In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df= pd.read_csv("C:/Users/aryan/OneDrive/Documents/Iris.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [4]:

```
df.drop("Id",axis=1,inplace=True)
```

In [5]:

```
df["Species"].value_counts()
```

Out[5]:

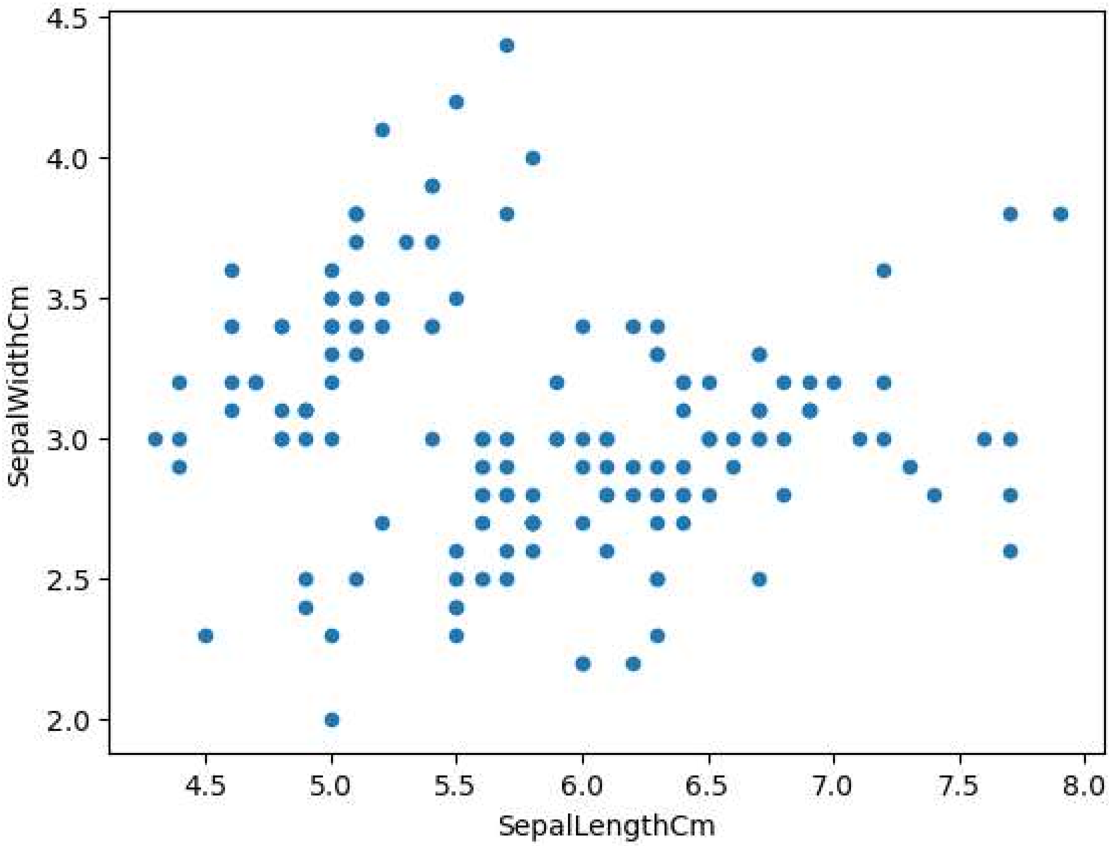
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [6]:

```
df.plot(kind="scatter", x="SepalLengthCm", y="SepalWidthCm")
```

Out[6]:

```
<AxesSubplot:xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```

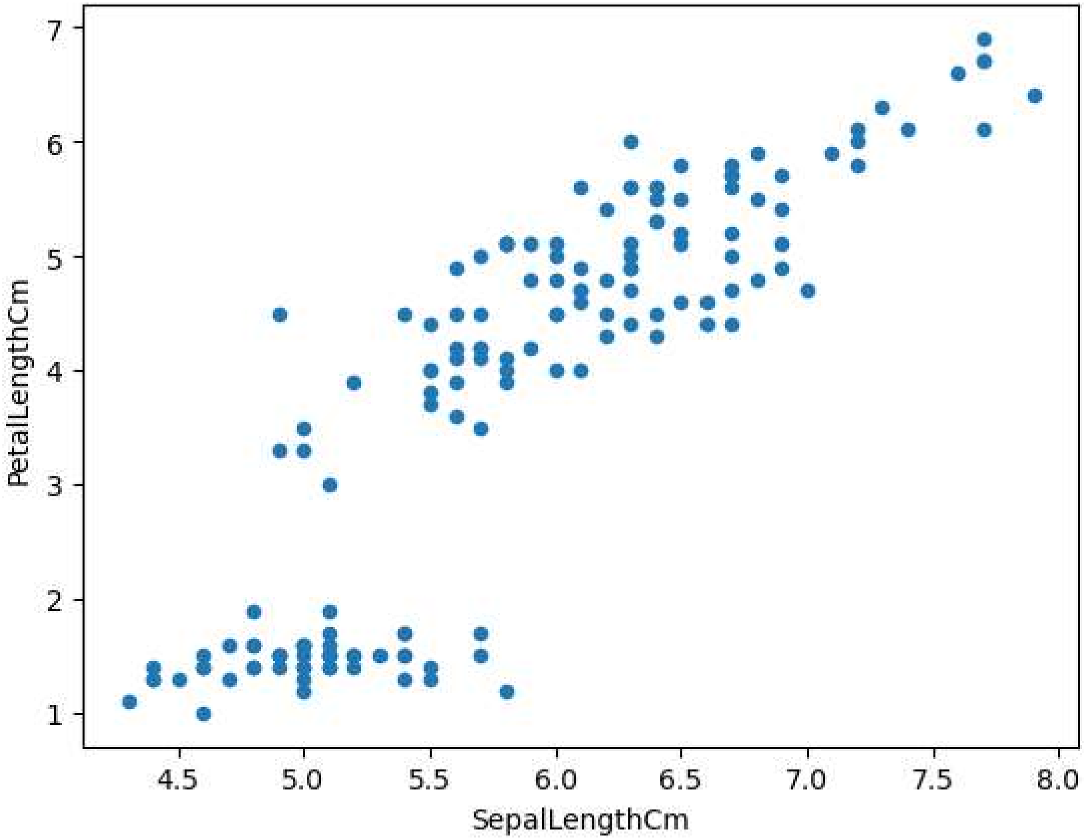


In [7]:

```
df.plot(kind="scatter", x="SepalLengthCm", y="PetalLengthCm")
```

Out[7]:

```
<AxesSubplot:xlabel='SepalLengthCm', ylabel='PetalLengthCm'>
```

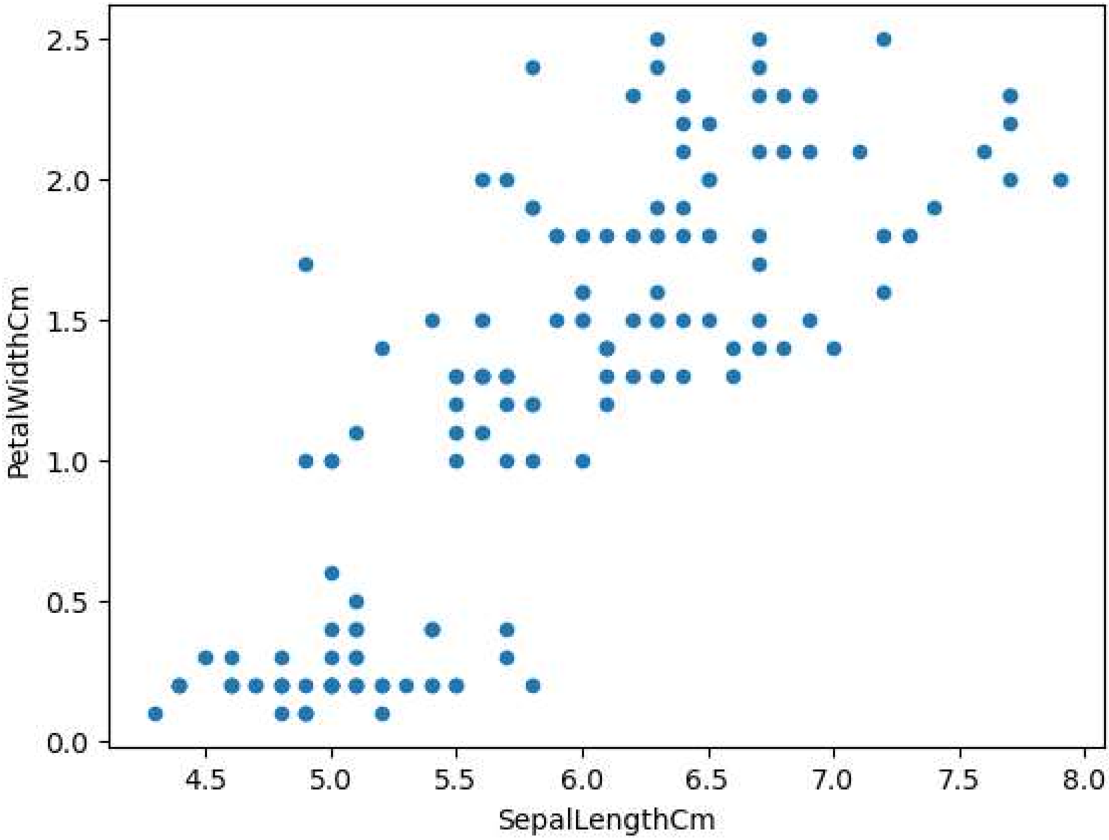


In [8]:

```
df.plot(kind="scatter", x="SepalLengthCm", y="PetalWidthCm")
```

Out[8]:

```
<AxesSubplot:xlabel='SepalLengthCm', ylabel='PetalWidthCm'>
```

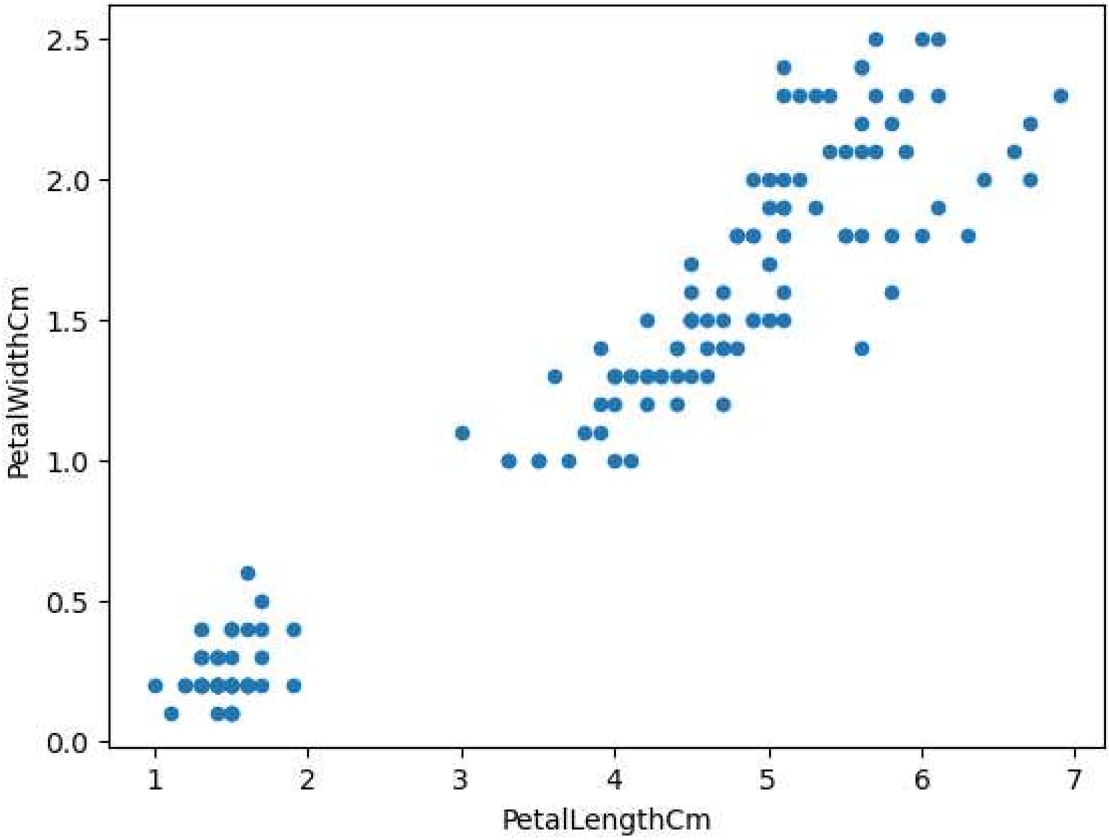


In [9]:

```
df.plot(kind="scatter", x="PetalLengthCm", y="PetalWidthCm")
```

Out[9]:

```
<AxesSubplot:xlabel='PetalLengthCm', ylabel='PetalWidthCm'>
```

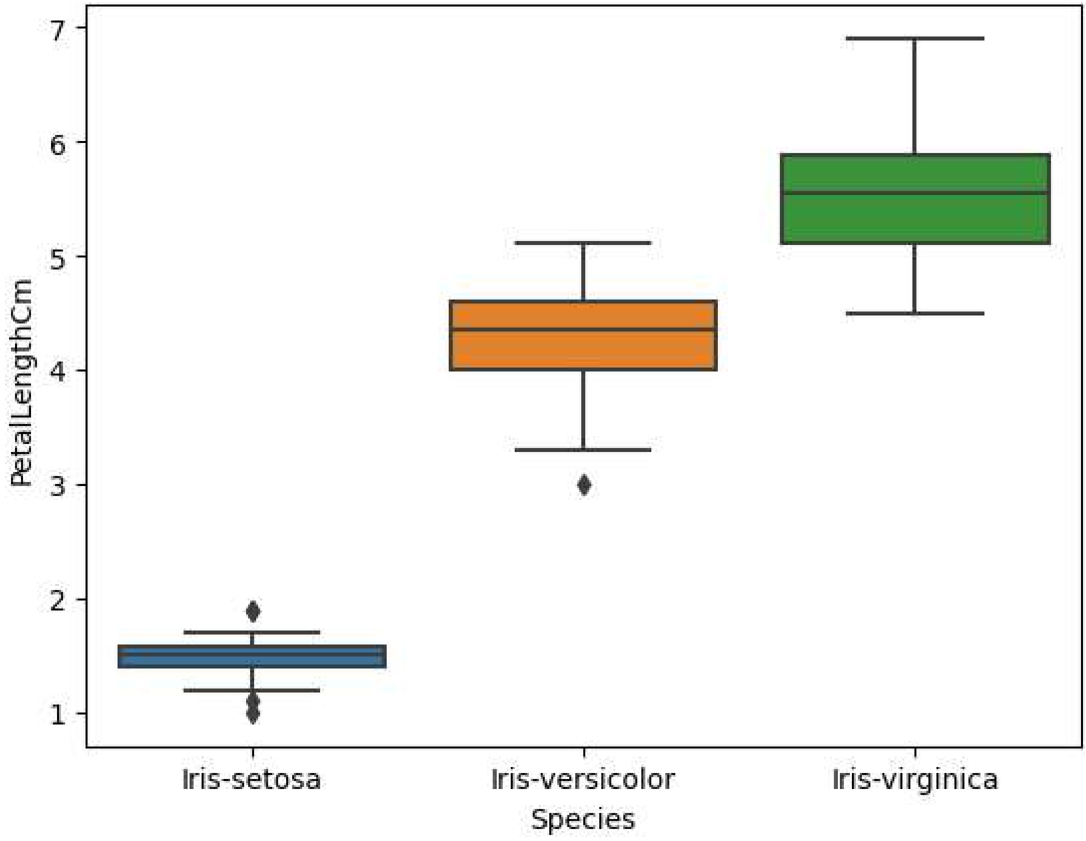


In [10]:

```
sns.boxplot(x="Species", y="PetalLengthCm", data=df)
```

Out[10]:

```
<AxesSubplot:xlabel='Species', ylabel='PetalLengthCm'>
```

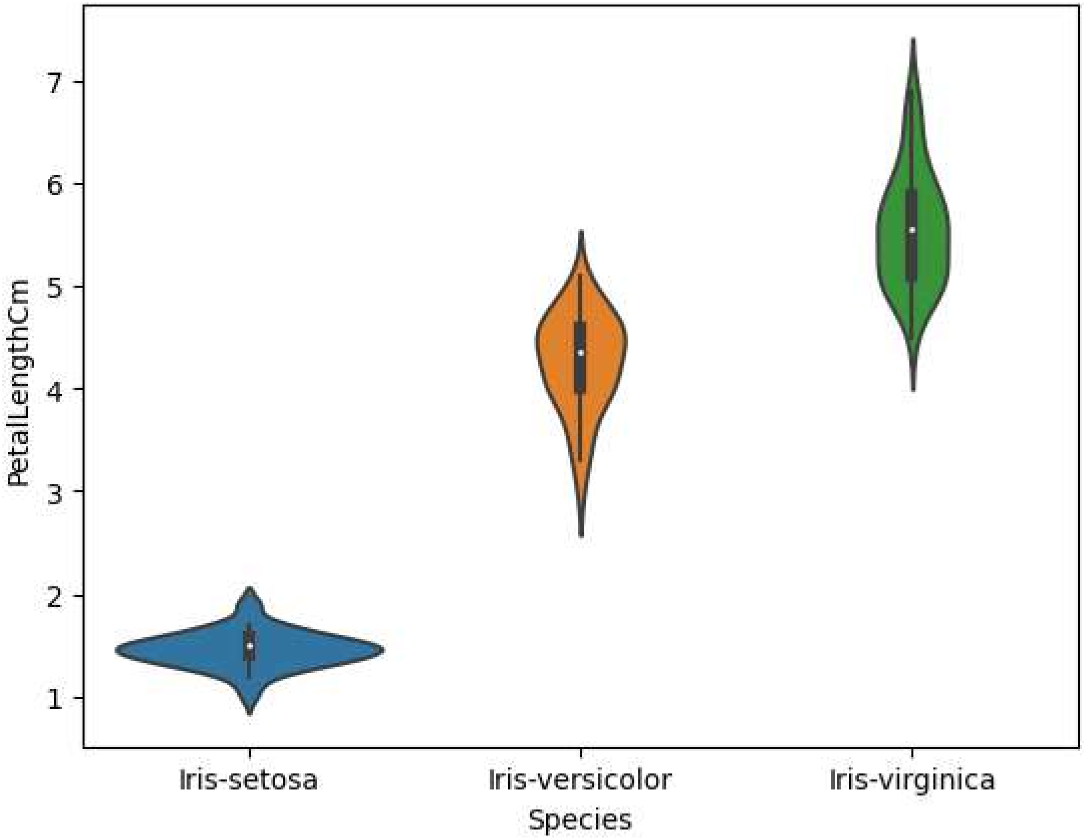


In [11]:

```
sns.violinplot(x="Species", y="PetalLengthCm", data=df)
```

Out[11]:

```
<AxesSubplot:xlabel='Species', ylabel='PetalLengthCm'>
```



In [12]:

```
df1= pd.get_dummies(df)
df1
```

Out[12]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species_Iris-setosa	Species_Ir versico
0	5.1	3.5	1.4	0.2	1	
1	4.9	3.0	1.4	0.2	1	
2	4.7	3.2	1.3	0.2	1	
3	4.6	3.1	1.5	0.2	1	
4	5.0	3.6	1.4	0.2	1	
...	...	...	...	...	...	
145	6.7	3.0	5.2	2.3	0	
146	6.3	2.5	5.0	1.9	0	
147	6.5	3.0	5.2	2.0	0	
148	6.2	3.4	5.4	2.3	0	
149	5.9	3.0	5.1	1.8	0	

150 rows × 7 columns

In [13]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [14]:

```
x=df.iloc[:,0:4]
y=df["Species"]
x.head()
```

Out[14]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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

In [15]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)

print("Train Shape",x_train.shape)
print("Test Shape",x_test.shape)
```

```
Train Shape (112, 4)
Test Shape (38, 4)
```

In [16]:

```
log = LogisticRegression()
log.fit(x_train,y_train)
prediction=log.predict(x_test)
print('The accuracy of the Logistic Regression is',metrics.accuracy_score(prediction,y_t
```

```
The accuracy of the Logistic Regression is 0.9736842105263158
```

In [17]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
```

In [18]:

```
tree=DecisionTreeClassifier()
tree.fit(x_train,y_train)
prediction=tree.predict(x_test)
print('The accuracy of the Decision Tree is',metrics.accuracy_score(prediction,y_test))
```

```
The accuracy of the Decision Tree is 0.9736842105263158
```

In [20]:

```
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train,y_train)
prediction=knn.predict(x_test)
print('The accuracy of the KNN is',metrics.accuracy_score(prediction,y_test))
```

```
The accuracy of the KNN is 0.9736842105263158

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neighbors\_classificati
on.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `
kurtosis`), the default behavior of `mode` typically preserves the axis it
acts along. In SciPy 1.11.0, this behavior will change: the default value
of `keepdims` will become False, the `axis` over which the statistic is ta
ken will be eliminated, and the value None will no longer be accepted. Set
`keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```