted cancer type is: Malignant")
    else:
        print("The predicted cancer type is: Benign")

if __name__ == "__main__":
    main()
```