In [1]:
```
from PIL import Image
```

In [4]:
```
irf=Image.open('irisimages.jpg')
```

In [5]:
```
print(irf)
```

```
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=367x137 at 0x26F664170A0>
```

In [6]:
```
irf
```

Out[6]:

In [8]:
```
#importing the libraries
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
```

In [9]:
```
#Load and read the iris dataset
data=pd.read_csv('IRIS.csv')
```

In [10]:
```
print(data)
```

```
     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
..            ...          ...           ...          ...             ...
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

[150 rows x 5 columns]
```

In [11]:
```
data.head()
```

Out[11]:

	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 [12]:
```
data.tail()
```

Out[12]:

	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 [13]:
```
data.shape
```

Out[13]:
```
(150, 5)
```

In [14]:
```
data.value_counts()
```

Out[14]:
```
sepal_length  sepal_width  petal_length  petal_width  species
4.9           3.1          1.5           0.1          Iris-setosa        3
5.8           2.7          5.1           1.9          Iris-virginica     2
              4.0          1.2           0.2          Iris-setosa        1
5.9           3.0          4.2           1.5          Iris-versicolor    1
6.2           3.4          5.4           2.3          Iris-virginica     1
                                                                       ..
5.5           2.3          4.0           1.3          Iris-versicolor    1
              2.4          3.7           1.0          Iris-versicolor    1
                           3.8           1.1          Iris-versicolor    1
              2.5          4.0           1.3          Iris-versicolor    1
7.9           3.8          6.4           2.0          Iris-virginica     1
Length: 147, dtype: int64
```

In [15]:
```
data.isnull()
```

Out[15]:

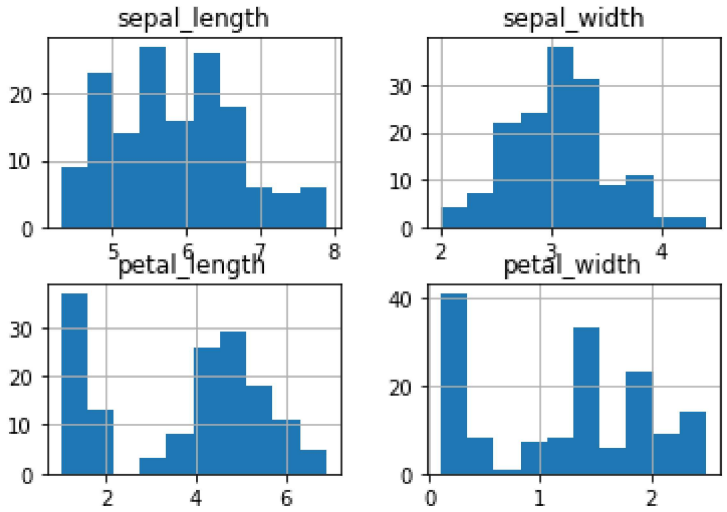
	sepal_length	sepal_width	petal_length	petal_width	species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

In [16]:
```
data.isnull().sum()
```

Out[16]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

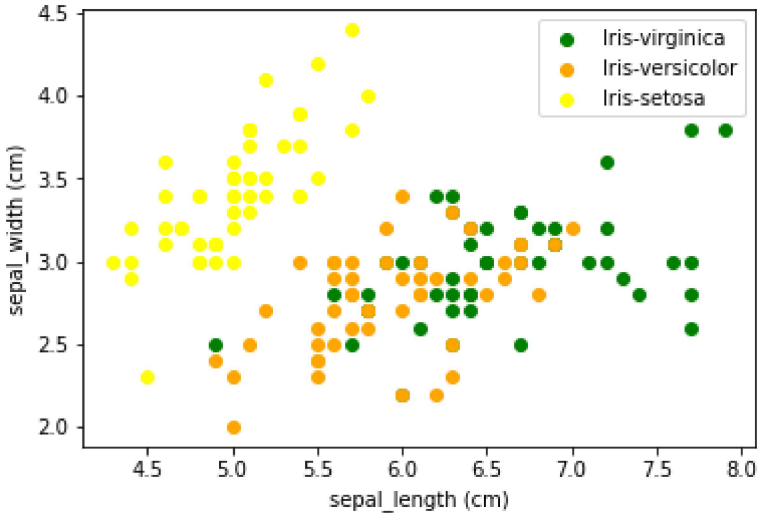
In [17]:
```
data.hist()
plt.show()
```



In [21]:
```
c=['green','orange','yellow']
f=['Iris-virginica','Iris-versicolor','Iris-setosa']
for i in range(3):
    x=data[data['species'] == f[i]]
    plt.scatter(x['sepal_length'],x['sepal_width'],c=c[i],label=f[i])
plt.xlabel('sepal_length (cm)')
```

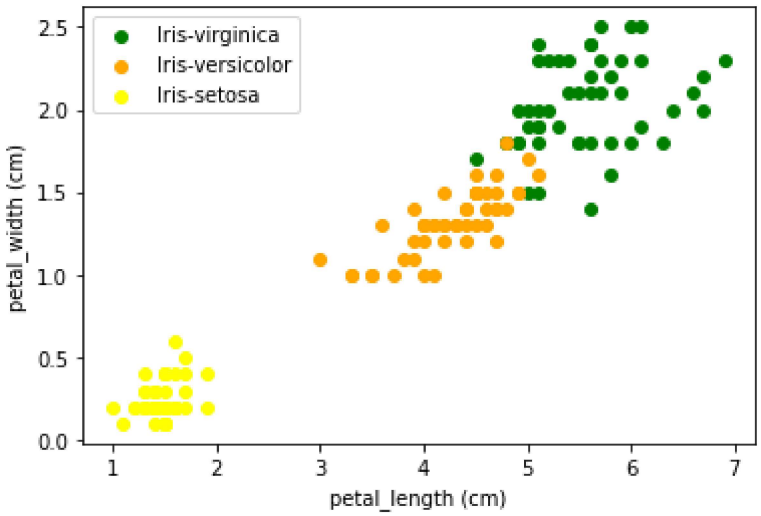
```
plt.ylabel('sepal_width (cm)')
plt.legend()
```

Out[21]: <matplotlib.legend.Legend at 0x26f6e4890a0>



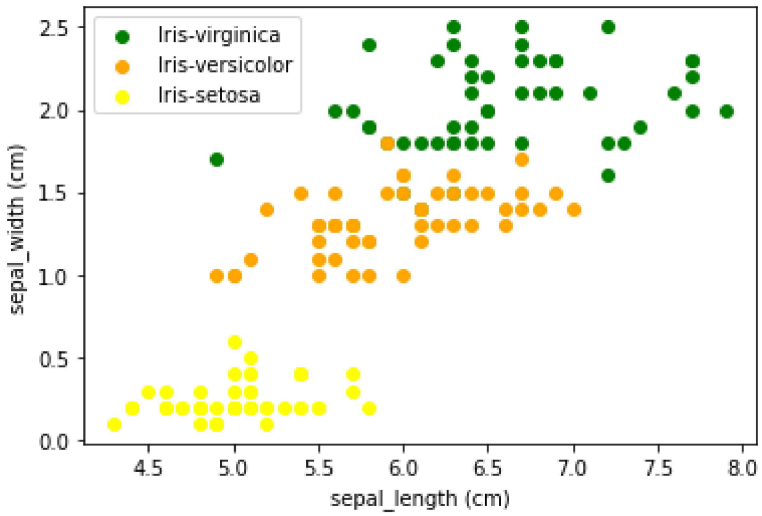
In [22]:
```
c=['green','orange','yellow']
f=['Iris-virginica','Iris-versicolor','Iris-setosa']
for i in range(3):
    x=data[data['species'] == f[i]]
    plt.scatter(x['petal_length'],x['petal_width'],c=c[i],label=f[i])
plt.xlabel('petal_length (cm)')
plt.ylabel('petal_width (cm)')
plt.legend()
```

Out[22]: <matplotlib.legend.Legend at 0x26f6e56aca0>



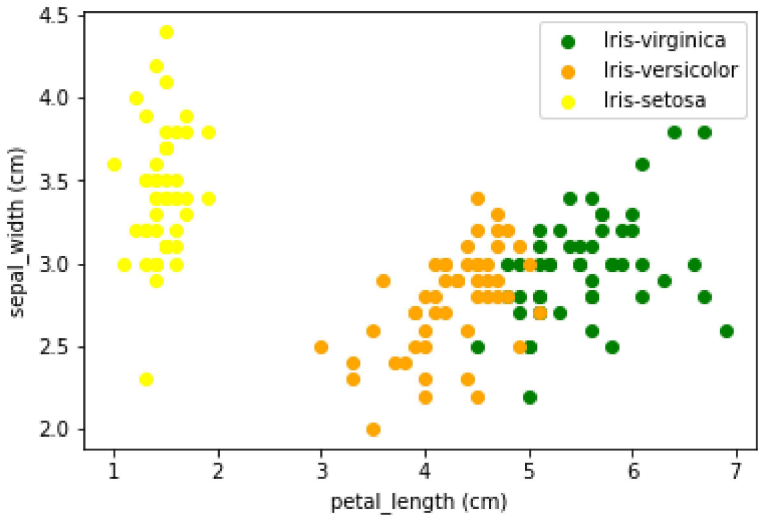
In [24]:
```
c=['green','orange','yellow']
f=['Iris-virginica','Iris-versicolor','Iris-setosa']
for i in range(3):
    x=data[data['species'] == f[i]]
    plt.scatter(x['sepal_length'],x['petal_width'],c=c[i],label=f[i])
plt.xlabel('sepal_length (cm)')
plt.ylabel('petal_width (cm)')
plt.legend()
```

Out[24]: <matplotlib.legend.Legend at 0x26f6e4f8a30>



In [25]:
```
c=['green','orange','yellow']
f=['Iris-virginica','Iris-versicolor','Iris-setosa']
for i in range(3):
    x=data[data['species'] == f[i]]
    plt.scatter(x['petal_length'],x['sepal_width'],c=c[i],label=f[i])
plt.xlabel('petal_length (cm)')
plt.ylabel('sepal_width (cm)')
plt.legend()
```

Out[25]: <matplotlib.legend.Legend at 0x26f6acf8d00>



In [26]:
```
data.corr()
```

Out[26]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

In [27]:
```
sns.heatmap(data.corr())
```

Out[27]:
```
<AxesSubplot:>
```

In [29]:
```
from sklearn.linear_model import LinearRegression,LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
```

In [31]:
```
e=LabelEncoder()
data['species']=e.fit_transform(data['species'])
```

In [32]:
```
data.head(10)
```

Out[32]:

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

In [33]:
```
data.tail()
```

Out[33]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

In [34]:
```
x=data.drop(columns=['species'])
y=data['species']
```

In [35]:
```
x
```

Out[35]:

	sepal_length	sepal_width	petal_length	petal_width
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 [36]:
```
y
```

Out[36]:
```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: species, Length: 150, dtype: int32
```

In [37]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
```

In [38]:
```
lg=LogisticRegression()
```

In [39]:
```
lg.fit(x_train,y_train)
```

Out[39]:
```
LogisticRegression()
```

In [40]:
```
s=lg.score(x_test,y_test)*100
```

In [41]:
```
print(s)
```

```
97.77777777777777
```

In [43]:
```
print("the accuracy of logistic regerssion is",s)
```

```
the accuracy of logistic regerssion is 97.77777777777777
```

In [44]:
```
y_predict=lg.predict(x_test)
```

In [45]:
```
print(y_predict)
```

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

In [46]:
```
dt=DecisionTreeClassifier()
```

In [47]:
```
dt.fit(x_train,y_train)
```

Out[47]:
```
DecisionTreeClassifier()
```

In [48]:
```
dscore=dt.score(x_test,y_test)*100
```

In [49]:
```
print(dscore)
```

```
91.11111111111111
```

In [50]:
```
print("the accuracy of logistic regerssion is",dscore)
```

```
the accuracy of logistic regerssion is 91.11111111111111
```

In [51]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_predict)
print(cm)
```

```
[[17  0  0]
 [ 0 14  1]
 [ 0  0 13]]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: