PROBLEM STATEMENT

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The attribute to be predicted is the class of iris plant. The classes are as follows: 1. Iris Setosa, 2. Iris Versicolour, 3. Iris Virginica

There are 4 features:

1. sepalLength: sepal length in cm
2. sepalWidth: sepal width in cm
3. petalLength: petal length in cm
4. petalWidth: petal width in cm

There are 3 classes represneting class label of iris flower (1,2,3)

1. Iris Setosa
2. Iris Versicolour
3. Iris Virginica

IMPORTING DATA

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

In [2]:
```
iris_df=pd.read_csv(r'C:\Users\psc\Documents\Iris.csv')
```

EXPLORING THE DATASET

In [3]:
```
iris_df.head()
```

Out[3]:

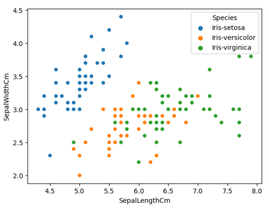
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [4]:
```
iris_df.shape
```

Out[4]:
```
(150, 5)
```

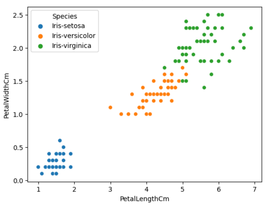
In [5]:
```
sns.scatterplot(x='SepalLengthCm' , y='SepalWidthCm', data=iris_df ,hue='Species')
```

Out[5]:
```
<AxesSubplot: xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```



In [6]:
```
sns.scatterplot(y='PetalWidthCm', x='PetalLengthCm', data=iris_df ,hue='Species')
```

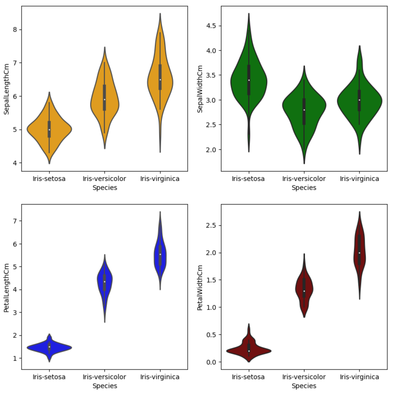
Out[6]:
```
<AxesSubplot: xlabel='PetalLengthCm', ylabel='PetalWidthCm'>
```



Let's show the Violin plot

In [7]:
```
plt.figure(figsize=(10,10))

plt.subplot(2,2,1)
sns.violinplot(x='Species', y='SepalLengthCm', data=iris_df, color='orange' )

plt.subplot(2,2,2)
sns.violinplot(x='Species', y='SepalWidthCm', data=iris_df, color='green')

plt.subplot(2,2,3)
sns.violinplot(x='Species', y='PetalLengthCm', data=iris_df, color='blue')

plt.subplot(2,2,4)
sns.violinplot(x='Species', y='PetalWidthCm', data=iris_df, color='maroon')
```

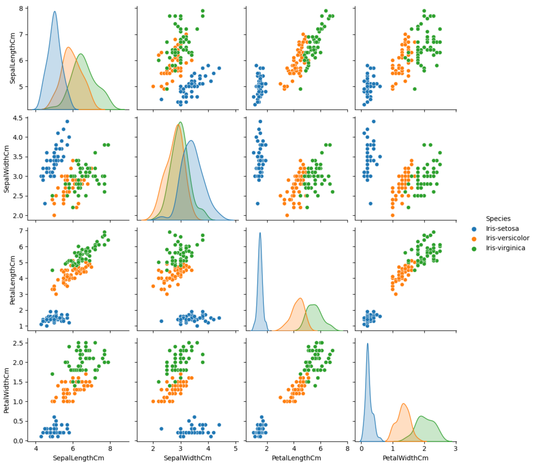
Out[7]:
```
<AxesSubplot: xlabel='Species', ylabel='PetalWidthCm'>
```



Let's try the Seaborn pairplot

In [8]:
```
sns.pairplot(iris_df,hue='Species')
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x28c73083f10>
```



TRAINING THE MODEL

In [9]:
```
x=iris_df.iloc[:,:-1]
y=iris_df.iloc[:,-1]
```

In [10]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=4)
```

Fitting K-NN to the Training set

In [12]:
```
kneighbor_classifier=KNeighborsClassifier(n_neighbors=5)
kneighbor_classifier.fit(x_train,y_train)
```

Out[12]:
```
KNeighborsClassifier()
```

In [13]:
```
y_pred= kneighbor_classifier.predict(x_test)
```

In [14]:
```
acr=accuracy_score(y_pred,y_test)
acr
```

Out[14]:
```
0.9666666666666667
```

EVALUATING THE MODEL

In [17]:
```
cm=confusion_matrix(y_test,y_pred)
sns.heatmap(cm)
```

Out[17]:
```
<AxesSubplot: >
```