In [9]:

```python
 #for manipulation
import numpy as np
import pandas as pd

#for data visualization
import matplotlib.pyplot as plt
import seaborn as sns

#for implement ML
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score

#for interacitivity
from ipywidgets import interact
```

In [10]:

```python
data = pd.read_csv('IrisDATA.csv')
```

In [13]:

```python
print("Shape of data is: ", data.shape)
```

```
Shape of data is:  (150, 5)
```

In [14]:

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

Out[14]:

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

In [15]:

```python
data.isnull().sum()
```

Out[15]:

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

In [21]:

```python
data.value_counts()
```

Out[21]:

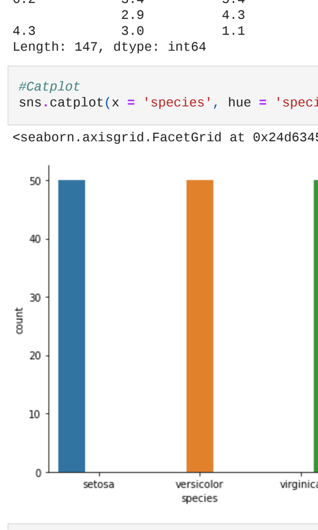
```
sepal_length  sepal_width  petal_length  petal_width  species
4.9           3.1          1.5           0.1          setosa        3
5.8           2.7          5.1           1.9          virginica     2
5.4           3.4          1.7           0.2          setosa        1
5.5           2.5          4.0           1.3          versicolor    1
              2.4          3.8           1.1          versicolor    1
                                                                   ..
6.3           2.5          4.9           1.5          versicolor    1
              2.3          4.4           1.3          versicolor    1
6.2           3.4          5.4           2.3          virginica     1
              2.9          4.3           1.3          versicolor    1
4.3           3.0          1.1           0.1          setosa        1
Length: 147, dtype: int64
```

In [24]:

```python
#Catplot
sns.catplot(x = 'species', hue = 'species', kind = 'count', data = data)
```

Out[24]:

```
<seaborn.axisgrid.FacetGrid at 0x24d634535f8>
```



In [26]:

```python
#Paired Plot
sns.set()
sns.pairplot(data[['sepal_length','sepal_width','petal_length','petal_width','species']], hue = "species", diag_ki
```

Out[26]:

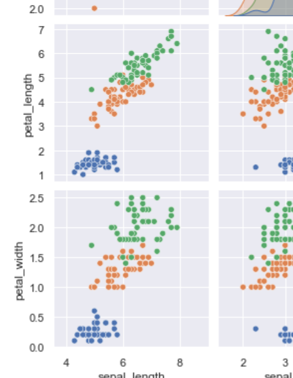
```
<seaborn.axisgrid.PairGrid at 0x24d65638ba8>
```

In [30]:

```python
#Bar plot for Species Vs Petal Width
plt.bar(data['species'],data['petal_width'])
```

Out[30]:

```
<BarContainer object of 150 artists>
```



In [31]:

```python
#Data Processing
data.describe()
```

Out[31]:

	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 [32]:

```python
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 [33]:

```python
data
```

Out[33]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [41]:

```python
Label_Encode = LabelEncoder()
Y = data['species']
Y = Label_Encode.fit_transform(Y)
```

In [42]:

```python
Y
```

Out[42]:

```
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 [44]:

```python
data['species'].nunique()
```

Out[44]:

```
3
```

In [45]:

```python
X = np.array(X)
```

In [46]:

```python
X
```

Out[46]:

```
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, 0.1],
       [5. , 3.2, 1.2, 0.2],
       [5.5, 3.5, 1.3, 0.2],
       [4.9, 3.1, 1.5, 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 [49]:

```python
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state=0)
```

In [50]:

```python
X_train
```

Out[50]:

```
array([[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, 0.1],
       [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]])
```

In [51]:

```python
X_train.shape
```

Out[51]:

```
(105, 4)
```

In [52]:

```python
X_test.shape
```

Out[52]:

```
(45, 4)
```

In [53]:

```python
Y_test.shape
```

Out[53]:

```
(45,)
```

In [54]:

```python
Y_train.shape
```

Out[54]:

```
(105,)
```

In [55]:

```python
#training the model

from sklearn.preprocessing import StandardScaler
standard_scaler = StandardScaler().fit(X_train)
X_train_std = standard_scaler.transform(X_train)
X_test_std = standard_scaler.transform(X_test)
```

In [56]:

```python
X_train_std
```

Out[56]:

```
array([[-1.02366372, -2.37846268, -0.18295039, -0.29145882],
       [ 0.69517462, -0.10190314,  0.93066067,  0.73721938],
       [ 0.92435306,  0.58106472,  1.04202177,  1.6373128 ],
       [ 0.1222285 , -1.92315077,  0.6522579 ,  0.35146505],
       [ 0.92435306, -1.24018291,  1.09770233,  0.73721938],
       [-0.33612839, -1.24018291,  0.03977182, -0.16287405],
       [ 2.07024529, -0.10190314,  1.26474398,  1.38014325],
       [ 0.46599617,  0.58106472,  0.48521625,  0.48004983],
       [-0.45071761, -1.46783886, -0.01590873, -0.16287405],
       [ 0.46599617, -0.784871  ,  0.59657735,  0.73721938],
       [ 0.46599617, -0.55721505,  0.70793846,  0.35146505],
       [-1.13825295, -1.24018291,  0.37385514,  0.6086346 ],
       [ 0.46599617, -1.24018291,  0.6522579 ,  0.86580415],
       [ 1.26812073,  0.35340877,  0.48521625,  0.22288028],
       [ 0.69517462, -0.10190314,  0.76361901,  0.99438893],
       [ 0.1222285 ,  0.80872067,  0.37385514,  0.48004983],
       [-1.25284217,  0.12575281, -1.24088089, -1.32013702],
       [-0.10694994, -0.784871  ,  0.70793846,  0.86580415],
       [-0.33612839, -0.784871  ,  0.20681348,  0.0942955 ],
       [-0.33612839, -0.32955909, -0.12726983,  0.0942955 ],
       [-0.45071761, -1.24018291,  0.09545238,  0.0942955 ],
       [ 0.23681773, -0.10190314,  0.42953569,  0.22288028],
       [ 1.49729918,  0.35340877,  1.20906343,  0.73721938],
       [-0.67989605,  1.49168853, -1.29656144, -1.32013702],
       [-1.82578828, -0.10190314, -1.51928365, -1.4487218 ],
       [ 0.5805854 , -0.784871  ,  0.81929956,  0.86580415],
       [-0.22153916, -0.10190314,  0.20681348, -0.03428927],
       [-0.56530683,  0.80872067, -1.18520034, -1.32013702],
       [-0.22153916,  3.08528021, -1.29656144, -1.06296747],
       [ 1.15353151,  0.12575281,  0.59657735,  0.35146505],
       [-1.48202061,  0.12575281, -1.29656144, -1.32013702],
       [ 0.00763928, -0.10190314,  0.70793846,  0.73721938],
       [-0.9090745 , -1.24018291, -0.46135315, -0.16287405],
       [-1.48202061,  0.80872067, -1.35224199, -1.19155225],
       [ 0.35140695, -1.92315077,  0.37385514,  0.35146505],
       [ 1.49729918,  1.26403258,  1.26474398,  1.6373128 ],
       [-0.22153916, -0.32955909,  0.20681348,  0.0942955 ],
       [-1.25284217, -0.10190314, -1.35224199, -1.4487218 ],
       [ 1.38270995, -0.10190314,  1.15338288,  1.1229737 ],
       [ 1.15353151,  0.35340877,  1.04202177,  1.38014325],
       [ 0.69517462, -0.10190314,  1.09770233,  1.25155848],
       [ 0.5805854 , -0.55721505,  0.98634122,  1.1229737 ],
       [-0.9090745 ,  1.71934449, -1.24088089, -1.32013702],
       [-1.25284217,  0.80872067, -1.24088089, -1.32013702],
       [ 0.69517462,  0.35340877,  0.70793846,  0.99438893],
       [ 0.92435306,  0.58106472,  1.04202177,  1.1229737 ],
       [-1.59660984, -1.69549482, -1.40792255, -1.19155225],
       [ 0.35140695,  0.80872067,  0.87498011,  1.38014325],
       [-1.13825295, -0.10190314, -1.35224199, -1.32013702],
       [-0.22153916, -1.24018291,  0.6522579 ,  0.99438893],
       [ 1.15353151,  0.12575281,  0.87498011,  1.1229737 ],
       [-1.71119906,  0.35340877, -1.40792255, -1.32013702],
       [-1.02366372,  1.26403258, -1.35224199, -1.32013702],
       [ 1.49729918, -0.10190314,  1.09770233,  0.48004983],
       [-0.9090745 ,  1.03637663, -1.35224199, -1.19155225],
       [-1.71119906, -0.10190314, -1.40792255, -1.32013702],
       [-0.56530683,  1.94700044, -1.18520034, -1.06296747],
       [-0.45071761, -1.69549482,  0.09545238,  0.0942955 ],
       [ 1.03894229,  0.35340877,  1.15338288,  1.38014325],
       [ 1.95565607, -0.10190314,  1.54314675,  1.1229737 ],
       [-0.9090745 ,  1.03637663, -1.35224199, -1.32013702],
       [-1.13825295,  0.12575281, -1.29656144, -1.4487218 ],
       [-0.79448528,  0.80872067, -1.35224199, -1.32013702],
       [-0.22153916, -0.55721505,  0.37385514,  0.0942955 ],
       [ 0.80976384, -0.10190314,  0.31817459,  0.22288028],
       [-1.02366372,  0.35340877, -1.4636031 , -1.32013702],
       [-0.9090745 ,  0.58106472, -1.18520034, -0.9343827 ],
       [ 0.5805854 , -0.32955909,  0.26249403,  0.0942955 ],
       [-0.56530683,  0.80872067, -1.29656144, -1.06296747],
       [ 2.07024529, -1.01252695,  1.71018841,  1.38014325],
       [-1.13825295, -1.46783886, -0.29431149, -0.29145882],
       [ 2.29942374,  1.71934449,  1.43178564,  0.99438893],
       [ 0.92435306,  0.12575281,  0.31817459,  0.22288028],
       [-0.79448528,  2.40231235, -1.29656144, -1.4487218 ],
       [ 0.1222285 , -0.10190314,  0.5408968 ,  0.73721938],
       [-0.10694994,  2.17465639, -1.4636031 , -1.32013702],
       [ 2.07024529, -0.55721505,  1.5988273 ,  0.99438893],
       [-0.9090745 ,  1.71934449, -1.29656144, -1.19155225],
       [-1.36743139,  0.35340877, -1.24088089, -1.32013702],
       [ 1.72647762, -0.55721505,  1.26474398,  0.86580415],
       [-1.02366372,  0.58106472, -1.35224199, -1.32013702],
       [ 0.46599617,  0.80872067,  0.98634122,  1.50872803],
       [-0.22153916, -0.55721505,  0.15113293,  0.0942955 ],
       [-0.10694994, -0.784871  ,  0.03977182, -0.03428927],
       [-0.22153916, -1.01252695, -0.18295039, -0.29145882],
       [ 0.5805854 ,  0.35340877,  0.81929956,  1.38014325],
       [ 0.92435306, -0.10190314,  0.76361901,  1.38014325],
       [ 0.46599617, -1.24018291,  0.59657735,  0.35146505],
       [ 0.92435306, -0.10190314,  0.6522579 ,  0.6086346 ],
       [-1.02366372, -0.10190314, -1.24088089, -1.32013702],
       [-0.45071761, -1.46783886, -0.07158928, -0.29145882],
       [ 0.92435306,  0.12575281,  0.98634122,  1.50872803],
       [-0.10694994, -0.784871  ,  0.70793846,  0.86580415],
       [-0.9090745 ,  0.80872067, -1.29656144, -1.32013702],
       [ 0.80976384, -0.32955909,  0.42953569,  0.0942955 ],
       [-0.33612839, -0.10190314,  0.15113293,  0.0942955 ],
       [ 0.00763928,  0.35340877,  0.5408968 ,  0.73721938],
       [ 0.46599617, -1.69549482,  0.31817459,  0.0942955 ],
       [-0.45071761,  1.03637663, -1.40792255, -1.32013702],
       [-0.9090745 ,  1.49168853, -1.29656144, -1.06296747],
       [-1.13825295,  0.12575281, -1.29656144, -1.4487218 ],
       [ 0.46599617, -0.32955909,  0.98634122,  0.73721938],
       [-0.10694994, -0.784871  ,  0.15113293, -0.29145882],
       [ 2.07024529,  1.71934449,  1.5988273 ,  1.25155848],
       [-1.48202061,  0.35340877, -1.35224199, -1.32013702]])
```

In [57]:

```python
Y_train
```

Out[57]:

```
array([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])
```

In [61]:

```python
#KNN Algo
```

In [58]:

```python
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)

knn.fit(X_train_std,Y_train)
```

Out[58]:

```
KNeighborsClassifier()
```

In [59]:

```python
predict_knn=knn.predict(X_test_std)
accuracy_knn=accuracy_score(Y_test,predict_knn)*100
```

In [60]:

```python
accuracy_knn
```

Out[60]:

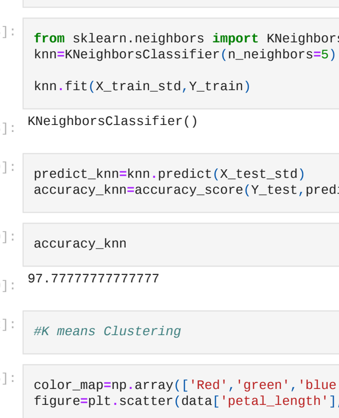
```
97.77777777777777
```

In [62]:

```python
#K means Clustering
```

In [65]:

```python
color_map=np.array(['Red','green','blue'])
figure=plt.scatter(data['petal_length'],data['petal_width'],c=color_map[Y],s=30)
```



In [66]:

```python
from sklearn.cluster import KMeans
k_means =KMeans(n_clusters=3,random_state=2, n_jobs=4)
k_means.fit(X)
```

```
c:\users\aakas\appdata\local\programs\python\python37\lib\site-packages\sklearn\cluster\_kmeans.py:793: FutureWarning: 'n_jobs' was deprecated in version 0.23 and will be removed i
n 1.0 (renaming of 0.25).
  " removed in 1.0 (renaming of 0.25).", FutureWarning)
```

Out[66]:

```
KMeans(n_clusters=3, n_jobs=4, random_state=2)
```

In [67]:

```python
y_k_means = k_means.fit_predict(X)
```

```
c:\users\aakas\appdata\local\programs\python\python37\lib\site-packages\sklearn\cluster\_kmeans.py:793: FutureWarning: 'n_jobs' was deprecated in version 0.23 and will be removed i
n 1.0 (renaming of 0.25).
  " removed in 1.0 (renaming of 0.25).", FutureWarning)
```

In [68]:

```python
centers = k_means.cluster_centers_
```

In [69]:

```python
centers
```

Out[69]:

```
array([[5.006     , 3.418     , 1.464     , 0.244     ],
       [5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
       [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```

In [71]:

```python
color_map=np.array(['Red','green','blue'])

labels=np.array(['Iris-setosa','Iris-virginica','Iris-versicolour'])
figure=plt.scatter(data['petal_length'],data['petal_width'],c=color_map[k_means.labels_],s=20)
```