In [4]:

```python
import pandas as pd
```

In [12]:

```python
data = pd.read_csv(r"C:\Users\admin\Desktop\Iris.csv") #Load Iris.csv into a Pandas
```

In [13]:

```python
data.head()
```

Out[13]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [14]:

```python
data.tail()
```

Out[14]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

In [15]:

```python
data.head(10)
```

Out[15]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

In [16]:

```python
data.sample(5) # #Displaying the number of rows randomly
```

Out[16]:

	sepal_length	sepal_width	petal_length	petal_width	species
118	7.7	2.6	6.9	2.3	Iris-virginica
55	5.7	2.8	4.5	1.3	Iris-versicolor

	sepal_length	sepal_width	petal_length	petal_width	species
59	5.2	2.7	3.9	1.4	Iris-versicolor
83	6.0	2.7	5.1	1.6	Iris-versicolor
113	5.7	2.5	5.0	2.0	Iris-virginica

In [17]:

```
data.columns  #Displaying the number of columns and names of the columns
```

Out[17]:

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [18]:

```
data.shape #Displaying number of rows and no of columns i the data set. #The first
```

Out[18]:

```
(150, 5)
```

In [19]:

```
print(data[10:21]) # it will print the rows from 10 to 20
```

```
    sepal_length  sepal_width  petal_length  petal_width      species
10           5.4          3.7           1.5          0.2  Iris-setosa
11           4.8          3.4           1.6          0.2  Iris-setosa
12           4.8          3.0           1.4          0.1  Iris-setosa
13           4.3          3.0           1.1          0.1  Iris-setosa
14           5.8          4.0           1.2          0.2  Iris-setosa
15           5.7          4.4           1.5          0.4  Iris-setosa
16           5.4          3.9           1.3          0.4  Iris-setosa
17           5.1          3.5           1.4          0.3  Iris-setosa
18           5.7          3.8           1.7          0.3  Iris-setosa
19           5.1          3.8           1.5          0.3  Iris-setosa
20           5.4          3.4           1.7          0.2  Iris-setosa
```

In [82]:

```
specific_data = data[["sepal_length","species"]]
print(specific_data)#data[["column_name1","column_name2","column_name3"]]
```

```
     sepal_length  species
0             5.1        0
1             4.9        0
2             4.7        0
3             4.6        0
4             5.0        0
..            ...      ...
145           6.7        2
146           6.3        2
147           6.5        2
148           6.2        2
149           5.9        2

[150 rows x 2 columns]
```

In [83]:

```
print(specific_data.head(5))
```

```
   sepal_length  species
0           5.1        0
1           4.9        0
2           4.7        0
3           4.6        0
4           5.0        0
```

In [26]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
```

```
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [27]:

```
data.iloc[5] # Filtering: Displaying the specific rows using “iloc” and “loc” functi
# The “loc” functions use the index name of the row to display
# the particular row of the dataset.
# The “iloc” functions use the index integer of the row,
# which gives complete information about the row.
#loc[] is used to select rows and columns by Names/Labels
#iloc[] is used to select rows and columns by Integer Index/Position. zero ␣
#↪based index position.
```

Out[27]:

```
sepal_length            5.4
sepal_width             3.9
petal_length            1.7
petal_width             0.4
species         Iris-setosa
Name: 5, dtype: object
```

In [28]:

```
data.iloc[4]
```

Out[28]:

```
sepal_length            5.0
sepal_width             3.6
petal_length            1.4
petal_width             0.2
species         Iris-setosa
Name: 4, dtype: object
```

In [29]:

```
data.describe()
```

Out[29]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [30]:

```
 #Data Formatting: Ensuring all data formats are correct (e.g. object, text, floatin
data.isnull() #if there is data is missing, it will display True else False.
```

Out[30]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False

	sepal_length	sepal_width	petal_length	petal_width	species
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

In [31]:

```
data.isnull().sum() #isnull() function is also used to get the count of missing val
```

Out[31]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [59]:

```
data.groupby(["sepal_length"])['sepal_width'].apply(lambda x: x.isnull().sum())
   #In order to get the count of missing values of the particular column by group i
   #pandas we will be using isnull() and sum() function with apply() and groupby()
   #which performs the group wise count of missing values as shown below.
```

Out[59]:

```
sepal_length
4.3    0
4.4    0
4.5    0
4.6    0
4.7    0
4.8    0
4.9    0
5.0    0
5.1    0
5.2    0
5.3    0
5.4    0
5.5    0
5.6    0
5.7    0
5.8    0
5.9    0
6.0    0
6.1    0
6.2    0
6.3    0
6.4    0
6.5    0
6.6    0
6.7    0
6.8    0
6.9    0
7.0    0
7.1    0
7.2    0
7.3    0
7.4    0
7.6    0
7.7    0
7.9    0
Name: sepal_width, dtype: int64
```

In [35]:
```
data.dtypes
```

Out[35]:
```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

In [60]:
```
 data['petal_length']= data['petal_length'].astype("int")
```

In [61]:
```
data.dtypes #To check the data #type
```

Out[61]:
```
sepal_length    float64
sepal_width     float64
petal_length      int32
petal_width     float64
species           int32
dtype: object
```

In [62]:
```
 data['sepal_width'] = data['sepal_width'].astype("int") #To change the datatype (da
```

In [63]:
```
data.dtypes
```

Out[63]:
```
sepal_length    float64
sepal_width       int32
petal_length      int32
petal_width     float64
species           int32
dtype: object
```

In [42]:
```
 ! pip install sklearn
```

```
Collecting sklearn
  Downloading sklearn-0.0.post1.tar.gz (3.6 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py): started
  Building wheel for sklearn (setup.py): finished with status 'done'
  Created wheel for sklearn: filename=sklearn-0.0.post1-py3-none-any.whl size=2936 s
ha256=7537cba627871bed07f38f6bc9bfe59503cc282d616f4a6f4e0f6e8b8ca887a9
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\f8\e0\3d\9d0c20
20c44a519b9f02ab4fa6d2a4a996c98d79ab2f569fa1
Successfully built sklearn
Installing collected packages: sklearn
Successfully installed sklearn-0.0.post1
```

In [44]:
```
from sklearn import preprocessing
```

In [45]:
```
 min_max_scaler = preprocessing.MinMaxScaler()
```

In [46]:
```
 x=data.iloc[:,:4]
```

In [47]:
```
 x_scaled = min_max_scaler.fit_transform(x)
```

In [48]:
```
 df_normalized = pd.DataFrame(x_scaled)
```

In [49]:
```
 df_normalized
```

Out[49]:

	0	1	2	3
0	0.222222	0.625000	0.067797	0.041667

	0	1	2	3
1	0.166667	0.416667	0.067797	0.041667
2	0.111111	0.500000	0.050847	0.041667
3	0.083333	0.458333	0.084746	0.041667
4	0.194444	0.666667	0.067797	0.041667
...	...	...	...	...
145	0.666667	0.416667	0.711864	0.916667
146	0.555556	0.208333	0.677966	0.750000
147	0.611111	0.416667	0.711864	0.791667
148	0.527778	0.583333	0.745763	0.916667
149	0.444444	0.416667	0.694915	0.708333

150 rows × 4 columns

In [50]:
```
 #Label Encoding on iris dataset: For iris dataset the target column which is Specie
#Label Encoding: Label Encoding refers to converting the labels into a numeric form
```

In [52]:
```
data['species'].unique()
```

Out[52]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [53]:
```
 from sklearn import preprocessing
```

In [54]:
```
 label_encoder = preprocessing.LabelEncoder()
```

In [56]:
```
data['species']= label_encoder.fit_transform(data['species'])
```

In [58]:
```
data['species'].unique()
```

Out[58]:
```
array([0, 1, 2])
```