Breast Cancer Detection Using Machine Learning Classifier

Import essential libraries

In [1]:
```
# import libraries
import pandas as pd # for data manupulation or analysis
import numpy as np # for numeric calculation
import matplotlib.pyplot as plt # for data visualization
import seaborn as sns # for data visualization
```

Data Load

In [2]:
```
#Load breast cancer dataset
from sklearn.datasets import load_breast_cancer
cancer_dataset = load_breast_cancer()
```

Data Manupulation

In [3]:
```
cancer_dataset
```

Out[3]:
```
{'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, 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, 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, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
        1, 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, 0, 1, 1, 1, 1,
        1, 1, 1, 0, 1, 0, 1, 1, 0, 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, 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, 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):                 0.0    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': 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\sklearn\\datasets\\dat
a\\breast_cancer.csv'}
```

In [4]:
```
type(cancer_dataset)
```

Out[4]:
```
sklearn.utils.Bunch
```

In [5]:
```
# keys in dataset
cancer_dataset.keys()
```

Out[5]:
```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
```

In [6]:
```
# featurs of each cells in numeric format
cancer_dataset['data']
```

Out[6]:
```
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]])
```

In [7]:
```
type(cancer_dataset['data'])
```

Out[7]:
```
numpy.ndarray
```

In [8]:
```
# malignant or benign value
cancer_dataset['target']
```

Out[8]:
```
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, 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, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 0, 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, 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, 0, 0, 0, 0, 0, 0, 1])
```

In [9]:
```
# target value name malignant or benign tumor
cancer_dataset['target_names']
```

Out[9]:
```
array(['malignant', 'benign'], dtype='<U9')
```

In [10]:
```
# description of data
print(cancer_dataset['DESCR'])
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
        - fractal dimension ("coastline approximation" - 1)

        The mean, standard error, and "worst" or largest (mean of the three
        largest values) of these features were computed for each image,
        resulting in 30 features.  For instance, field 3 is Mean Radius, field
        13 is Radius SE, field 23 is Worst Radius.

        - class:
                - WDBC-Malignant
                - WDBC-Benign

    :Summary Statistics:

    ===================================== ====== ======
                                           Min    Max
    ===================================== ====== ======
    radius (mean):                        6.981  28.11
    texture (mean):                       9.71   39.28
    perimeter (mean):                     43.79  188.5
    area (mean):                          143.5  2501.0
    smoothness (mean):                    0.053  0.163
    compactness (mean):                   0.019  0.345
    concavity (mean):                     0.0    0.427
    concave points (mean):                0.0    0.201
    symmetry (mean):                      0.106  0.304
    fractal dimension (mean):             0.05   0.097
    radius (standard error):              0.112  2.873
    texture (standard error):             0.36   4.885
    perimeter (standard error):           0.757  21.98
    area (standard error):                6.802  542.2
    smoothness (standard error):          0.002  0.031
    compactness (standard error):         0.002  0.135
    concavity (standard error):           0.0    0.396
    concave points (standard error):      0.0    0.053
    symmetry (standard error):            0.008  0.079
    fractal dimension (standard error):   0.001  0.03
```

```
    radius (worst):                       7.93   36.04
    texture (worst):                      12.02  49.54
    perimeter (worst):                    50.41  251.2
    area (worst):                         185.2  4254.0
    smoothness (worst):                   0.071  0.223
    compactness (worst):                  0.027  1.058
    concavity (worst):                    0.0    1.252
    concave points (worst):               0.0    0.291
    symmetry (worst):                     0.156  0.664
    fractal dimension (worst):            0.055  0.208
    ===================================== ====== ======

    :Missing Attribute Values: None

    :Class Distribution: 212 - Malignant, 357 - Benign

    :Creator:  Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

    :Donor: Nick Street

    :Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.
https://goo.gl/U2Uwz2

Features are computed from a digitized image of a fine needle
aspirate (FNA) of a breast mass.  They describe
characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree.  Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

.. topic:: References

   - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction
     for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on
     Electronic Imaging: Science and Technology, volume 1905, pages 861-870,
     San Jose, CA, 1993.
   - O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and
     prognosis via linear programming. Operations Research, 43(4), pages 570-577,
     July-August 1995.
   - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques
     to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994)
     163-171.
```

In [11]:
```
# name of features
print(cancer_dataset['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']
```

In [12]:
```
# location/path of data file
print(cancer_dataset['filename'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\datasets\data\breast_cancer.csv
```

Create DataFrame

In [13]:
```
# create datafrmae
cancer_df = pd.DataFrame(np.c_[cancer_dataset['data'],cancer_dataset['target']],
             columns = np.append(cancer_dataset['feature_names'], ['target']))
```

In [14]:
```
# DataFrame to CSV file
cancer_df.to_csv('breast_cancer_dataframe.csv')
```

In [15]:
```
# Head of cancer DataFrame
cancer_df.head(6)
```

Out[15]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	din
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	12.45	15.70	82.57	477.1	0.12780	0.17000	0.1578	0.08089	0.2087	

6 rows × 31 columns

In [16]:
```
# Tail of cancer DataFrame
cancer_df.tail(6)
```

Out[16]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	c
563	20.92	25.09	143.00	1347.0	0.10990	0.22360	0.31740	0.14740	0.2149	
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	

6 rows × 31 columns

In [17]:
```
# Information of cancer Dataframe
cancer_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
mean radius                569 non-null float64
mean texture               569 non-null float64
mean perimeter             569 non-null float64
mean area                  569 non-null float64
mean smoothness            569 non-null float64
mean compactness           569 non-null float64
mean concavity             569 non-null float64
mean concave points        569 non-null float64
mean symmetry              569 non-null float64
mean fractal dimension     569 non-null float64
radius error               569 non-null float64
texture error              569 non-null float64
perimeter error            569 non-null float64
area error                 569 non-null float64
smoothness error           569 non-null float64
compactness error          569 non-null float64
concavity error            569 non-null float64
concave points error       569 non-null float64
symmetry error             569 non-null float64
fractal dimension error    569 non-null float64
worst radius               569 non-null float64
worst texture              569 non-null float64
worst perimeter            569 non-null float64
worst area                 569 non-null float64
worst smoothness           569 non-null float64
worst compactness          569 non-null float64
worst concavity            569 non-null float64
worst concave points       569 non-null float64
worst symmetry             569 non-null float64
worst fractal dimension    569 non-null float64
target                     569 non-null float64
dtypes: float64(31)
memory usage: 137.9 KB
```

In [18]:
```
# Numerical distribution of data
cancer_df.describe()
```

Out[18]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	co
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.0
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.0
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.0
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.0
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.0
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.0
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.0
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.2

8 rows × 31 columns

In [19]:
```
cancer_df.isnull().sum()
```

Out[19]:
```
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
target                     0
dtype: int64
```

Data Visualization

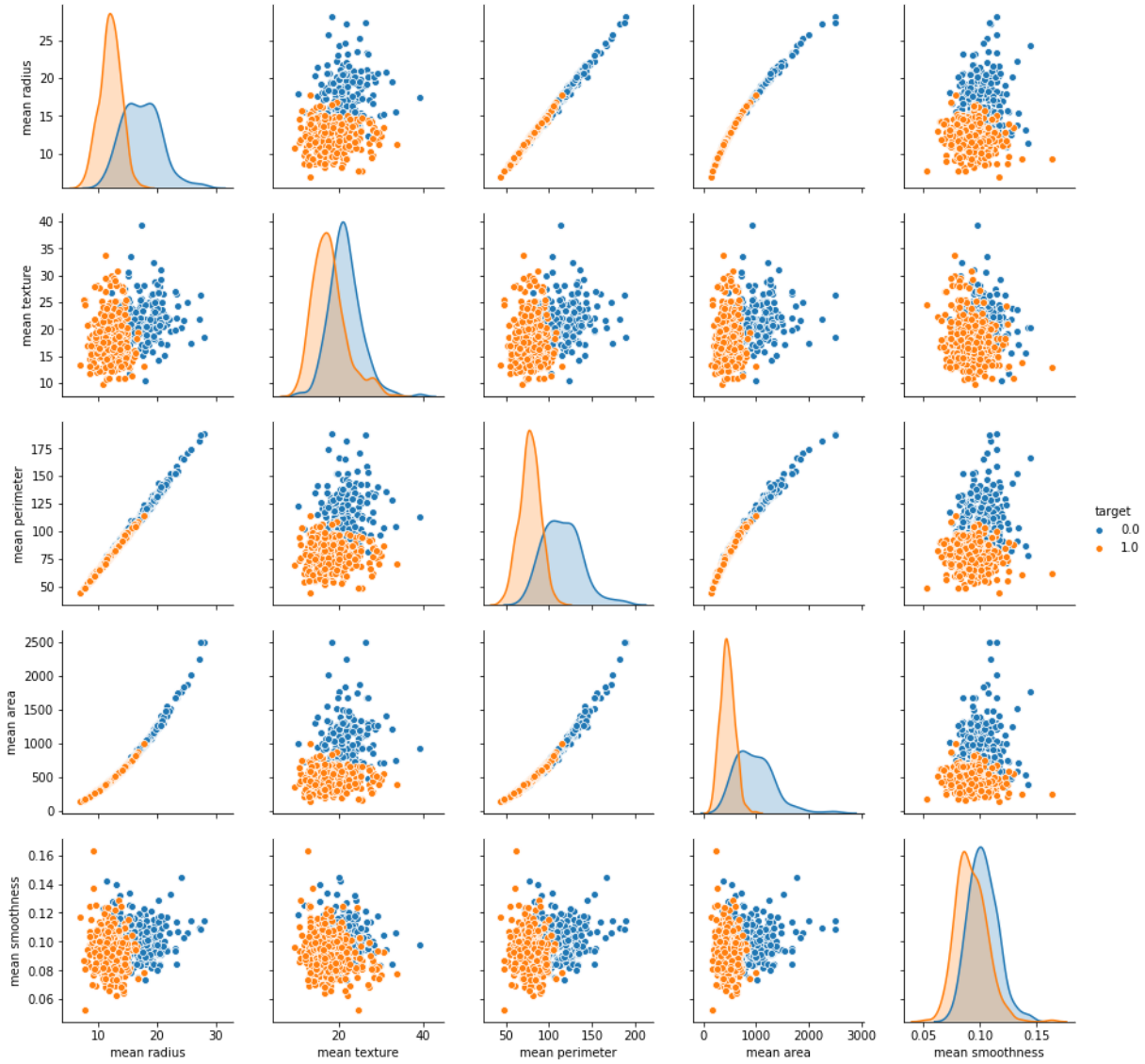
In [20]:
```
# Paiplot of cancer dataframe
sns.pairplot(cancer_df, hue = 'target')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py:488: Run
timeWarning: invalid value encountered in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdetools.py:34:
RuntimeWarning: invalid value encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
```

Out[20]:
```
<seaborn.axisgrid.PairGrid at 0x2723ae3c2e8>
```

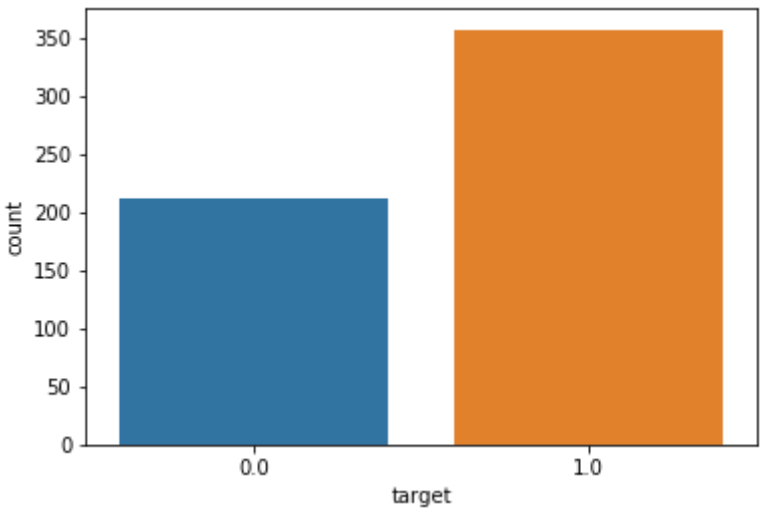
In [21]:
```
# pair plot of sample feature
sns.pairplot(cancer_df, hue = 'target',
             vars = ['mean radius', 'mean texture', 'mean perimeter', 'mean area', '
```

Out[21]:
```
<seaborn.axisgrid.PairGrid at 0x2725f914cc0>
```



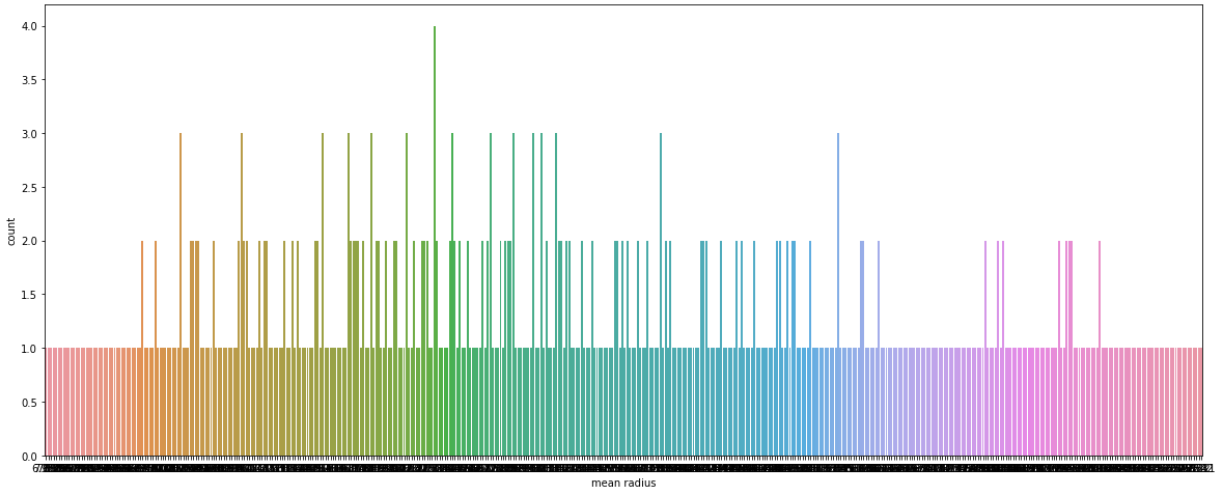
In [22]:
```
# Count the target class
sns.countplot(cancer_df['target']) #  ************************** img 5 **********
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x27272ef3898>



In [23]:
```
# counter plot of feature mean radius
plt.figure(figsize = (20,8))
sns.countplot(cancer_df['mean radius']) # *** img 7 ****
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x27272ebdb00>



Heatmap

In [1]:
```
# heatmap of DataFrame
plt.figure(figsize=(16,9))
sns.heatmap(cancer_df) # **** img 8 ****
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-1bcbe5553910> in <module>
      1 # heatmap of DataFrame
----> 2 plt.figure(figsize=(16,9))
      3 sns.heatmap(cancer_df) # **** img 8 ****

NameError: name 'plt' is not defined
```

Heatmap of a correlation matrix

In [25]:
```
cancer_df.corr()
```

Out[25]:

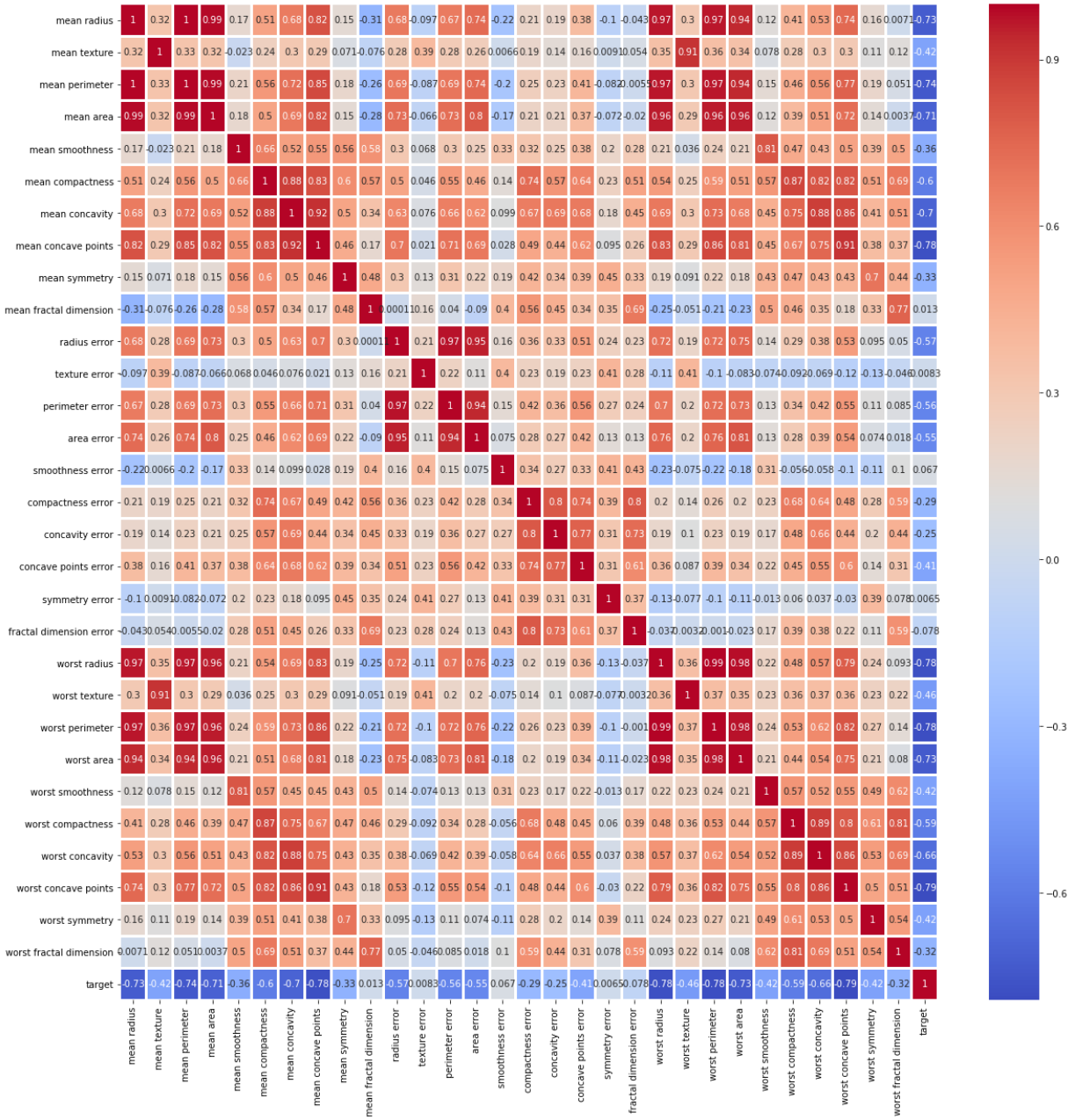
	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	m con po
mean radius	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822
mean texture	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.29
mean perimeter	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.85
mean area	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.82
mean smoothness	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.55
mean compactness	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.83
mean concavity	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.92
mean concave points	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.00
mean	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.46

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	m con po
symmetry								
mean fractal dimension	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.16
radius error	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925	0.69
texture error	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218	0.02
perimeter error	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391	0.71
area error	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427	0.69
smoothness error	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564	0.02
compactness error	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	0.670279	0.49
concavity error	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	0.691270	0.43
concave points error	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	0.683260	0.61
symmetry error	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	0.178009	0.09
fractal dimension error	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	0.449301	0.25
worst radius	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	0.688236	0.83
worst texture	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	0.299879	0.29
worst perimeter	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	0.729565	0.85
worst area	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	0.675987	0.80
worst smoothness	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	0.448822	0.45
worst compactness	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	0.754968	0.66
worst concavity	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	0.884103	0.75
worst concave points	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	0.861323	0.91
worst symmetry	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	0.409464	0.37
worst fractal dimension	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	0.514930	0.36
target	-0.730029	-0.415185	-0.742636	-0.708984	-0.358560	-0.596534	-0.696360	-0.77

31 rows × 31 columns

In [26]:
```
# Heatmap of Correlation matrix of breast cancer DataFrame
plt.figure(figsize=(20,20))
sns.heatmap(cancer_df.corr(), annot = True, cmap ='coolwarm', linewidths=2) # *** im
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x27274bb2748>



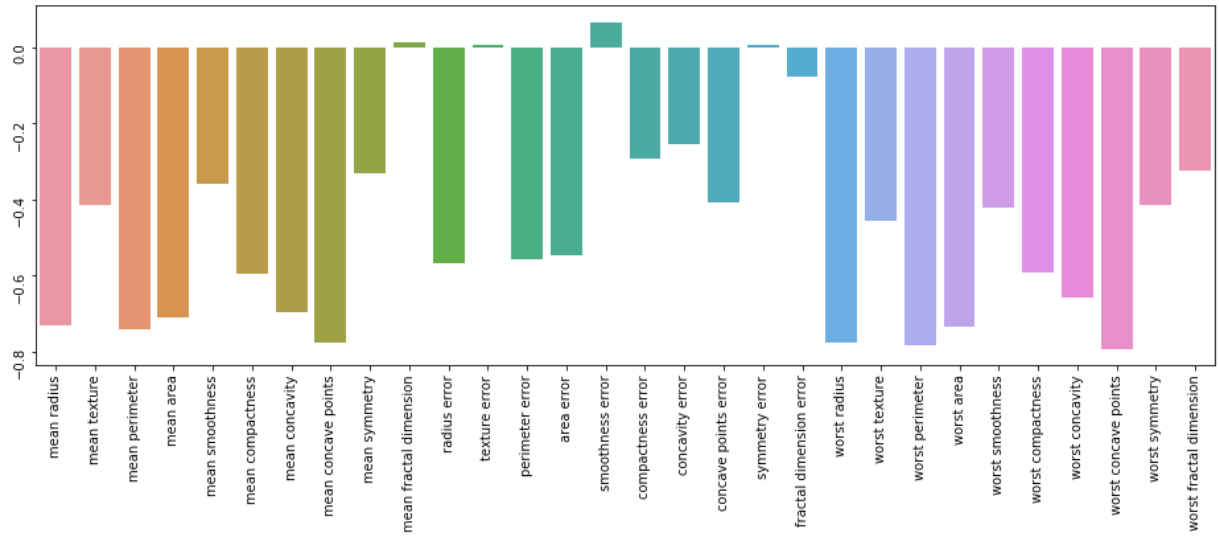
Correlation Barplot

In [27]:
```
# create second DataFrame by droping target
cancer_df2 = cancer_df.drop(['target'], axis = 1)
print("The shape of 'cancer_df2' is : ", cancer_df2.shape)
```

```
The shape of 'cancer_df2' is :  (569, 30)
```

In [28]:
```
#cancer_df2.corrwith(cancer_df.target)
```

In [29]:
```
# visualize correlation barplot
plt.figure(figsize = (16,5))
ax = sns.barplot(cancer_df2.corrwith(cancer_df.target).index, cancer_df2.corrwith(ca
ax.tick_params(labelrotation = 90) # **** img 10 ***
```



In [30]:
```
cancer_df2.corrwith(cancer_df.target).index
```

Out[30]:
```
Index(['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='object')
```

Split DatFrame in Train and Test

In [31]:
```
# input variable
X = cancer_df.drop(['target'], axis = 1)
X.head(6)
```

Out[31]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	din
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	12.45	15.70	82.57	477.1	0.12780	0.17000	0.1578	0.08089	0.2087	

6 rows × 30 columns

In [32]:
```
# output variable
y = cancer_df['target']
y.head(6)
```

Out[32]:
```
0    0.0
1    0.0
```

```
2    0.0
3    0.0
4    0.0
5    0.0
Name: target, dtype: float64
```

In [33]:
```
# split dataset into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_st
```

In [34]:
```
X_train
```

Out[34]:

	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

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
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
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 [35]:
```
X_test
```

Out[35]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean 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 [36]:
```
y_train
```

Out[36]:
```
306    1.0
410    1.0
197    0.0
```

```
376    1.0
244    0.0
299    1.0
312    1.0
331    1.0
317    0.0
341    1.0
156    0.0
71     1.0
218    0.0
344    1.0
247    1.0
212    0.0
559    1.0
176    1.0
422    1.0
248    1.0
232    1.0
444    0.0
383    1.0
279    1.0
494    1.0
316    1.0
523    1.0
90     1.0
469    1.0
373    0.0
      ...
539    1.0
110    1.0
5      0.0
144    1.0
103    1.0
210    0.0
446    0.0
41     0.0
362    1.0
377    1.0
254    0.0
146    0.0
86     0.0
542    1.0
431    1.0
65     0.0
205    0.0
44     0.0
27     0.0
80     1.0
437    1.0
113    1.0
204    1.0
519    1.0
411    1.0
8      0.0
73     0.0
400    0.0
118    0.0
206    1.0
Name: target, Length: 455, dtype: float64
```

In [37]:
```
y_test
```

Out[37]:
```
28     0.0
163    1.0
123    1.0
361    1.0
549    1.0
339    0.0
286    1.0
```

```
354    1.0
421    1.0
124    1.0
543    1.0
537    1.0
567    0.0
555    1.0
511    1.0
333    1.0
68     1.0
189    1.0
557    1.0
436    1.0
479    0.0
52     1.0
401    1.0
355    1.0
318    1.0
359    1.0
40     0.0
323    0.0
495    1.0
45     0.0
      ...
7      0.0
155    1.0
56     0.0
151    1.0
203    0.0
34     0.0
417    0.0
42     0.0
453    1.0
500    1.0
258    0.0
369    0.0
313    1.0
426    1.0
140    1.0
388    1.0
116    1.0
198    0.0
490    1.0
50     1.0
199    0.0
366    0.0
455    1.0
162    0.0
403    1.0
414    0.0
515    1.0
186    0.0
3      0.0
261    0.0
Name: target, Length: 114, dtype: float64
```

Feature scaling

In [38]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_sc = sc.fit_transform(X_train)
X_test_sc = sc.transform(X_test)
```

Machine Learning Model Building

In [39]:
```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

Logistic Regression

In [42]:
```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
lr_classifier = LogisticRegression(random_state = 51, penalty = 'l1')
lr_classifier.fit(X_train, y_train)
y_pred_lr = lr_classifier.predict(X_test)
accuracy_score(y_test, y_pred_lr)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: Fut
ureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to s
ilence this warning.
  FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:931: ConvergenceWarni
ng: Liblinear failed to converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
```

Out[42]:
```
0.9736842105263158
```