IRIS FLOWER CLASSIFICATION:

(Task-1)

In [1]:
```
#importing basic libraries
import pandas as pd
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 classification_report,accuracy_score
```

In [2]:
```
data_set=pd.read_csv("C:\Users\HP\OneDrive\Documents\oasis infobytes\Iris.csv")
data_set.head()
```

Out[2]:

	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 [3]:
```
data_set.tail()
```

Out[3]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
295	6.7	3.0	5.2	2.3	Iris-virginica
296	6.3	2.5	5.0	1.9	Iris-virginica
297	6.5	3.0	5.2	2.0	Iris-virginica
298	6.2	3.4	5.4	2.3	Iris-virginica
299	5.9	3.0	5.1	1.8	Iris-virginica

In [4]:
```
data_set["Species"].unique()
```

Out[4]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [ ]:

In [ ]:

checking missing value

In [5]:
```
data_set.isnull().sum()
```

Out[5]:
```
SepalLengthCm   0
SepalWidthCm    0
PetalLengthCm   0
PetalWidthCm    0
Species         0
dtype: int64
```

In [6]:
```
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SepalLengthCm  300 non-null    float64
 1   SepalWidthCm   300 non-null    float64
 2   PetalLengthCm  300 non-null    float64
 3   PetalWidthCm   300 non-null    float64
 4   Species        300 non-null    object 
dtypes: float64(4), object(1)
memory usage: 11.8+ KB
```

In [7]:
```
type(data_set) #checking the type of dataset
```

Out[7]:
```
pandas.core.frame.DataFrame
```

In [8]:
```
print(data_set.shape) #in this csv file we have 300 rows and 5 columns .
```

```
(300, 5)
```

In [9]:
```
data_set.describe()
```

Out[9]:

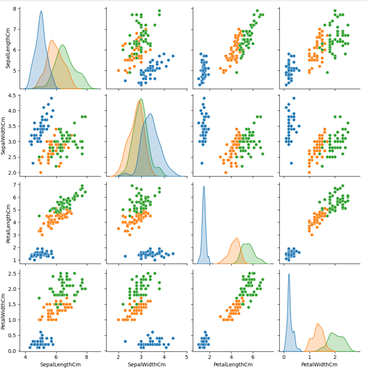
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	300.000000	300.000000	300.000000	300.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433989	1.764467	0.763863
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [ ]:

In [ ]:

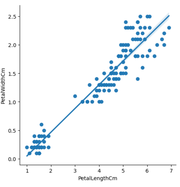
Data visualizataion

In [10]:
```
sns.pairplot(data_set,hue="Species")
plt.show()
```



In [11]:
```
sns.lmplot(x='PetalLengthCm',y='PetalWidthCm',data=data_set)
```

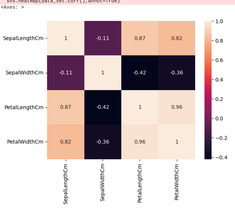
Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x144b733a9b0>
```



In [12]:
```
sns.heatmap(data_set.corr(),annot=True)
```

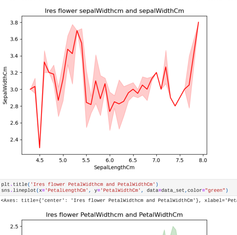
```
C:\Users\HP\AppData\Local\Temp\ipykernel_22100\3008311531.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will defau
lt to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(data_set.corr(),annot=True)
```

Out[12]:
```
<Axes: >
```



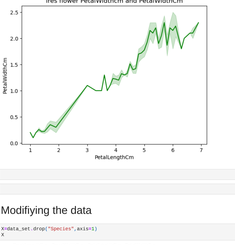
In [13]:
```
plt.title('Ires flower sepalWidthcm and sepalWidthCm')
sns.lineplot(x='SepalLengthCm', y='SepalWidthCm', data=data_set,color="red")
```

Out[13]:
```
<Axes: title={'center': 'Ires flower sepalWidthcm and sepalWidthCm'}, xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```



In [14]:
```
plt.title('Ires flower PetalWidthcm and PetalWidthCm')
sns.lineplot(x='PetalLengthCm', y='PetalWidthCm', data=data_set,color="green")
```

Out[14]:
```
<Axes: title={'center': 'Ires flower PetalWidthcm and PetalWidthCm'}, xlabel='PetalLengthCm', ylabel='PetalWidthCm'>
```



In [ ]:

In [ ]:

Modifiying the data

In [15]:
```
X=data_set.drop("Species",axis=1)
X
```

Out[15]:

	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
...	...	...	...	...
295	6.7	3.0	5.2	2.3
296	6.3	2.5	5.0	1.9
297	6.5	3.0	5.2	2.0
298	6.2	3.4	5.4	2.3
299	5.9	3.0	5.1	1.8

300 rows × 4 columns

In [16]:
```
Y=data_set["Species"]
Y
```

Out[16]:
```
0          Iris-setosa
1          Iris-setosa
2          Iris-setosa
3          Iris-setosa
4          Iris-setosa
             ...      
295     Iris-virginica
296     Iris-virginica
297     Iris-virginica
298     Iris-virginica
299     Iris-virginica
Name: Species, Length: 300, dtype: object
```

In [ ]:

In [ ]:

Training the model

In [17]:
```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3,random_state=42)
```

In [18]:
```
knn=KNeighborsClassifier(n_neighbors=3) #KNeighbourClassification Algorithe.
knn.fit(x_train, y_train)
```

Out[18]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [19]:
```
y_predict=knn.predict(x_test)
```

In [20]:
```
print("Accuracy:" ,accuracy_score(y_test,y_predict))
```

```
Accuracy: 0.9555555555555556
```

In [ ]:

In [ ]:

Finally created model prediction (or) Final Result.

In [21]:
```
import pandas as pd
new_dataset=pd.DataFrame({'SepalLengthCm':2.3,"SepalWidthCm":3.0,"PetalLengthCm":4.2,"PetalWidthCm":7.2}, index=[0])
predictionknn.predict(new_dataset)
prediction[0]
```

Out[21]:
```
'Iris-virginica'
```

In [ ]:

In [22]:
```
                                    #Thanking you...
```


- Thiruvalluvan G