Working with Dataset

1. Import the load_iris dataset

Analyze the dataset and print below values. Learn what is the me values.

In [1]:
```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
df=load_iris()
```

In [2]:
```
df  #It's load data frame (iris _flower)
```

```
m\n        - class:\n                - Iris-Setosa\n
rsicolour\n                - Iris-Virginica\n
tatistics:\n\n    ============== ==== ==== ======= =====
\n                    Min  Max   Mean    SD   Class Corr
====== ==== ==== ======= ===== ====================\n
7.9   5.84   0.83    0.7826\n    sepal width:    2.0  4.
4194\n    petal length:   1.0  6.9   3.76   1.76    0.94
l width:    0.1  2.5   1.20   0.76    0.9565  (high!)\n
= ==== ======= ===== ====================\n\n    :Missir
one\n    :Class Distribution: 33.3% for each of 3 classe
A. Fisher\n    :Donor: Michael Marshall (MARSHALL%PLU@ic
Date: July, 1988\n\nThe famous Iris database, first usec
The dataset is taken\nfrom Fisher\'s paper. Note that it
but not as in the UCI\nMachine Learning Repository, whic
points.\n\nThis is perhaps the best known database to be
n recognition literature.  Fisher\'s paper is a classic
referenced frequently to this day.  (See Duda & Hart, fc
a set contains 3 classes of 50 instances each, where eac
type of iris plant.  One class is linearly separable frc
atter are NOT linearly separable from each other.\n\n..
```

In [3]:
```
print(df.data)    #Only print it's data value
```

```
[[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]
```

In [4]:
```
df.data.shape  #That is shape or dimention of data model
```

Out[4]:
```
(150, 4)
```

In [5]:
```
df.feature_names      # Here by , Total 4 column which
```

Out[5]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [6]:
```
df.target_names    # this respresent the Three diffrent
```

Out[6]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U1
```

In [7]:
```
df.data  # It's data given by obesarvation
```

```
       [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],
```

In [8]:
```
df.target          # Target value in numpy as given bel
```

Out[8]:
```
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, 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
```

In [9]:
```
df.DESCR     # Description of DATA
```

Out[9]:
```
'.. _iris_dataset:\n\nIris plants dataset\n-------------
haracteristics:**\n\n    :Number of Instances: 150 (50 i
s)\n    :Number of Attributes: 4 numeric, predictive att
:Attribute Information:\n        - sepal length in cm\n
cm\n        - petal length in cm\n        - petal width
s:\n                - Iris-Setosa\n                - Iri
- Iris-Virginica\n                \n    :Summary Statist
=== ==== ==== ======= ===== ====================\n
Mean    SD   Class Correlation\n    ============== ====
================\n    sepal length:   4.3  7.9   5.84
al width:    2.0  4.4   3.05   0.43   -0.4194\n    petal
3.76   1.76    0.9490  (high!)\n    petal width:    0.1
0.9565  (high!)\n    ============== ==== ==== ======= ==
=\n\n    :Missing Attribute Values: None\n    :Class Dis
ch of 3 classes.\n    :Creator: R.A. Fisher\n    :Donor:
HALL%PLU@io.arc.nasa.gov)\n    :Date: July, 1988\n\nThe
irst used by Sir R.A. Fisher. The dataset is taken\nfrom
that it\'s the same as in R, but not as in the UCI\nMach
y, which has two wrong data points.\n\nThis is perhaps t
to be found in the\npattern recognition literature.  Fis
ic in the field and\nis referenced frequently to this da
or example.)  The\ndata set contains 3 classes of 50 ins
class refers to a\ntype of iris plant.  One class is lin
e other 2; the\nlatter are NOT linearly separable from e
c:: References\n\n   - Fisher, R.A. "The use of multiple
mic problems"\n     Annual Eugenics, 7, Part II, 179-188
ibutions to\n     Mathematical Statistics" (John Wiley,
R.O., & Hart, P.E. (1973) Pattern Classification and Sce
7.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page
B.V. (1980) "Nosing Around the Neighborhood: A New Syste
lassification Rule for Recognition in Partially Exposed\
EEE Transactions on Pattern Analysis and Machine\n     I
2, No. 1, 67-71.\n   - Gates, G.W. (1972) "The Reduced N
IEEE Transactions\n     on Information Theory, May 1972,
o: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOC
l clustering system finds 3 classes in the data.\n   - M
```

In [ ]:
```

```

2. Write a logic to Filter the rows of iris data that h and sepallength < 5.0

In [10]:
```
for i in df.data:
    for j in i:
      if i[0] < 5 and i[2] >1.5:
            print(i)
print("\n")
```

```
[4.8 3.4 1.6 0.2]
[4.8 3.4 1.6 0.2]
[4.8 3.4 1.6 0.2]
[4.8 3.4 1.6 0.2]
[4.8 3.4 1.9 0.2]
[4.8 3.4 1.9 0.2]
[4.8 3.4 1.9 0.2]
[4.8 3.4 1.9 0.2]
[4.7 3.2 1.6 0.2]
[4.7 3.2 1.6 0.2]
[4.7 3.2 1.6 0.2]
[4.7 3.2 1.6 0.2]
[4.8 3.1 1.6 0.2]
[4.8 3.1 1.6 0.2]
[4.8 3.1 1.6 0.2]
[4.8 3.1 1.6 0.2]
[4.9 2.4 3.3 1. ]
[4.9 2.4 3.3 1. ]
[4.9 2.4 3.3 1. ]
[4.9 2.4 3.3 1. ]
[4.9 2.5 4.5 1.7]
[4.9 2.5 4.5 1.7]
[4.9 2.5 4.5 1.7]
[4.9 2.5 4.5 1.7]
```

3. Calculate the mean, median and standard deviat

In [11]:
```
Mean = np.mean(df.data)
Mean                           #Mean of data
```

Out[11]: 3.4644999999999997

In [12]:
```
Median = np.median(df.data)
Median                       #Median of data
```

Out[12]: 3.2

In [13]:
```
STd = np.std(df.data)
STd                          #Starndard deviation
```

Out[13]: 1.9738430577598278

4. Use the petal length (3rd) column of iris data to categorical data, such that if petal length is:

a. Less than 3 --> 'Small'

b. Between 3to 5 --> 'Medium'

c. Greater than 5 --> 'Large'

In [14]:

```
for i in df.data:
    for j in i:
        if i[2] < 3 :
            print("Smalller : {} ".format(i[2]))
        elif i[2] >= 3 and i[2] <= 5 :
            print("Medium : {} ".format(i[2]))
        else :
            print("Larger : {}".format(i[2]))
```

```
Larger : 5.1
Larger : 5.1
Larger : 5.1
Larger : 5.1
Larger : 5.9
Larger : 5.9
Larger : 5.9
Larger : 5.9
Larger : 5.7
Larger : 5.7
Larger : 5.7
Larger : 5.7
Larger : 5.2
Larger : 5.2
Larger : 5.2
Larger : 5.2
Medium : 5.0
Medium : 5.0
Medium : 5.0
```

In [ ]: