Untitled

December 25, 2019

[1]:
```python
from sklearn.datasets import  load_iris
iris = load_iris()

print(iris)
```

```
{'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]), '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 three
classes)\n    :Number of Attributes: 4 numeric, predictive attributes and the
class\n    :Attribute Information:\n        - sepal length in cm\n        -
sepal 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    sepal 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 each of 3 classes.\n    :Creator: R.A. Fisher\n
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n    :Date: July,
1988\n\nThe famous Iris database, first 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 Repository, 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 classic in the field and\nis referenced
frequently to this day.  (See Duda & Hart, for 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 the other 2; the\nlatter are NOT
linearly separable from each other.\n\n.. topic:: References\n\n   - Fisher,
R.A. "The use of multiple measurements in taxonomic problems"\n     Annual
Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to\n
Mathematical Statistics" (John Wiley, NY, 1950).\n   - Duda, R.O., & Hart, P.E.
(1973) Pattern Classification and Scene Analysis.\n     (Q327.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 Classification Rule
for Recognition in Partially Exposed\n     Environments".  IEEE 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 also: 1988 MLC Proceedings,
54-64.  Cheeseman et al"s AUTOCLASS II\n     conceptual 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': '/home/cesar/miniconda3/lib/python3.7/site-
packages/sklearn/datasets/data/iris.csv'}
```

[2]:
```
"""
en si el listado de iris es un diccionario

keys y elemento

el elemento en este caso es un arreglo de listas de listas
o sea un numpy array
"""
```

[2]:
```
'\nen si el listado de iris es un diccionario\n\nkeys y elemento\n\nel elemento
en este caso es un arreglo de listas de listas \no sea un numpy array \n'
```

[3]:
```
iris["target_names"]
```

[3]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

[4]:
```
iris["feature_names"]
```

[4]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

[5]:
```
type(iris)
```

[5]: sklearn.utils.Bunch

[6]:
```
type(iris["data"])
```

[6]: numpy.ndarray

[7]:
```
iris["data"]
```

[7]:
```
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]])
```

[8]:
```
iris["data"].shape
```

[8]: (150, 4)

[9]:
```
#un total de 150 flores o muestras
#con 4 features
```

[10]:
```
type(iris["target"])
```

[10]: numpy.ndarray

[69]:
```
iris["target"]

type (iris["target"])
```

[69]: numpy.ndarray

[12]:
```
"""
target means flower species
0 setosa
1 versicolor
2 virginica

"""
```

[12]: '\ntarget means flower species\n0 setosa\n1 versicolor\n2 virginica\n\n'

1 Training and testing data

The part of the data is used to build our machine learning model, and is called the training data or trainning set.

The rest of the data will be used to access how well the model works and is called test data, test set or hold-out set.

Scikit-learn contains a functaion that shuffles the dataset and splits it for you, the function:

train_test_split

[13]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(iris["data"],iris["target"]
                                                    , random_state = 0)
```

[35]:
```
X_train.shape
#see the number of instances and attributes  of Train data
X_train
```

[35]:
```
array([[5.9, 3. , 4.2, 1.5],
       [5.8, 2.6, 4. , 1.2],
       [6.8, 3. , 5.5, 2.1],
       [4.7, 3.2, 1.3, 0.2],
       [6.9, 3.1, 5.1, 2.3],
       [5. , 3.5, 1.6, 0.6],
       [5.4, 3.7, 1.5, 0.2],
       [5. , 2. , 3.5, 1. ],
       [6.5, 3. , 5.5, 1.8],
       [6.7, 3.3, 5.7, 2.5],
       [6. , 2.2, 5. , 1.5],
       [6.7, 2.5, 5.8, 1.8],
       [5.6, 2.5, 3.9, 1.1],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.3, 4.7, 1.6],
       [5.5, 2.4, 3.8, 1.1],
       [6.3, 2.7, 4.9, 1.8],
       [6.3, 2.8, 5.1, 1.5],
       [4.9, 2.5, 4.5, 1.7],
       [6.3, 2.5, 5. , 1.9],
       [7. , 3.2, 4.7, 1.4],
       [6.5, 3. , 5.2, 2. ],
       [6. , 3.4, 4.5, 1.6],
       [4.8, 3.1, 1.6, 0.2],
       [5.8, 2.7, 5.1, 1.9],
       [5.6, 2.7, 4.2, 1.3],
       [5.6, 2.9, 3.6, 1.3],
       [5.5, 2.5, 4. , 1.3],
       [6.1, 3. , 4.6, 1.4],
```

```
[7.2, 3.2, 6. , 1.8],
[5.3, 3.7, 1.5, 0.2],
[4.3, 3. , 1.1, 0.1],
[6.4, 2.7, 5.3, 1.9],
[5.7, 3. , 4.2, 1.2],
[5.4, 3.4, 1.7, 0.2],
[5.7, 4.4, 1.5, 0.4],
[6.9, 3.1, 4.9, 1.5],
[4.6, 3.1, 1.5, 0.2],
[5.9, 3. , 5.1, 1.8],
[5.1, 2.5, 3. , 1.1],
[4.6, 3.4, 1.4, 0.3],
[6.2, 2.2, 4.5, 1.5],
[7.2, 3.6, 6.1, 2.5],
[5.7, 2.9, 4.2, 1.3],
[4.8, 3. , 1.4, 0.1],
[7.1, 3. , 5.9, 2.1],
[6.9, 3.2, 5.7, 2.3],
[6.5, 3. , 5.8, 2.2],
[6.4, 2.8, 5.6, 2.1],
[5.1, 3.8, 1.6, 0.2],
[4.8, 3.4, 1.6, 0.2],
[6.5, 3.2, 5.1, 2. ],
[6.7, 3.3, 5.7, 2.1],
[4.5, 2.3, 1.3, 0.3],
[6.2, 3.4, 5.4, 2.3],
[4.9, 3. , 1.4, 0.2],
[5.7, 2.5, 5. , 2. ],
[6.9, 3.1, 5.4, 2.1],
[4.4, 3.2, 1.3, 0.2],
[5. , 3.6, 1.4, 0.2],
[7.2, 3. , 5.8, 1.6],
[5.1, 3.5, 1.4, 0.3],
[4.4, 3. , 1.3, 0.2],
[5.4, 3.9, 1.7, 0.4],
[5.5, 2.3, 4. , 1.3],
[6.8, 3.2, 5.9, 2.3],
[7.6, 3. , 6.6, 2.1],
[5.1, 3.5, 1.4, 0.2],
[4.9, 3.1, 1.5, 0.2],
[5.2, 3.4, 1.4, 0.2],
[5.7, 2.8, 4.5, 1.3],
[6.6, 3. , 4.4, 1.4],
[5. , 3.2, 1.2, 0.2],
[5.1, 3.3, 1.7, 0.5],
[6.4, 2.9, 4.3, 1.3],
[5.4, 3.4, 1.5, 0.4],
```

```
       [7.7, 2.6, 6.9, 2.3],
       [4.9, 2.4, 3.3, 1. ],
       [7.9, 3.8, 6.4, 2. ],
       [6.7, 3.1, 4.4, 1.4],
       [5.2, 4.1, 1.5, 0.1],
       [6. , 3. , 4.8, 1.8],
       [5.8, 4. , 1.2, 0.2],
       [7.7, 2.8, 6.7, 2. ],
       [5.1, 3.8, 1.5, 0.3],
       [4.7, 3.2, 1.6, 0.2],
       [7.4, 2.8, 6.1, 1.9],
       [5. , 3.3, 1.4, 0.2],
       [6.3, 3.4, 5.6, 2.4],
       [5.7, 2.8, 4.1, 1.3],
       [5.8, 2.7, 3.9, 1.2],
       [5.7, 2.6, 3.5, 1. ],
       [6.4, 3.2, 5.3, 2.3],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 4.9, 1.5],
       [6.7, 3. , 5. , 1.7],
       [5. , 3. , 1.6, 0.2],
       [5.5, 2.4, 3.7, 1. ],
       [6.7, 3.1, 5.6, 2.4],
       [5.8, 2.7, 5.1, 1.9],
       [5.1, 3.4, 1.5, 0.2],
       [6.6, 2.9, 4.6, 1.3],
       [5.6, 3. , 4.1, 1.3],
       [5.9, 3.2, 4.8, 1.8],
       [6.3, 2.3, 4.4, 1.3],
       [5.5, 3.5, 1.3, 0.2],
       [5.1, 3.7, 1.5, 0.4],
       [4.9, 3.1, 1.5, 0.1],
       [6.3, 2.9, 5.6, 1.8],
       [5.8, 2.7, 4.1, 1. ],
       [7.7, 3.8, 6.7, 2.2],
       [4.6, 3.2, 1.4, 0.2]])
```

[153]:
```
X_test.shape
#see the number of instances and attributes  of Test data

print(y_train)
print()
y_train
```

```
[1 1 2 0 2 0 0 1 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 0 2 1 1 1 1 2 0 0 2 1 0 0 1
 0 2 1 0 1 2 1 0 2 2 2 2 0 0 2 2 0 2 0 2 2 0 0 2 0 0 0 1 2 2 0 0 0 1 1 0 0
 1 0 2 1 2 1 0 2 0 2 0 0 2 0 2 1 1 1 2 2 1 1 0 1 2 2 0 1 1 1 1 0 0 0 2 1 2
```

0]

[153]:
```
array([1, 1, 2, 0, 2, 0, 0, 1, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2,
       1, 0, 2, 1, 1, 1, 1, 2, 0, 0, 2, 1, 0, 0, 1, 0, 2, 1, 0, 1, 2, 1,
       0, 2, 2, 2, 2, 0, 0, 2, 2, 0, 2, 0, 2, 2, 0, 0, 2, 0, 0, 0, 1, 2,
       2, 0, 0, 0, 1, 1, 0, 0, 1, 0, 2, 1, 2, 1, 0, 2, 0, 2, 0, 0, 2, 0,
       2, 1, 1, 1, 2, 2, 1, 1, 0, 1, 2, 2, 0, 1, 1, 1, 1, 0, 0, 0, 2, 1,
       2, 0])
```

[164]:
```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(3, 3, figsize=(15,15))
plt.suptitle("iris_pairplot")

for j in range(3):
    for i in range(3):

        print("i=",i,"j=",j,"i +1=",i+1)
        ax[i, j].scatter(X_train[:,j], X_train[:, i +1 ], c=y_train, s=70)

        ␣
 ↪print("------------------------------------------------------------------------------------
        print("X_train[:,j] =",X_train[:,j])
        print()
        print("X_train[:, i +1 ]=",X_train[:, i +1 ])
        ␣
 ↪print("------------------------------------------------------------------------------------
        ax[i, j].set_xticks(())
        ax[i, j].set_yticks(())
        if i == 2:
            ax[i, j].set_xlabel(iris["feature_names"][j])
        if j == 0:
            ax[i, j].set_ylabel(iris["feature_names"][i +1])
        if j > i:
            ax[i, j].set_visible(False)


print(y_train)
```

```
i= 0 j= 0 i +1= 1
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [5.9 5.8 6.8 4.7 6.9 5.  5.4 5.  6.5 6.7 6.  6.7 5.6 7.7 6.3 5.5
6.3 6.3
 4.9 6.3 7.  6.5 6.  4.8 5.8 5.6 5.6 5.5 6.1 7.2 5.3 4.3 6.4 5.7 5.4 5.7
 6.9 4.6 5.9 5.1 4.6 6.2 7.2 5.7 4.8 7.1 6.9 6.5 6.4 5.1 4.8 6.5 6.7 4.5
 6.2 4.9 5.7 6.9 4.4 5.  7.2 5.1 4.4 5.4 5.5 6.8 7.6 5.1 4.9 5.2 5.7 6.6
```

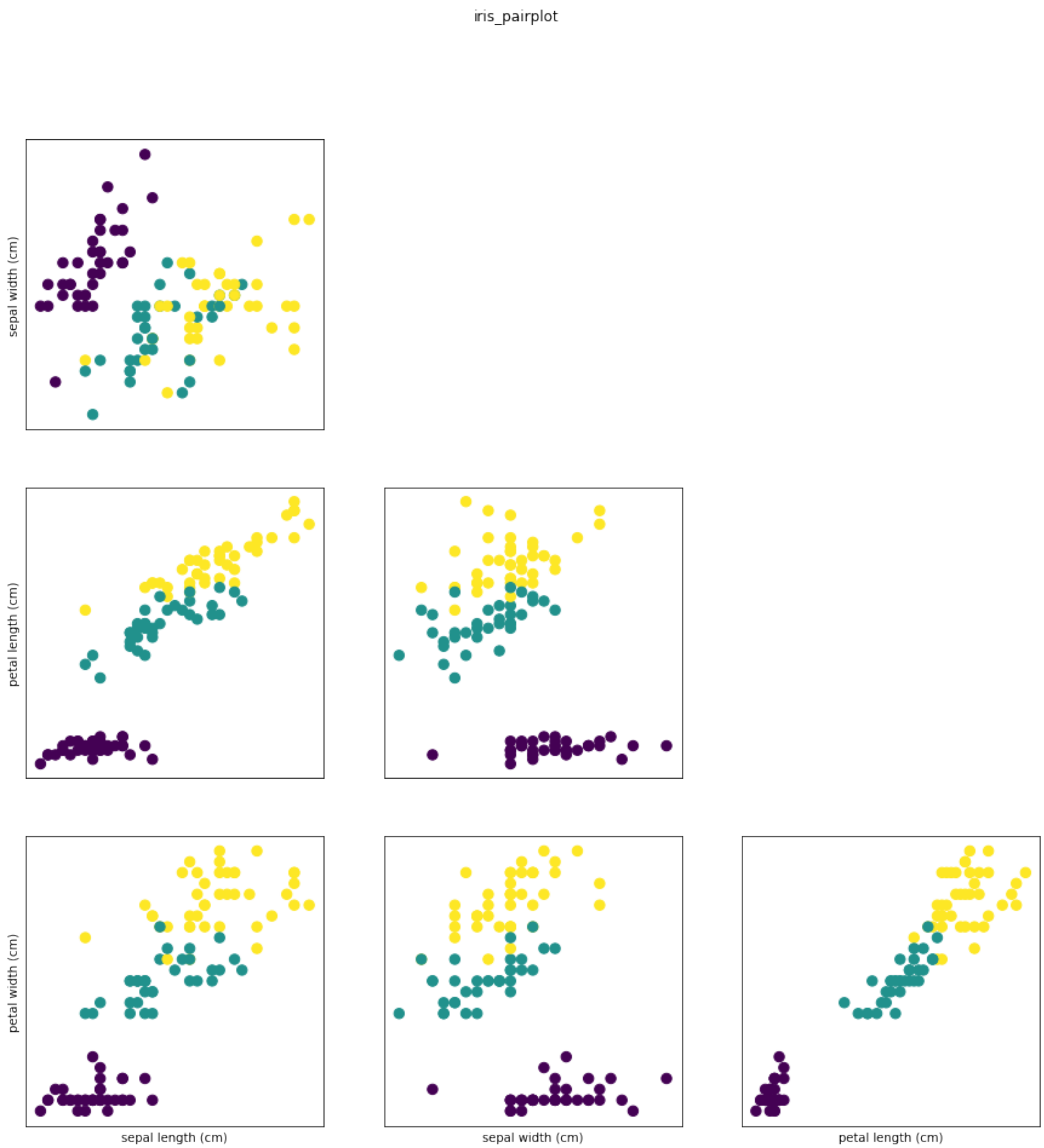
```
 5.  5.1 6.4 5.4 7.7 4.9 7.9 6.7 5.2 6.  5.8 7.7 5.1 4.7 7.4 5.  6.3 5.7
 5.8 5.7 6.4 6.7 6.3 6.7 5.  5.5 6.7 5.8 5.1 6.6 5.6 5.9 6.3 5.5 5.1 4.9
 6.3 5.8 7.7 4.6]

X_train[:, i +1 ]= [3.  2.6 3.  3.2 3.1 3.5 3.7 2.  3.  3.3 2.2 2.5 2.5 3.  3.3
2.4 2.7 2.8
 2.5 2.5 3.2 3.  3.4 3.1 2.7 2.7 2.9 2.5 3.  3.2 3.7 3.  2.7 3.  3.4 4.4
 3.1 3.1 3.  2.5 3.4 2.2 3.6 2.9 3.  3.  3.2 3.  2.8 3.8 3.4 3.2 3.3 2.3
 3.4 3.  2.5 3.1 3.2 3.6 3.  3.5 3.  3.9 2.3 3.2 3.  3.5 3.1 3.4 2.8 3.
 3.2 3.3 2.9 3.4 2.6 2.4 3.8 3.1 4.1 3.  4.  2.8 3.8 3.2 2.8 3.3 3.4 2.8
 2.7 2.6 3.2 3.  2.5 3.  3.  2.4 3.1 2.7 3.4 2.9 3.  3.2 2.3 3.5 3.7 3.1
 2.9 2.7 3.8 3.2]
--------------------------------------------------------------------------------
----------------
i= 1 j= 0 i +1= 2
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [5.9 5.8 6.8 4.7 6.9 5.  5.4 5.  6.5 6.7 6.  6.7 5.6 7.7 6.3 5.5
6.3 6.3
 4.9 6.3 7.  6.5 6.  4.8 5.8 5.6 5.6 5.5 6.1 7.2 5.3 4.3 6.4 5.7 5.4 5.7
 6.9 4.6 5.9 5.1 4.6 6.2 7.2 5.7 4.8 7.1 6.9 6.5 6.4 5.1 4.8 6.5 6.7 4.5
 6.2 4.9 5.7 6.9 4.4 5.  7.2 5.1 4.4 5.4 5.5 6.8 7.6 5.1 4.9 5.2 5.7 6.6
 5.  5.1 6.4 5.4 7.7 4.9 7.9 6.7 5.2 6.  5.8 7.7 5.1 4.7 7.4 5.  6.3 5.7
 5.8 5.7 6.4 6.7 6.3 6.7 5.  5.5 6.7 5.8 5.1 6.6 5.6 5.9 6.3 5.5 5.1 4.9
 6.3 5.8 7.7 4.6]

X_train[:, i +1 ]= [4.2 4.  5.5 1.3 5.1 1.6 1.5 3.5 5.5 5.7 5.  5.8 3.9 6.1 4.7
3.8 4.9 5.1
 4.5 5.  4.7 5.2 4.5 1.6 5.1 4.2 3.6 4.  4.6 6.  1.5 1.1 5.3 4.2 1.7 1.5
 4.9 1.5 5.1 3.  1.4 4.5 6.1 4.2 1.4 5.9 5.7 5.8 5.6 1.6 1.6 5.1 5.7 1.3
 5.4 1.4 5.  5.4 1.3 1.4 5.8 1.4 1.3 1.7 4.  5.9 6.6 1.4 1.5 1.4 4.5 4.4
 1.2 1.7 4.3 1.5 6.9 3.3 6.4 4.4 1.5 4.8 1.2 6.7 1.5 1.6 6.1 1.4 5.6 4.1
 3.9 3.5 5.3 5.2 4.9 5.  1.6 3.7 5.6 5.1 1.5 4.6 4.1 4.8 4.4 1.3 1.5 1.5
 5.6 4.1 6.7 1.4]
--------------------------------------------------------------------------------
----------------
i= 2 j= 0 i +1= 3
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [5.9 5.8 6.8 4.7 6.9 5.  5.4 5.  6.5 6.7 6.  6.7 5.6 7.7 6.3 5.5
6.3 6.3
 4.9 6.3 7.  6.5 6.  4.8 5.8 5.6 5.6 5.5 6.1 7.2 5.3 4.3 6.4 5.7 5.4 5.7
 6.9 4.6 5.9 5.1 4.6 6.2 7.2 5.7 4.8 7.1 6.9 6.5 6.4 5.1 4.8 6.5 6.7 4.5
 6.2 4.9 5.7 6.9 4.4 5.  7.2 5.1 4.4 5.4 5.5 6.8 7.6 5.1 4.9 5.2 5.7 6.6
 5.  5.1 6.4 5.4 7.7 4.9 7.9 6.7 5.2 6.  5.8 7.7 5.1 4.7 7.4 5.  6.3 5.7
 5.8 5.7 6.4 6.7 6.3 6.7 5.  5.5 6.7 5.8 5.1 6.6 5.6 5.9 6.3 5.5 5.1 4.9
 6.3 5.8 7.7 4.6]
```

```
X_train[:, i +1 ]= [1.5 1.2 2.1 0.2 2.3 0.6 0.2 1.  1.8 2.5 1.5 1.8 1.1 2.3 1.6
1.1 1.8 1.5
 1.7 1.9 1.4 2.  1.6 0.2 1.9 1.3 1.3 1.3 1.4 1.8 0.2 0.1 1.9 1.2 0.2 0.4
 1.5 0.2 1.8 1.1 0.3 1.5 2.5 1.3 0.1 2.1 2.3 2.2 2.1 0.2 0.2 2.  2.1 0.3
 2.3 0.2 2.  2.1 0.2 0.2 1.6 0.3 0.2 0.4 1.3 2.3 2.1 0.2 0.2 0.2 1.3 1.4
 0.2 0.5 1.3 0.4 2.3 1.  2.  1.4 0.1 1.8 0.2 2.  0.3 0.2 1.9 0.2 2.4 1.3
 1.2 1.  2.3 2.3 1.5 1.7 0.2 1.  2.4 1.9 0.2 1.3 1.3 1.8 1.3 0.2 0.4 0.1
 1.8 1.  2.2 0.2]
--------------------------------------------------------------------------------
----------------
i= 0 j= 1 i +1= 1
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [3.  2.6 3.  3.2 3.1 3.5 3.7 2.  3.  3.3 2.2 2.5 2.5 3.  3.3 2.4
2.7 2.8
 2.5 2.5 3.2 3.  3.4 3.1 2.7 2.7 2.9 2.5 3.  3.2 3.7 3.  2.7 3.  3.4 4.4
 3.1 3.1 3.  2.5 3.4 2.2 3.6 2.9 3.  3.  3.2 3.  2.8 3.8 3.4 3.2 3.3 2.3
 3.4 3.  2.5 3.1 3.2 3.6 3.  3.5 3.  3.9 2.3 3.2 3.  3.5 3.1 3.4 2.8 3.
 3.2 3.3 2.9 3.4 2.6 2.4 3.8 3.1 4.1 3.  4.  2.8 3.8 3.2 2.8 3.3 3.4 2.8
 2.7 2.6 3.2 3.  2.5 3.  3.  2.4 3.1 2.7 3.4 2.9 3.  3.2 2.3 3.5 3.7 3.1
 2.9 2.7 3.8 3.2]

X_train[:, i +1 ]= [3.  2.6 3.  3.2 3.1 3.5 3.7 2.  3.  3.3 2.2 2.5 2.5 3.  3.3
2.4 2.7 2.8
 2.5 2.5 3.2 3.  3.4 3.1 2.7 2.7 2.9 2.5 3.  3.2 3.7 3.  2.7 3.  3.4 4.4
 3.1 3.1 3.  2.5 3.4 2.2 3.6 2.9 3.  3.  3.2 3.  2.8 3.8 3.4 3.2 3.3 2.3
 3.4 3.  2.5 3.1 3.2 3.6 3.  3.5 3.  3.9 2.3 3.2 3.  3.5 3.1 3.4 2.8 3.
 3.2 3.3 2.9 3.4 2.6 2.4 3.8 3.1 4.1 3.  4.  2.8 3.8 3.2 2.8 3.3 3.4 2.8
 2.7 2.6 3.2 3.  2.5 3.  3.  2.4 3.1 2.7 3.4 2.9 3.  3.2 2.3 3.5 3.7 3.1
 2.9 2.7 3.8 3.2]
--------------------------------------------------------------------------------
----------------
i= 1 j= 1 i +1= 2
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [3.  2.6 3.  3.2 3.1 3.5 3.7 2.  3.  3.3 2.2 2.5 2.5 3.  3.3 2.4
2.7 2.8
 2.5 2.5 3.2 3.  3.4 3.1 2.7 2.7 2.9 2.5 3.  3.2 3.7 3.  2.7 3.  3.4 4.4
 3.1 3.1 3.  2.5 3.4 2.2 3.6 2.9 3.  3.  3.2 3.  2.8 3.8 3.4 3.2 3.3 2.3
 3.4 3.  2.5 3.1 3.2 3.6 3.  3.5 3.  3.9 2.3 3.2 3.  3.5 3.1 3.4 2.8 3.
 3.2 3.3 2.9 3.4 2.6 2.4 3.8 3.1 4.1 3.  4.  2.8 3.8 3.2 2.8 3.3 3.4 2.8
 2.7 2.6 3.2 3.  2.5 3.  3.  2.4 3.1 2.7 3.4 2.9 3.  3.2 2.3 3.5 3.7 3.1
 2.9 2.7 3.8 3.2]

X_train[:, i +1 ]= [4.2 4.  5.5 1.3 5.1 1.6 1.5 3.5 5.5 5.7 5.  5.8 3.9 6.1 4.7
3.8 4.9 5.1
 4.5 5.  4.7 5.2 4.5 1.6 5.1 4.2 3.6 4.  4.6 6.  1.5 1.1 5.3 4.2 1.7 1.5
 4.9 1.5 5.1 3.  1.4 4.5 6.1 4.2 1.4 5.9 5.7 5.8 5.6 1.6 1.6 5.1 5.7 1.3
```

```
 5.4 1.4 5.  5.4 1.3 1.4 5.8 1.4 1.3 1.7 4.  5.9 6.6 1.4 1.5 1.4 4.5 4.4
 1.2 1.7 4.3 1.5 6.9 3.3 6.4 4.4 1.5 4.8 1.2 6.7 1.5 1.6 6.1 1.4 5.6 4.1
 3.9 3.5 5.3 5.2 4.9 5.  1.6 3.7 5.6 5.1 1.5 4.6 4.1 4.8 4.4 1.3 1.5 1.5
 5.6 4.1 6.7 1.4]
--------------------------------------------------------------------------------
----------------
i= 2 j= 1 i +1= 3
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [3.  2.6 3.  3.2 3.1 3.5 3.7 2.  3.  3.3 2.2 2.5 2.5 3.  3.3 2.4
2.7 2.8
 2.5 2.5 3.2 3.  3.4 3.1 2.7 2.7 2.9 2.5 3.  3.2 3.7 3.  2.7 3.  3.4 4.4
 3.1 3.1 3.  2.5 3.4 2.2 3.6 2.9 3.  3.  3.2 3.  2.8 3.8 3.4 3.2 3.3 2.3
 3.4 3.  2.5 3.1 3.2 3.6 3.  3.5 3.  3.9 2.3 3.2 3.  3.5 3.1 3.4 2.8 3.
 3.2 3.3 2.9 3.4 2.6 2.4 3.8 3.1 4.1 3.  4.  2.8 3.8 3.2 2.8 3.3 3.4 2.8
 2.7 2.6 3.2 3.  2.5 3.  3.  2.4 3.1 2.7 3.4 2.9 3.  3.2 2.3 3.5 3.7 3.1
 2.9 2.7 3.8 3.2]

X_train[:, i +1 ]= [1.5 1.2 2.1 0.2 2.3 0.6 0.2 1.  1.8 2.5 1.5 1.8 1.1 2.3 1.6
1.1 1.8 1.5
 1.7 1.9 1.4 2.  1.6 0.2 1.9 1.3 1.3 1.3 1.4 1.8 0.2 0.1 1.9 1.2 0.2 0.4
 1.5 0.2 1.8 1.1 0.3 1.5 2.5 1.3 0.1 2.1 2.3 2.2 2.1 0.2 0.2 2.  2.1 0.3
 2.3 0.2 2.  2.1 0.2 0.2 1.6 0.3 0.2 0.4 1.3 2.3 2.1 0.2 0.2 0.2 1.3 1.4
 0.2 0.5 1.3 0.4 2.3 1.  2.  1.4 0.1 1.8 0.2 2.  0.3 0.2 1.9 0.2 2.4 1.3
 1.2 1.  2.3 2.3 1.5 1.7 0.2 1.  2.4 1.9 0.2 1.3 1.3 1.8 1.3 0.2 0.4 0.1
 1.8 1.  2.2 0.2]
--------------------------------------------------------------------------------
----------------
i= 0 j= 2 i +1= 1
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [4.2 4.  5.5 1.3 5.1 1.6 1.5 3.5 5.5 5.7 5.  5.8 3.9 6.1 4.7 3.8
4.9 5.1
 4.5 5.  4.7 5.2 4.5 1.6 5.1 4.2 3.6 4.  4.6 6.  1.5 1.1 5.3 4.2 1.7 1.5
 4.9 1.5 5.1 3.  1.4 4.5 6.1 4.2 1.4 5.9 5.7 5.8 5.6 1.6 1.6 5.1 5.7 1.3
 5.4 1.4 5.  5.4 1.3 1.4 5.8 1.4 1.3 1.7 4.  5.9 6.6 1.4 1.5 1.4 4.5 4.4
 1.2 1.7 4.3 1.5 6.9 3.3 6.4 4.4 1.5 4.8 1.2 6.7 1.5 1.6 6.1 1.4 5.6 4.1
 3.9 3.5 5.3 5.2 4.9 5.  1.6 3.7 5.6 5.1 1.5 4.6 4.1 4.8 4.4 1.3 1.5 1.5
 5.6 4.1 6.7 1.4]

X_train[:, i +1 ]= [3.  2.6 3.  3.2 3.1 3.5 3.7 2.  3.  3.3 2.2 2.5 2.5 3.  3.3
2.4 2.7 2.8
 2.5 2.5 3.2 3.  3.4 3.1 2.7 2.7 2.9 2.5 3.  3.2 3.7 3.  2.7 3.  3.4 4.4
 3.1 3.1 3.  2.5 3.4 2.2 3.6 2.9 3.  3.  3.2 3.  2.8 3.8 3.4 3.2 3.3 2.3
 3.4 3.  2.5 3.1 3.2 3.6 3.  3.5 3.  3.9 2.3 3.2 3.  3.5 3.1 3.4 2.8 3.
 3.2 3.3 2.9 3.4 2.6 2.4 3.8 3.1 4.1 3.  4.  2.8 3.8 3.2 2.8 3.3 3.4 2.8
 2.7 2.6 3.2 3.  2.5 3.  3.  2.4 3.1 2.7 3.4 2.9 3.  3.2 2.3 3.5 3.7 3.1
 2.9 2.7 3.8 3.2]
```

```
--------------------------------------------------------------------------------
----------------
i= 1 j= 2 i +1= 2
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [4.2 4.  5.5 1.3 5.1 1.6 1.5 3.5 5.5 5.7 5.  5.8 3.9 6.1 4.7 3.8
4.9 5.1
 4.5 5.  4.7 5.2 4.5 1.6 5.1 4.2 3.6 4.  4.6 6.  1.5 1.1 5.3 4.2 1.7 1.5
 4.9 1.5 5.1 3.  1.4 4.5 6.1 4.2 1.4 5.9 5.7 5.8 5.6 1.6 1.6 5.1 5.7 1.3
 5.4 1.4 5.  5.4 1.3 1.4 5.8 1.4 1.3 1.7 4.  5.9 6.6 1.4 1.5 1.4 4.5 4.4
 1.2 1.7 4.3 1.5 6.9 3.3 6.4 4.4 1.5 4.8 1.2 6.7 1.5 1.6 6.1 1.4 5.6 4.1
 3.9 3.5 5.3 5.2 4.9 5.  1.6 3.7 5.6 5.1 1.5 4.6 4.1 4.8 4.4 1.3 1.5 1.5
 5.6 4.1 6.7 1.4]

X_train[:, i +1 ]= [4.2 4.  5.5 1.3 5.1 1.6 1.5 3.5 5.5 5.7 5.  5.8 3.9 6.1 4.7
3.8 4.9 5.1
 4.5 5.  4.7 5.2 4.5 1.6 5.1 4.2 3.6 4.  4.6 6.  1.5 1.1 5.3 4.2 1.7 1.5
 4.9 1.5 5.1 3.  1.4 4.5 6.1 4.2 1.4 5.9 5.7 5.8 5.6 1.6 1.6 5.1 5.7 1.3
 5.4 1.4 5.  5.4 1.3 1.4 5.8 1.4 1.3 1.7 4.  5.9 6.6 1.4 1.5 1.4 4.5 4.4
 1.2 1.7 4.3 1.5 6.9 3.3 6.4 4.4 1.5 4.8 1.2 6.7 1.5 1.6 6.1 1.4 5.6 4.1
 3.9 3.5 5.3 5.2 4.9 5.  1.6 3.7 5.6 5.1 1.5 4.6 4.1 4.8 4.4 1.3 1.5 1.5
 5.6 4.1 6.7 1.4]
--------------------------------------------------------------------------------
----------------
i= 2 j= 2 i +1= 3
--------------------------------------------------------------------------------
----------------
X_train[:,j] = [4.2 4.  5.5 1.3 5.1 1.6 1.5 3.5 5.5 5.7 5.  5.8 3.9 6.1 4.7 3.8
4.9 5.1
 4.5 5.  4.7 5.2 4.5 1.6 5.1 4.2 3.6 4.  4.6 6.  1.5 1.1 5.3 4.2 1.7 1.5
 4.9 1.5 5.1 3.  1.4 4.5 6.1 4.2 1.4 5.9 5.7 5.8 5.6 1.6 1.6 5.1 5.7 1.3
 5.4 1.4 5.  5.4 1.3 1.4 5.8 1.4 1.3 1.7 4.  5.9 6.6 1.4 1.5 1.4 4.5 4.4
 1.2 1.7 4.3 1.5 6.9 3.3 6.4 4.4 1.5 4.8 1.2 6.7 1.5 1.6 6.1 1.4 5.6 4.1
 3.9 3.5 5.3 5.2 4.9 5.  1.6 3.7 5.6 5.1 1.5 4.6 4.1 4.8 4.4 1.3 1.5 1.5
 5.6 4.1 6.7 1.4]

X_train[:, i +1 ]= [1.5 1.2 2.1 0.2 2.3 0.6 0.2 1.  1.8 2.5 1.5 1.8 1.1 2.3 1.6
1.1 1.8 1.5
 1.7 1.9 1.4 2.  1.6 0.2 1.9 1.3 1.3 1.3 1.4 1.8 0.2 0.1 1.9 1.2 0.2 0.4
 1.5 0.2 1.8 1.1 0.3 1.5 2.5 1.3 0.1 2.1 2.3 2.2 2.1 0.2 0.2 2.  2.1 0.3
 2.3 0.2 2.  2.1 0.2 0.2 1.6 0.3 0.2 0.4 1.3 2.3 2.1 0.2 0.2 0.2 1.3 1.4
 0.2 0.5 1.3 0.4 2.3 1.  2.  1.4 0.1 1.8 0.2 2.  0.3 0.2 1.9 0.2 2.4 1.3
 1.2 1.  2.3 2.3 1.5 1.7 0.2 1.  2.4 1.9 0.2 1.3 1.3 1.8 1.3 0.2 0.4 0.1
 1.8 1.  2.2 0.2]
--------------------------------------------------------------------------------
----------------
[1 1 2 0 2 0 0 1 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 0 2 1 1 1 1 2 0 0 2 1 0 0 1
 0 2 1 0 1 2 1 0 2 2 2 2 0 0 2 2 0 2 0 2 2 0 0 2 0 0 0 1 2 2 0 0 0 1 1 0 0
```

```
1 0 2 1 2 1 0 2 0 2 0 0 2 0 2 1 1 1 2 2 1 1 0 1 2 2 0 1 1 1 1 0 0 0 2 1 2
0]
```



As you can see, without using a Machine Learning algorithm you determined that there are certain ways to determine if your data can be classified for example plotting the data first, you can that there are attributes that reflect very well iris' classifications

matplotlib.pyplot.subplots

matplotlib.pyplot.subplots(nrows=1, ncols=1, sharex=False, sharey=False, squeeze=True, subplot_kw=None, gridspec_kw=None, **fig_kw)

[152]:
```
y_train
```

[152]:
```
array([1, 1, 2, 0, 2, 0, 0, 1, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2,
       1, 0, 2, 1, 1, 1, 1, 2, 0, 0, 2, 1, 0, 0, 1, 0, 2, 1, 0, 1, 2, 1,
       0, 2, 2, 2, 2, 0, 0, 2, 2, 0, 2, 0, 2, 2, 0, 0, 2, 0, 0, 0, 1, 2,
       2, 0, 0, 0, 1, 1, 0, 0, 1, 0, 2, 1, 2, 1, 0, 2, 0, 2, 0, 0, 2, 0,
       2, 1, 1, 1, 2, 2, 1, 1, 0, 1, 2, 2, 0, 1, 1, 1, 1, 0, 0, 0, 2, 1,
       2, 0])
```

[170]:
```
y_train.size
```

[170]:
```
112
```

[171]:
```
X_train.size
```

[171]:
```
448
```

[38]:
```
X_test
```

[38]:
```
array([[5.8, 2.8, 5.1, 2.4],
       [6. , 2.2, 4. , 1. ],
       [5.5, 4.2, 1.4, 0.2],
       [7.3, 2.9, 6.3, 1.8],
       [5. , 3.4, 1.5, 0.2],
       [6.3, 3.3, 6. , 2.5],
       [5. , 3.5, 1.3, 0.3],
       [6.7, 3.1, 4.7, 1.5],
       [6.8, 2.8, 4.8, 1.4],
       [6.1, 2.8, 4. , 1.3],
       [6.1, 2.6, 5.6, 1.4],
       [6.4, 3.2, 4.5, 1.5],
       [6.1, 2.8, 4.7, 1.2],
       [6.5, 2.8, 4.6, 1.5],
       [6.1, 2.9, 4.7, 1.4],
       [4.9, 3.6, 1.4, 0.1],
       [6. , 2.9, 4.5, 1.5],
       [5.5, 2.6, 4.4, 1.2],
       [4.8, 3. , 1.4, 0.3],
       [5.4, 3.9, 1.3, 0.4],
       [5.6, 2.8, 4.9, 2. ],
       [5.6, 3. , 4.5, 1.5],
       [4.8, 3.4, 1.9, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [6.2, 2.8, 4.8, 1.8],
       [4.6, 3.6, 1. , 0.2],
       [5.1, 3.8, 1.9, 0.4],
       [6.2, 2.9, 4.3, 1.3],
```

```
        [5. , 2.3, 3.3, 1. ],
        [5. , 3.4, 1.6, 0.4],
        [6.4, 3.1, 5.5, 1.8],
        [5.4, 3. , 4.5, 1.5],
        [5.2, 3.5, 1.5, 0.2],
        [6.1, 3. , 4.9, 1.8],
        [6.4, 2.8, 5.6, 2.2],
        [5.2, 2.7, 3.9, 1.4],
        [5.7, 3.8, 1.7, 0.3],
        [6. , 2.7, 5.1, 1.6]])
```

[52]:

```
{x= 0
for i in range(3):
    print("a")
    for j in range(3):
        print (i, j)
        x = x +1
```

```
a
0 0
0 1
0 2
a
1 0
1 1
1 2
a
2 0
2 1
2 2
```

[186]:

```
lol = np.array([[2,3,5,560] , [266,56,14,23], [22,35,14, 2],[2,33,18,56]])
print(lol)
maria = np.array([1, 2, 2, 0])
feature = np.array(["john", "luis", "2", "0"])
```

```
[[  2   3   5 560]
 [266  56  14  23]
 [ 22  35  14   2]
 [  2  33  18  56]]
```

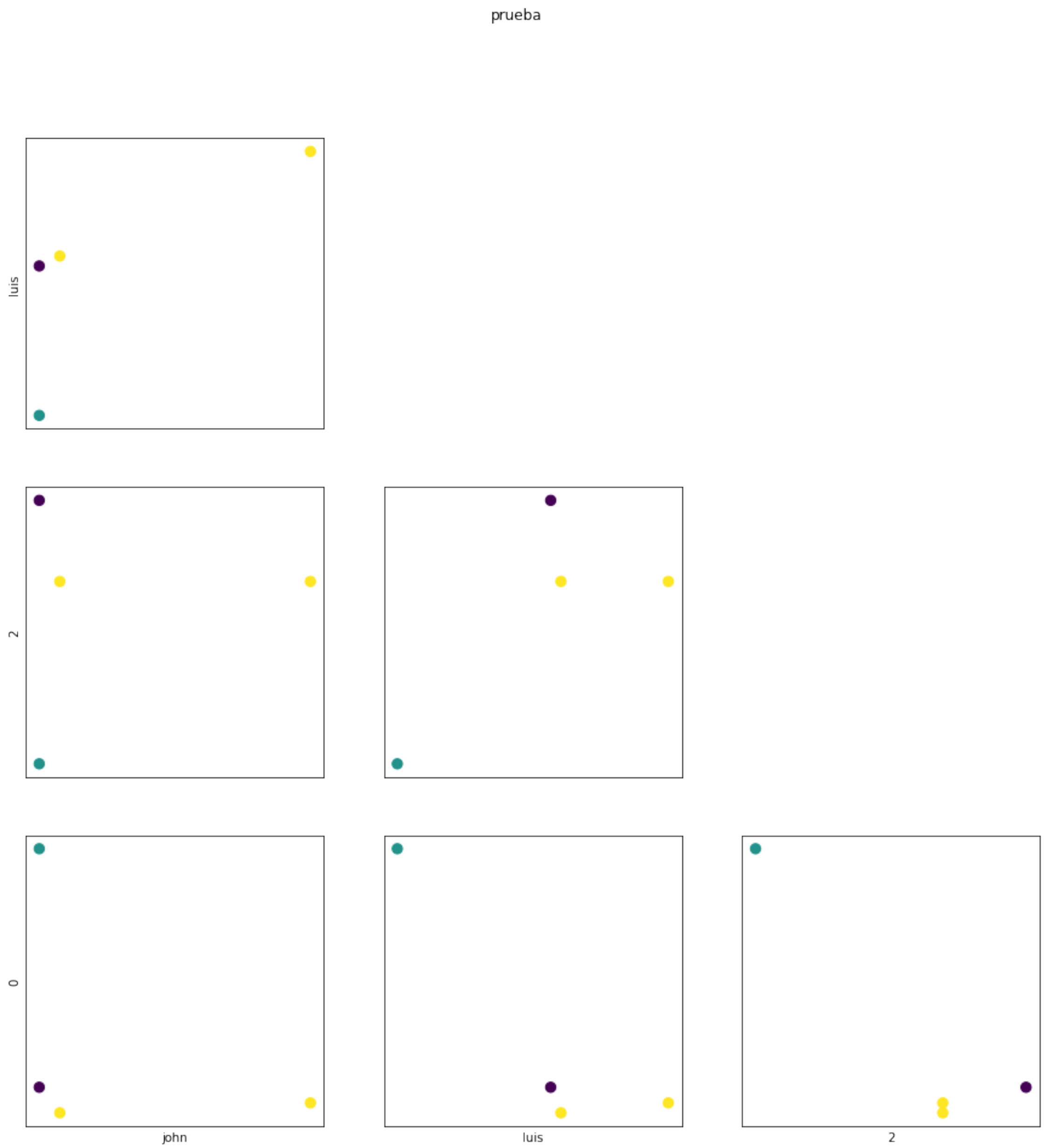
[187]:

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(3, 3, figsize=(15,15))
plt.suptitle("prueba")

for j in range(3):
```

```
    for i in range(3):
        ax[i, j].scatter(lol[:,j], lol[:, i +1 ], c=maria , s=70)
        print("i=",i,"j=",j,"  i+1=",i+1)
        print(lol[:,j])
        print(lol[:, i +1 ])
        print(".....")
        ax[i, j].set_xticks(())
        ax[i, j].set_yticks(())
        if i == 2:
            ax[i, j].set_xlabel(feature[j])
        if j == 0:
            ax[i, j].set_ylabel(feature[i +1])
        if j > i:
            ax[i, j].set_visible(False)
```

```
i= 0 j= 0   i+1= 1
[  2 266  22   2]
[ 3 56 35 33]
…
i= 1 j= 0   i+1= 2
[  2 266  22   2]
[ 5 14 14 18]
…
i= 2 j= 0   i+1= 3
[  2 266  22   2]
[560  23   2  56]
…
i= 0 j= 1   i+1= 1
[ 3 56 35 33]
[ 3 56 35 33]
…
i= 1 j= 1   i+1= 2
[ 3 56 35 33]
[ 5 14 14 18]
…
i= 2 j= 1   i+1= 3
[ 3 56 35 33]
[560  23   2  56]
…
i= 0 j= 2   i+1= 1
[ 5 14 14 18]
[ 3 56 35 33]
…
i= 1 j= 2   i+1= 2
[ 5 14 14 18]
[ 5 14 14 18]
```

```
…
i= 2 j= 2   i+1= 3
[ 5 14 14 18]
[560  23   2  56]
…
```




```
[ ]:
```

```
[ ]:
```