Iris Flower Classification

In [2]:
```
#import dataset
import pandas as pd
import numpy as np
a=pd.read_csv("Iris.csv")
a
```

Out[2]:

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [3]:
```
a.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]:
```
b=a["Species"].unique()
b
```

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

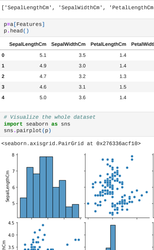
In [5]:
```
a.describe()
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

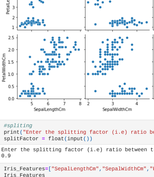
In [6]:
```
# Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(data=a,x='SepalLengthCm',hue='PetalWidthCm')
```

Out[6]:
```
<AxesSubplot:xlabel='SepalLengthCm', ylabel='count'>
```



In [7]:
```
import matplotlib.pyplot as plt
x=a["PetalLengthCm"].value_counts()
plt.plot(x)
```

Out[7]:
```
[<matplotlib.lines.Line2D at 0x27638ea88b0>]
```



In [8]:
```
Features=["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm","Species"]
Features
```

Out[8]:
```
['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm', 'Species']
```

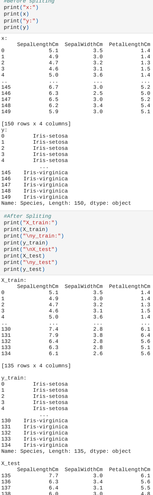
In [9]:
```
p=a[Features]
p.head()
```

Out[9]:

	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 [10]:
```
# Visualize the whole dataset
import seaborn as sns
sns.pairplot(p)
```

Out[10]:
```
<seaborn.axisgrid.PairGrid at 0x27633dacf10>
```



In [11]:
```
#splitting
print("Enter the splitting factor (i.e) ratio between train and test")
splitFactor = float(input())
```

```
Enter the splitting factor (i.e) ratio between train and test
0.9
```

In [12]:
```
Iris_Features=["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"]
Iris_Features
```

Out[12]:
```
['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']
```

In [13]:
```
x=a[Iris_Features]
x.head()
```

Out[13]:

	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 [14]:
```
y=a.Species
y.head()
```

Out[14]:
```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

In [15]:
```
import math
n_train = math.floor(splitFactor * x.shape[0])
n_test = math.ceil((1-splitFactor) * x.shape[0])
X_train = x[:n_train]
y_train = y[:n_train]
X_test = x[n_train:]
y_test = y[n_train:]
print("Total Number of rows in train:",X_train.shape[0])
print("Total Number of rows in test:",X_test.shape[0])
```

```
Total Number of rows in train: 135
Total Number of rows in test: 15
```

In [16]:
```
#before splitting
print("x:")
print(x)
print("y:")
print(y)
```

```
x:
     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
..             ...           ...            ...           ...
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 x 4 columns]
y:
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

In [17]:
```
#After Spliting
print("X_train:")
print(X_train)
print("\ny_train:")
print(y_train)
print("\nX_test:")
print(X_test)
print("\ny_test")
print(y_test)
```

```
X_train:
     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
..             ...           ...            ...           ...
130            7.4           2.8            6.1           1.9
131            7.9           3.8            6.4           2.0
132            6.4           2.8            5.6           2.2
133            6.3           2.8            5.1           1.5
134            6.1           2.6            5.6           1.4

[135 rows x 4 columns]

y_train:
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
130    Iris-virginica
131    Iris-virginica
132    Iris-virginica
133    Iris-virginica
134    Iris-virginica
Name: Species, Length: 135, dtype: object

X_test
     SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
135            7.7           3.0            6.1           2.3
136            6.3           3.4            5.6           2.4
137            6.4           3.1            5.5           1.8
138            6.0           3.0            4.8           1.8
139            6.9           3.1            5.4           2.1
140            6.7           3.1            5.6           2.4
141            6.9           3.1            5.1           2.3
142            5.8           2.7            5.1           1.9
143            6.8           3.2            5.9           2.3
144            6.7           3.3            5.7           2.5
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

y_test
135    Iris-virginica
136    Iris-virginica
137    Iris-virginica
138    Iris-virginica
139    Iris-virginica
140    Iris-virginica
141    Iris-virginica
142    Iris-virginica
143    Iris-virginica
144    Iris-virginica
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, dtype: object
```

Decision Tree

In [18]:
```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
model =tree. DecisionTreeClassifier()
```

In [19]:
```
tree= model.fit(X_train,y_train)
```

In [20]:
```
print("Accuracy on training set:{:.3f}".format(tree.score(X_train,y_train)))
print("Accuracy on test set:{:.3f}".format(tree.score(X_test,y_test)))
```

```
Accuracy on training set:1.000
Accuracy on test set:1.000
```

In [21]:
```
y_pred=tree.predict(X_test)
print("Test set prediction:\n{}".format(y_pred))
```

```
Test set prediction:
['Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica']
```

In [22]:
```
print("Test set score:{:.2f}".format(np.mean(y_pred==y_test)))
print("Test set score:{:.2f}".format(tree.score(X_test,y_test)))
```

```
Test set score:1.00
Test set score:1.00
```

In [23]:
```
features=np.array([[4.6,3.4,1.4,0.3]])
features
```

Out[23]:
```
array([[4.6, 3.4, 1.4, 0.3]])
```

In [24]:
```
prediction=tree.predict(features)
y_pred=tree.predict(features)
```

In [25]:
```
#Prediction trail-1
print("Test set predictions:\n{}".format(y_pred))
```

```
Test set predictions:
['Iris-setosa']
```