In [4]:

```
sklearn.datasets import load_iris6
iris= load_iris()
dir(iris)
iris
```

```
Out[4]: {'data': array([[5.1, 3.5, 1.4, 0.2],
                        [4.9, 3. , 1.4, 0.2],
                        [4.7, 3.2, 1.3, 0.2],
                        [4.6, 3.1, 1.5, 0.2],
                        [5. , 3.6, 1.4, 0.2],
                        [5.4, 3.9, 1.7, 0.4],
                        [4.6, 3.4, 1.4, 0.3],
                        [5. , 3.4, 1.5, 0.2],
                        [4.4, 2.9, 1.4, 0.2],
                        [4.9, 3.1, 1.5, 0.1],
                        [5.4, 3.7, 1.5, 0.2],
                        [4.8, 3.4, 1.6, 0.2],
                        [4.8, 3. , 1.4, 0.1],
                        [4.3, 3. , 1.1, 0.1],
                        [5.8, 4. , 1.2, 0.2],
                        [5.7, 4.4, 1.5, 0.4],
                        [5.4, 3.9, 1.3, 0.4],
                        [5.1, 3.5, 1.4, 0.3],
                        [5.7, 3.8, 1.7, 0.3],
                        [5.1, 3.8, 1.5, 0.3],
                        [5.4, 3.4, 1.7, 0.2],
                        [5.1, 3.7, 1.5, 0.4],
                        [4.6, 3.6, 1. , 0.2],
                        [5.1, 3.3, 1.7, 0.5],
                        [4.8, 3.4, 1.9, 0.2],
                        [5. , 3. , 1.6, 0.2],
                        [5. , 3.4, 1.6, 0.4],
                        [5.2, 3.5, 1.5, 0.2],
                        [5.2, 3.4, 1.4, 0.2],
                        [4.7, 3.2, 1.6, 0.2],
                        [4.8, 3.1, 1.6, 0.2],
                        [5.4, 3.4, 1.5, 0.4],
                        [5.2, 4.1, 1.5, 0.1],
                        [5.5, 4.2, 1.4, 0.2],
                        [4.9, 3.1, 1.5, 0.2],
                        [5. , 3.2, 1.2, 0.2],
                        [5.5, 3.5, 1.3, 0.2],
                        [4.9, 3.6, 1.4, 0.1],
                        [4.4, 3. , 1.3, 0.2],
                        [5.1, 3.4, 1.5, 0.2],
                        [5. , 3.5, 1.3, 0.3],
                        [4.5, 2.3, 1.3, 0.3],
                        [4.4, 3.2, 1.3, 0.2],
                        [5. , 3.5, 1.6, 0.6],
                        [5.1, 3.8, 1.9, 0.4],
                        [4.8, 3. , 1.4, 0.3],
                        [5.1, 3.8, 1.6, 0.2],
                        [4.6, 3.2, 1.4, 0.2],
                        [5.3, 3.7, 1.5, 0.2],
                        [5. , 3.3, 1.4, 0.2],
                        [7. , 3.2, 4.7, 1.4],
                        [6.4, 3.2, 4.5, 1.5],
                        [6.9, 3.1, 4.9, 1.5],
                        [5.5, 2.3, 4. , 1.3],
                        [6.5, 2.8, 4.6, 1.5],
                        [5.7, 2.8, 4.5, 1.3],
                        [6.3, 3.3, 4.7, 1.6],
                        [4.9, 2.4, 3.3, 1. ],
                        [6.6, 2.9, 4.6, 1.3],
                        [5.2, 2.7, 3.9, 1.4],
                        [5. , 2. , 3.5, 1. ],
                        [5.9, 3. , 4.2, 1.5],
                        [6. , 2.2, 4. , 1. ],
                        [6.1, 2.9, 4.7, 1.4],
                        [5.6, 2.9, 3.6, 1.3],
                        [6.7, 3.1, 4.4, 1.4],
                        [5.6, 3. , 4.5, 1.5],
                        [5.8, 2.7, 4.1, 1. ],
                        [6.2, 2.2, 4.5, 1.5],
                        [5.6, 2.5, 3.9, 1.1],
                        [5.9, 3.2, 4.8, 1.8],
                        [6.1, 2.8, 4. , 1.3],
                        [6.3, 2.5, 4.9, 1.5],
                        [6.1, 2.8, 4.7, 1.2],
                        [6.4, 2.9, 4.3, 1.3],
                        [6.6, 3. , 4.4, 1.4],
                        [6.8, 2.8, 4.8, 1.4],
                        [6.7, 3. , 5. , 1.7],
                        [6. , 2.9, 4.5, 1.5],
                        [5.7, 2.6, 3.5, 1. ],
                        [5.5, 2.4, 3.8, 1.1],
                        [5.5, 2.4, 3.7, 1. ],
                        [5.8, 2.7, 3.9, 1.2],
                        [6. , 2.7, 5.1, 1.6],
                        [5.4, 3. , 4.5, 1.5],
                        [6. , 3.4, 4.5, 1.6],
                        [6.7, 3.1, 4.7, 1.5],
                        [6.3, 2.3, 4.4, 1.3],
                        [5.6, 3. , 4.1, 1.3],
                        [5.5, 2.5, 4. , 1.3],
                        [5.5, 2.6, 4.4, 1.2],
                        [6.1, 3. , 4.6, 1.4],
                        [5.8, 2.6, 4. , 1.2],
                        [5. , 2.3, 3.3, 1. ],
                        [5.6, 2.7, 4.2, 1.3],
                        [5.7, 3. , 4.2, 1.2],
                        [5.7, 2.9, 4.2, 1.3],
                        [6.2, 2.9, 4.3, 1.3],
                        [5.1, 2.5, 3. , 1.1],
                        [5.7, 2.8, 4.1, 1.3],
                        [6.3, 3.3, 6. , 2.5],
                        [5.8, 2.7, 5.1, 1.9],
                        [7.1, 3. , 5.9, 2.1],
                        [6.3, 2.9, 5.6, 1.8],
                        [6.5, 3. , 5.8, 2.2],
                        [7.6, 3. , 6.6, 2.1],
                        [4.9, 2.5, 4.5, 1.7],
                        [7.3, 2.9, 6.3, 1.8],
                        [6.7, 2.5, 5.8, 1.8],
                        [7.2, 3.6, 6.1, 2.5],
                        [6.5, 3.2, 5.1, 2. ],
                        [6.4, 2.7, 5.3, 1.9],
                        [6.8, 3. , 5.5, 2.1],
                        [5.7, 2.5, 5. , 2. ],
                        [5.8, 2.8, 5.1, 2.4],
                        [6.4, 3.2, 5.3, 2.3],
                        [6.5, 3. , 5.5, 1.8],
                        [7.7, 3.8, 6.7, 2.2],
                        [7.7, 2.6, 6.9, 2.3],
                        [6. , 2.2, 5. , 1.5],
                        [6.9, 3.2, 5.7, 2.3],
                        [5.6, 2.8, 4.9, 2. ],
                        [7.7, 2.8, 6.7, 2. ],
                        [6.3, 2.7, 4.9, 1.8],
                        [6.7, 3.3, 5.7, 2.1],
                        [7.2, 3.2, 6. , 1.8],
                        [6.2, 2.8, 4.8, 1.8],
                        [6.1, 3. , 4.9, 1.8],
                        [6.4, 2.8, 5.6, 2.1],
                        [7.2, 3. , 5.8, 1.6],
                        [7.4, 2.8, 6.1, 1.9],
                        [7.9, 3.8, 6.4, 2. ],
                        [6.4, 2.8, 5.6, 2.2],
                        [6.3, 2.8, 5.1, 1.5],
                        [6.1, 2.6, 5.6, 1.4],
                        [7.7, 3. , 6.1, 2.3],
                        [6.3, 3.4, 5.6, 2.4],
                        [6.4, 3.1, 5.5, 1.8],
                        [6. , 3. , 4.8, 1.8],
                        [6.9, 3.1, 5.4, 2.1],
                        [6.7, 3.1, 5.6, 2.4],
                        [6.9, 3.1, 5.1, 2.3],
                        [5.8, 2.7, 5.1, 1.9],
                        [6.8, 3.2, 5.9, 2.3],
                        [6.7, 3.3, 5.7, 2.5],
                        [6.7, 3. , 5.2, 2.3],
                        [6.3, 2.5, 5. , 1.9],
                        [6.5, 3. , 5.2, 2. ],
                        [6.2, 3.4, 5.4, 2.3],
                        [5.9, 3. , 5.1, 1.8]]),
                 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0,
                        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                        0, 0, 0, 0, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
                 'frame': None,
                 'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
                 'DESCR': '.. _iris_dataset:\n\nIris plants dataset\n--------------------\n\n**
                Data Set Characteristics:**\n\n    :Number of Instances: 150 (50 in each of thr
                ee classes)\n    :Number of Attributes: 4 numeric, predictive attributes and th
                e class\n    :Attribute Information:\n        - sepal length in cm\n        - s
                epal width in cm\n        - petal length in cm\n        - petal width in cm\n
                - class:\n                - Iris-Setosa\n                - Iris-Versicolour\n
                - Iris-Virginica\n                \n    :Summary Statistics:\n\n    ===========
                === ==== ==== ======= ===== ====================\n                    Min  Max
                Mean    SD   Class Correlation\n    ============== ==== ==== ======= ===== ====
                ================\n    sepal length:   4.3  7.9   5.84   0.83    0.7826\n    sep
                al width:    2.0  4.4   3.05   0.43   -0.4194\n    petal length:   1.0  6.9
                3.76   1.76    0.9490  (high!)\n    petal width:    0.1  2.5   1.20   0.76
                0.9565  (high!)\n    ============== ==== ==== ======= ===== ==================
                =\n\n    :Missing Attribute Values: None\n    :Class Distribution: 33.3% for ea
                ch of 3 classes.\n    :Creator: R.A. Fisher\n    :Donor: Michael Marshall (MARS
                HALL%PLU@io.arc.nasa.gov)\n    :Date: July, 1988\n\nThe famous Iris database, f
                irst used by Sir R.A. Fisher. The dataset is taken\nfrom Fisher\'s paper. Note
                that it\'s the same as in R, but not as in the UCI\nMachine Learning Repositor
                y, which has two wrong data points.\n\nThis is perhaps the best known database
                to be found in the\npattern recognition literature.  Fisher\'s paper is a class
```

```
ic in the field and\nis referenced frequently to this day.  (See Duda & Hart, f
or example.)  The\ndata set contains 3 classes of 50 instances each, where each
class refers to a\ntype of iris plant.  One class is linearly separable from th
e other 2; the\nlatter are NOT linearly separable from each other.\n\n.. topi
c:: References\n\n   - Fisher, R.A. "The use of multiple measurements in taxono
mic problems"\n     Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contr
ibutions to\n     Mathematical Statistics" (John Wiley, NY, 1950).\n   - Duda,
R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis.\n     (Q32
7.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.\n   - Dasarathy,
B.V. (1980) "Nosing Around the Neighborhood: A New System\n     Structure and C
lassification Rule for Recognition in Partially Exposed\n     Environments".  I
EEE Transactions on Pattern Analysis and Machine\n     Intelligence, Vol. PAMI-
2, No. 1, 67-71.\n   - Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".
IEEE Transactions\n     on Information Theory, May 1972, 431-433.\n   - See als
o: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOCLASS II\n     conceptua
l clustering system finds 3 classes in the data.\n   - Many, many more ...',
 'feature_names': ['sepal length (cm)',
  'sepal width (cm)',
  'petal length (cm)',
  'petal width (cm)'],
 'filename': 'iris.csv',
 'data_module': 'sklearn.datasets.data'}
```

In [3]:

```
iris.feature_names
```

Out[3]:

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [5]:

```
iris.data
```

```
Out[5]:  array([[5.1, 3.5, 1.4, 0.2],
               [4.9, 3. , 1.4, 0.2],
               [4.7, 3.2, 1.3, 0.2],
               [4.6, 3.1, 1.5, 0.2],
               [5. , 3.6, 1.4, 0.2],
               [5.4, 3.9, 1.7, 0.4],
               [4.6, 3.4, 1.4, 0.3],
               [5. , 3.4, 1.5, 0.2],
               [4.4, 2.9, 1.4, 0.2],
               [4.9, 3.1, 1.5, 0.1],
               [5.4, 3.7, 1.5, 0.2],
               [4.8, 3.4, 1.6, 0.2],
               [4.8, 3. , 1.4, 0.1],
               [4.3, 3. , 1.1, 0.1],
               [5.8, 4. , 1.2, 0.2],
               [5.7, 4.4, 1.5, 0.4],
               [5.4, 3.9, 1.3, 0.4],
               [5.1, 3.5, 1.4, 0.3],
               [5.7, 3.8, 1.7, 0.3],
               [5.1, 3.8, 1.5, 0.3],
               [5.4, 3.4, 1.7, 0.2],
               [5.1, 3.7, 1.5, 0.4],
               [4.6, 3.6, 1. , 0.2],
               [5.1, 3.3, 1.7, 0.5],
               [4.8, 3.4, 1.9, 0.2],
               [5. , 3. , 1.6, 0.2],
               [5. , 3.4, 1.6, 0.4],
               [5.2, 3.5, 1.5, 0.2],
               [5.2, 3.4, 1.4, 0.2],
               [4.7, 3.2, 1.6, 0.2],
               [4.8, 3.1, 1.6, 0.2],
               [5.4, 3.4, 1.5, 0.4],
               [5.2, 4.1, 1.5, 0.1],
               [5.5, 4.2, 1.4, 0.2],
               [4.9, 3.1, 1.5, 0.2],
               [5. , 3.2, 1.2, 0.2],
               [5.5, 3.5, 1.3, 0.2],
               [4.9, 3.6, 1.4, 0.1],
               [4.4, 3. , 1.3, 0.2],
               [5.1, 3.4, 1.5, 0.2],
               [5. , 3.5, 1.3, 0.3],
               [4.5, 2.3, 1.3, 0.3],
               [4.4, 3.2, 1.3, 0.2],
               [5. , 3.5, 1.6, 0.6],
               [5.1, 3.8, 1.9, 0.4],
               [4.8, 3. , 1.4, 0.3],
               [5.1, 3.8, 1.6, 0.2],
               [4.6, 3.2, 1.4, 0.2],
               [5.3, 3.7, 1.5, 0.2],
               [5. , 3.3, 1.4, 0.2],
               [7. , 3.2, 4.7, 1.4],
               [6.4, 3.2, 4.5, 1.5],
               [6.9, 3.1, 4.9, 1.5],
               [5.5, 2.3, 4. , 1.3],
               [6.5, 2.8, 4.6, 1.5],
               [5.7, 2.8, 4.5, 1.3],
               [6.3, 3.3, 4.7, 1.6],
               [4.9, 2.4, 3.3, 1. ],
               [6.6, 2.9, 4.6, 1.3],
               [5.2, 2.7, 3.9, 1.4],
               [5. , 2. , 3.5, 1. ],
               [5.9, 3. , 4.2, 1.5],
               [6. , 2.2, 4. , 1. ],
               [6.1, 2.9, 4.7, 1.4],
               [5.6, 2.9, 3.6, 1.3],
               [6.7, 3.1, 4.4, 1.4],
               [5.6, 3. , 4.5, 1.5],
               [5.8, 2.7, 4.1, 1. ],
               [6.2, 2.2, 4.5, 1.5],
               [5.6, 2.5, 3.9, 1.1],
               [5.9, 3.2, 4.8, 1.8],
               [6.1, 2.8, 4. , 1.3],
               [6.3, 2.5, 4.9, 1.5],
               [6.1, 2.8, 4.7, 1.2],
               [6.4, 2.9, 4.3, 1.3],
               [6.6, 3. , 4.4, 1.4],
               [6.8, 2.8, 4.8, 1.4],
               [6.7, 3. , 5. , 1.7],
               [6. , 2.9, 4.5, 1.5],
               [5.7, 2.6, 3.5, 1. ],
               [5.5, 2.4, 3.8, 1.1],
               [5.5, 2.4, 3.7, 1. ],
               [5.8, 2.7, 3.9, 1.2],
               [6. , 2.7, 5.1, 1.6],
               [5.4, 3. , 4.5, 1.5],
               [6. , 3.4, 4.5, 1.6],
               [6.7, 3.1, 4.7, 1.5],
               [6.3, 2.3, 4.4, 1.3],
               [5.6, 3. , 4.1, 1.3],
               [5.5, 2.5, 4. , 1.3],
               [5.5, 2.6, 4.4, 1.2],
               [6.1, 3. , 4.6, 1.4],
               [5.8, 2.6, 4. , 1.2],
               [5. , 2.3, 3.3, 1. ],
               [5.6, 2.7, 4.2, 1.3],
               [5.7, 3. , 4.2, 1.2],
               [5.7, 2.9, 4.2, 1.3],
               [6.2, 2.9, 4.3, 1.3],
               [5.1, 2.5, 3. , 1.1],
               [5.7, 2.8, 4.1, 1.3],
               [6.3, 3.3, 6. , 2.5],
               [5.8, 2.7, 5.1, 1.9],
               [7.1, 3. , 5.9, 2.1],
               [6.3, 2.9, 5.6, 1.8],
               [6.5, 3. , 5.8, 2.2],
               [7.6, 3. , 6.6, 2.1],
               [4.9, 2.5, 4.5, 1.7],
               [7.3, 2.9, 6.3, 1.8],
               [6.7, 2.5, 5.8, 1.8],
               [7.2, 3.6, 6.1, 2.5],
               [6.5, 3.2, 5.1, 2. ],
               [6.4, 2.7, 5.3, 1.9],
               [6.8, 3. , 5.5, 2.1],
               [5.7, 2.5, 5. , 2. ],
               [5.8, 2.8, 5.1, 2.4],
               [6.4, 3.2, 5.3, 2.3],
               [6.5, 3. , 5.5, 1.8],
               [7.7, 3.8, 6.7, 2.2],
               [7.7, 2.6, 6.9, 2.3],
               [6. , 2.2, 5. , 1.5],
               [6.9, 3.2, 5.7, 2.3],
               [5.6, 2.8, 4.9, 2. ],
               [7.7, 2.8, 6.7, 2. ],
               [6.3, 2.7, 4.9, 1.8],
               [6.7, 3.3, 5.7, 2.1],
               [7.2, 3.2, 6. , 1.8],
               [6.2, 2.8, 4.8, 1.8],
               [6.1, 3. , 4.9, 1.8],
               [6.4, 2.8, 5.6, 2.1],
               [7.2, 3. , 5.8, 1.6],
               [7.4, 2.8, 6.1, 1.9],
               [7.9, 3.8, 6.4, 2. ],
               [6.4, 2.8, 5.6, 2.2],
               [6.3, 2.8, 5.1, 1.5],
               [6.1, 2.6, 5.6, 1.4],
               [7.7, 3. , 6.1, 2.3],
               [6.3, 3.4, 5.6, 2.4],
               [6.4, 3.1, 5.5, 1.8],
               [6. , 3. , 4.8, 1.8],
               [6.9, 3.1, 5.4, 2.1],
               [6.7, 3.1, 5.6, 2.4],
               [6.9, 3.1, 5.1, 2.3],
               [5.8, 2.7, 5.1, 1.9],
               [6.8, 3.2, 5.9, 2.3],
               [6.7, 3.3, 5.7, 2.5],
               [6.7, 3. , 5.2, 2.3],
               [6.3, 2.5, 5. , 1.9],
               [6.5, 3. , 5.2, 2. ],
               [6.2, 3.4, 5.4, 2.3],
               [5.9, 3. , 5.1, 1.8]])
```

In [12]:

```
import pandas as pd
df =pd.DataFrame(iris.data ,columns= iris.feature_names)
df.head()
```

Out[12]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [13]:

```
iris.target_names
```

Out[13]:

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [16]:

```
df['target'] =iris. target
df.head
```

Out[16]:

```
<bound method NDFrame.head of      sepal length (cm)  sepal width (cm)  petal l
ength (cm)  petal width (cm)  \
0                  5.1               3.5                1.4               0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2
4                  5.0               3.6                1.4               0.2
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

     target
0         0
1         0
2         0
3         0
4         0
..      ...
145       2
146       2
147       2
148       2
149       2

[150 rows x 5 columns]>
```

In [25]:

```
df[df.target==2].head()
```

Out[25]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
100	6.3	3.3	6.0	2.5	2
101	5.8	2.7	5.1	1.9	2
102	7.1	3.0	5.9	2.1	2
103	6.3	2.9	5.6	1.8	2
104	6.5	3.0	5.8	2.2	2

In [48]:

```
df0 =df[df.target==0]
df1 =df[df.target==1]
df2 =df[df.target==2]

df0.head(),df1.head(),df2.head()
```

```
Out[48]: (   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
          0                5.1               3.5                1.4               0.2
          1                4.9               3.0                1.4               0.2
          2                4.7               3.2                1.3               0.2
          3                4.6               3.1                1.5               0.2
          4                5.0               3.6                1.4               0.2

             target flowers_names Flowers_names
          0       0        setosa        setosa
          1       0        setosa        setosa
          2       0        setosa        setosa
          3       0        setosa        setosa
          4       0        setosa        setosa  ,
              sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
         \
          50                7.0               3.2                4.7               1.4
          51                6.4               3.2                4.5               1.5
          52                6.9               3.1                4.9               1.5
          53                5.5               2.3                4.0               1.3
          54                6.5               2.8                4.6               1.5

              target flowers_names Flowers_names
          50       1    versicolor    versicolor
          51       1    versicolor    versicolor
          52       1    versicolor    versicolor
          53       1    versicolor    versicolor
          54       1    versicolor    versicolor  ,
               sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
         \
          100                6.3               3.3                6.0               2.5
          101                5.8               2.7                5.1               1.9
          102                7.1               3.0                5.9               2.1
          103                6.3               2.9                5.6               1.8
          104                6.5               3.0                5.8               2.2

               target flowers_names Flowers_names
          100       2     virginica     virginica
          101       2     virginica     virginica
          102       2     virginica     virginica
          103       2     virginica     virginica
          104       2     virginica     virginica  )
```

```
In [45]: df['Flowers_names'] = df.target.apply( lambda x: iris.target_names[x])
         df
```

Out[45]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flowers_names	Flowers_names
0	5.1	3.5	1.4	0.2	0	setosa	setosa
1	4.9	3.0	1.4	0.2	0	setosa	setosa
2	4.7	3.2	1.3	0.2	0	setosa	setosa
3	4.6	3.1	1.5	0.2	0	setosa	setosa
4	5.0	3.6	1.4	0.2	0	setosa	setosa
...	...	...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2	virginica	virginica
146	6.3	2.5	5.0	1.9	2	virginica	virginica
147	6.5	3.0	5.2	2.0	2	virginica	virginica
148	6.2	3.4	5.4	2.3	2	virginica	virginica
149	5.9	3.0	5.1	1.8	2	virginica	virginica

150 rows × 7 columns

In [57]:

```
X=df.drop(['flowers_names','target','Flowers_names']  ,axis = 'columns')
X
```

Out[57]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

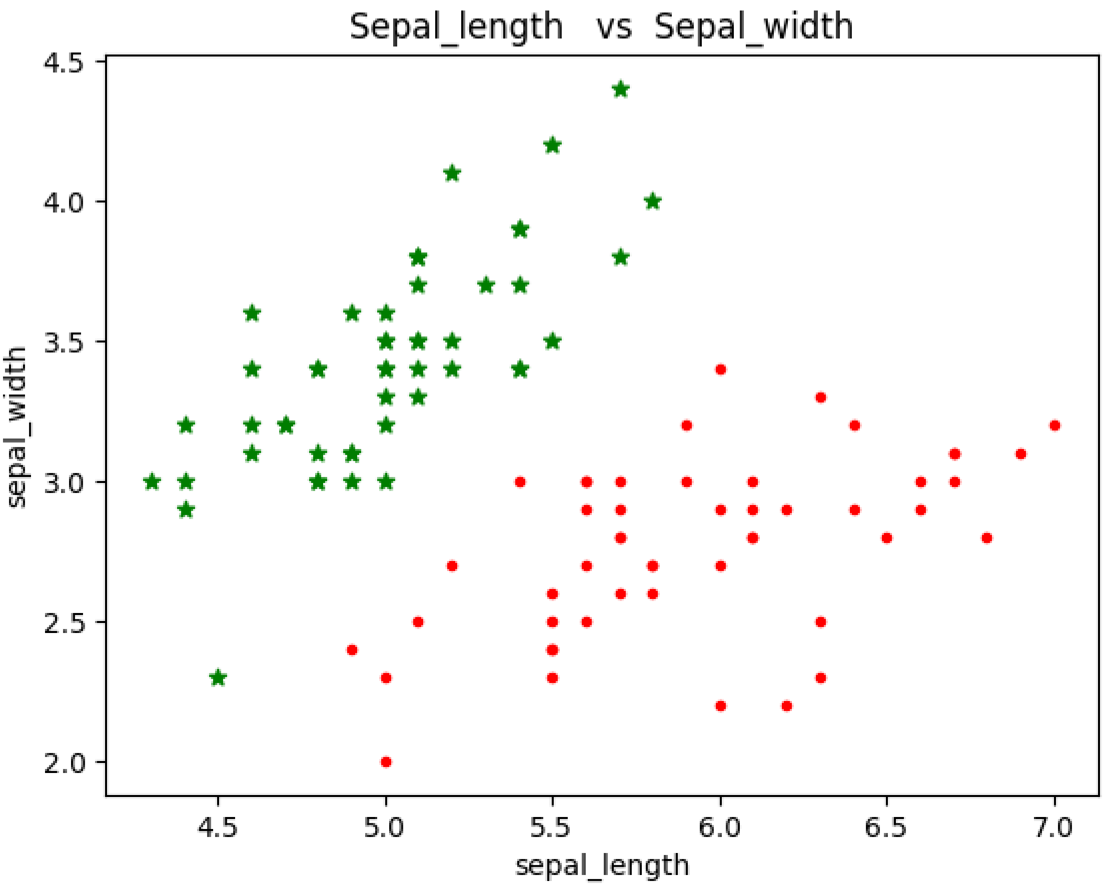
150 rows × 4 columns

In [50]:

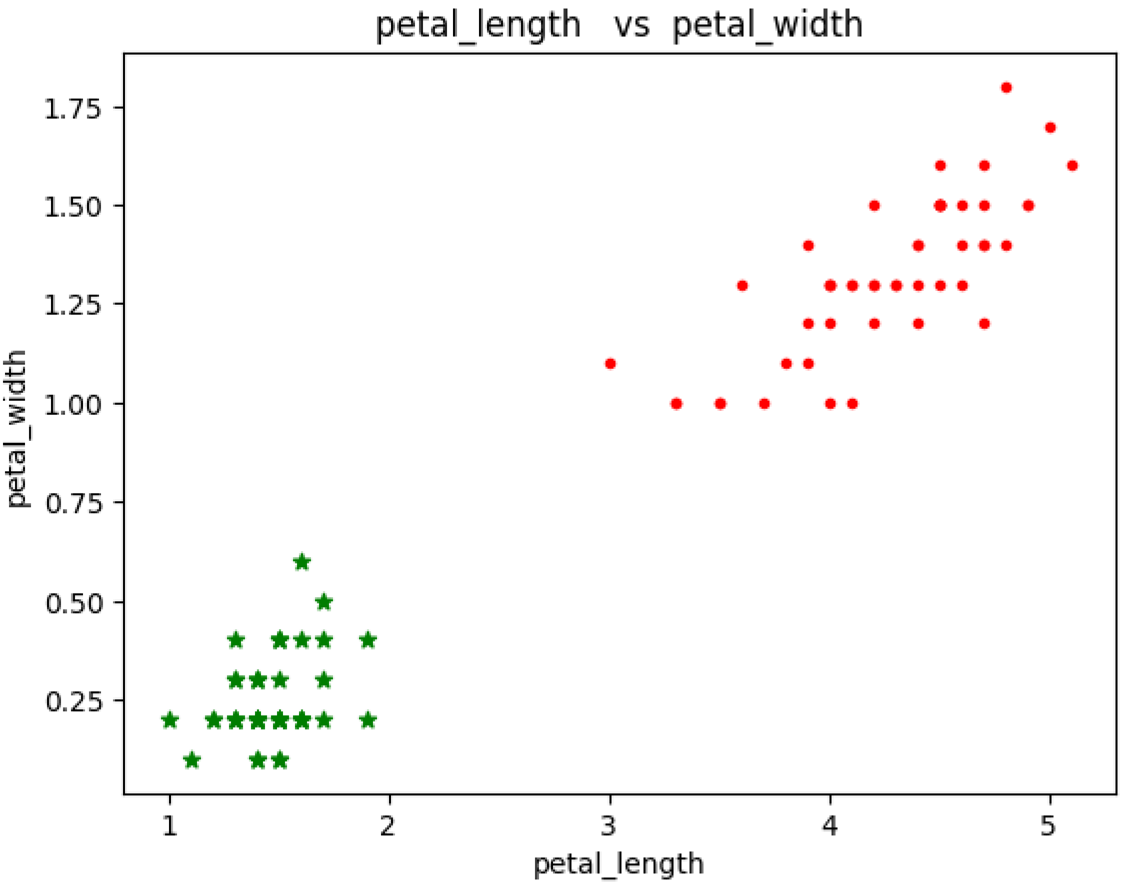
```
y= iris.target
y
```

```
Out[50]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
In [73]: from matplotlib import pyplot as pp
         pp.scatter(df0['sepal length (cm)'],df0['sepal width (cm)'], color= "green",mark
         pp.scatter(df1['sepal length (cm)'],df1['sepal width (cm)'], color= "red"  , mar
         pp.title('Sepal_length   vs  Sepal_width')
         pp.xlabel("sepal_length")
         pp.ylabel("sepal_width")
         pp.show()
```



```
In [76]: pp.scatter(df0['petal length (cm)'],df0['petal width (cm)'], color= "green",mark
         pp.scatter(df1['petal length (cm)'],df1['petal width (cm)'], color= "red",marker
         pp.title('petal_length   vs  petal_width')
         pp.xlabel("petal_length")
         pp.ylabel("petal_width")
         pp.show()
```



In [122…
```
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)
```

In [123…
```
len(X_train)
```

Out[123…
```
120
```

In [124…
```
len(X_test)
```

Out[124…
```
30
```

In [125…
```
from sklearn.svm import SVC
classifier = SVC()
```

In [126…
```
classifier.fit(X_train,y_train)
```

Out[126…
```
SVC()
```

In [127…
```
y_pred =classifier . predict(X_test)
(y_pred)
```

Out[127…
```
array([1, 2, 0, 1, 2, 0, 1, 0, 0, 2, 0, 0, 2, 2, 0, 0, 2, 0, 0, 2, 1, 1,
       1, 2, 0, 1, 2, 1, 0, 1])
```

In [130…
```
from sklearn.metrics import accuracy_score,confusion_matrix
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print('Confusion Matrix:')
print(cm)
```

```
Accuracy: 1.0
Confusion Matrix:
[[12  0  0]
 [ 0  9  0]
 [ 0  0  9]]
```

In [131…

```
#  evaluation model
classifier.score(X_test,y_test)
```

Out[131…

```
1.0
```