Support Vector Machine

Importing Important Libraries

In [53]:

```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

Importing Dataset from sklearn Library

In [54]:

```
from sklearn.datasets import load_iris
```

In [55]:

```
iris = load_iris()
```

Column's in Iris Dataset

In [56]:

```
dir(iris)
```

Out[56]:

```
['DESCR',
 'data',
 'data_module',
 'feature_names',
 'filename',
 'frame',
 'target',
 'target_names']
```

Feature's in Iris Dataset

In [57]:

```
iris.feature_names
```

Out[57]:

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

Creating a DataFrame from Data

In [58]:

```
df = pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

Out[58]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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

Adding Target Column to the DataFrame

In [59]:

```
df["target"] = iris.target
df.head()
```

Out[59]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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

Exploring the Target Set

In [60]:

```
iris.target_names
```

Out[60]:

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [80]:

```
df[df.target==0].head()
```

Out[80]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa

In [81]:

```
df[df.target==1].head()
```

Out[81]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
50	7.0	3.2	4.7	1.4	1	versicolor
51	6.4	3.2	4.5	1.5	1	versicolor
52	6.9	3.1	4.9	1.5	1	versicolor
53	5.5	2.3	4.0	1.3	1	versicolor
54	6.5	2.8	4.6	1.5	1	versicolor

In [82]:

```
df[df.target==2].head()
```

Out[82]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
100	6.3	3.3	6.0	2.5	2	virginica
101	5.8	2.7	5.1	1.9	2	virginica
102	7.1	3.0	5.9	2.1	2	virginica
103	6.3	2.9	5.6	1.8	2	virginica
104	6.5	3.0	5.8	2.2	2	virginica

Adding Target Name to the Data Frame

In [64]:

```
df["flower_name"] = df.target.apply(lambda x:iris.target_names[x])
df.head()
```

Out[64]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa

Visualization of Data

Creating 3 Different DataFrame from Orignal DataFrame

In [65]:

```
df0 = df[df.target==0]
df1 = df[df.target==1]
df2 = df[df.target==2]
```

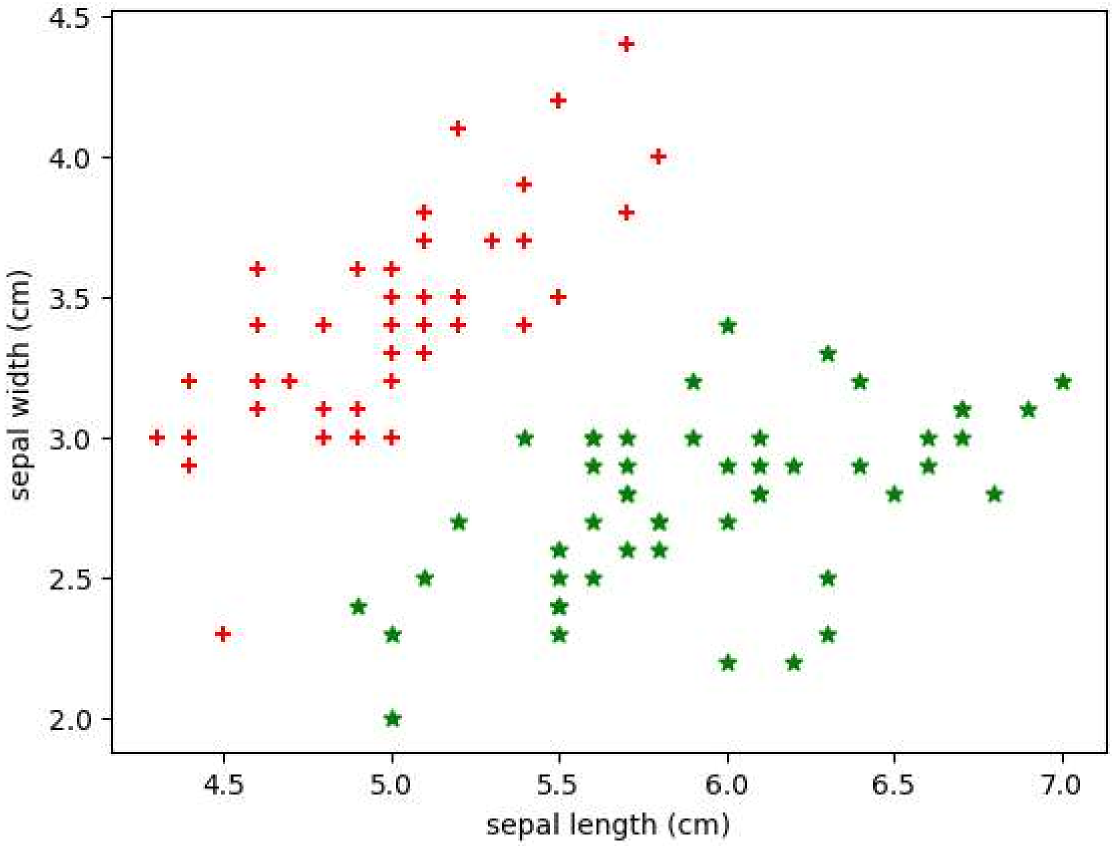
Creating Scatter Plot with "sepal length (cm)", "sepal width (cm)" as X and Y Axis

In [66]:

```
plt.scatter(df0["sepal length (cm)"],df0["sepal width (cm)"],color="red",marker="+")
plt.scatter(df1["sepal length (cm)"],df1["sepal width (cm)"],color="green",marker="*")
plt.xlabel("sepal length (cm)")
plt.ylabel("sepal width (cm)")
```

Out[66]:

```
Text(0, 0.5, 'sepal width (cm)')
```



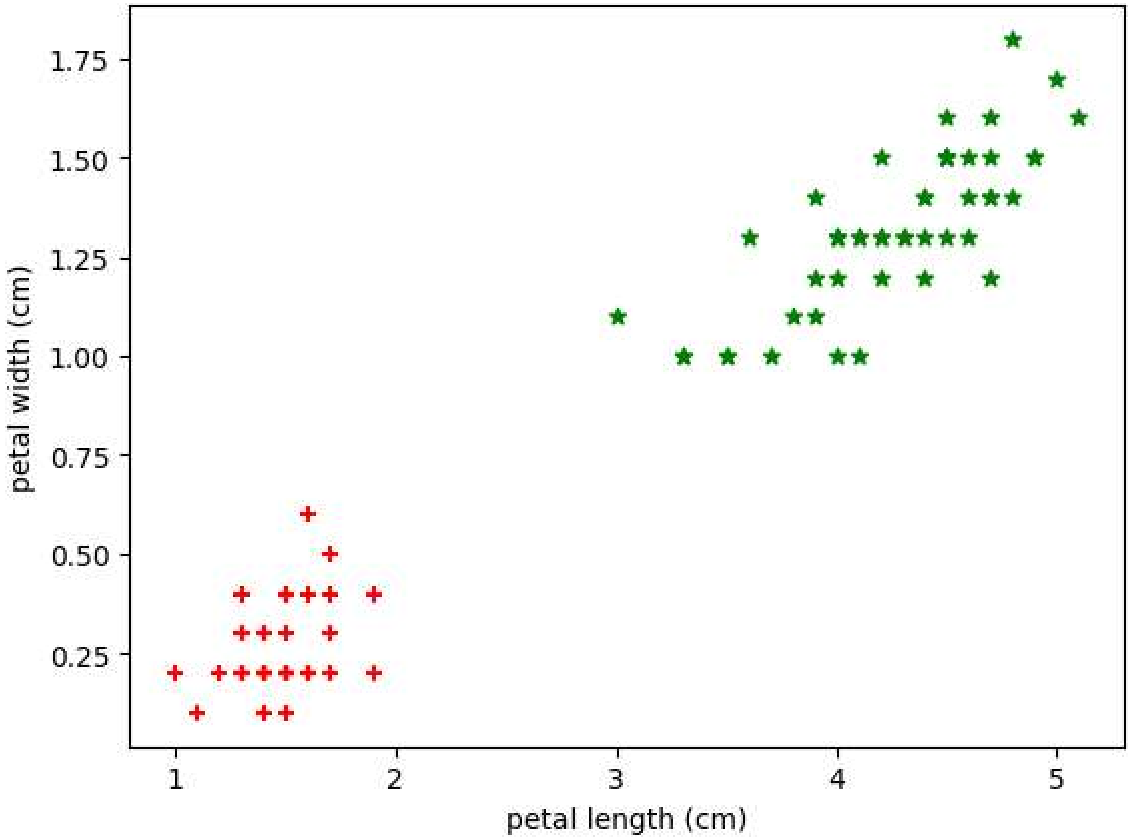
Creating Scatter Plot with "petal length (cm)", "petal width (cm)" as X and Y Axis

In [67]:

```
plt.scatter(df0["petal length (cm)"],df0["petal width (cm)"],color="red",marker="+")
plt.scatter(df1["petal length (cm)"],df1["petal width (cm)"],color="green",marker="*")
plt.xlabel("petal length (cm)")
plt.ylabel("petal width (cm)")
```

Out[67]:

```
Text(0, 0.5, 'petal width (cm)')
```



Creating Dependent and Independent Variable

In [68]:

```
X = df.drop(["target","flower_name"], axis = "columns")
X.head()
```

Out[68]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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 [69]:

```
y = df.target
y.head()
```

Out[69]:

```
0    0
1    0
2    0
3    0
4    0
Name: target, dtype: int32
```

Train Test Split

In [70]:

```
from sklearn.model_selection import train_test_split
```

In [71]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [72]:

```
len(X_train), len(X_test)
```

Out[72]:

```
(120, 30)
```

Importing SVM Model

In [73]:

```
from sklearn.svm import SVC
```

In [74]:

```
model = SVC()
```

Fitting the Model

In [75]:

```
model.fit(X_train, y_train)
```

Out[75]:

```
▾ SVC
SVC()
```

Predicting Result from Model

In [76]:

```
model.predict([[5.1,3.5,1.4,0.2]]) #Values from iris DataFrame with index=0
```

```
C:\Users\iamri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:409:
UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
```

Out[76]:

```
array([0])
```

In [77]:

```
model.predict([[7.0,3.2,4.7,1.4]]) #Values from iris DataFrame with index=50
```

```
C:\Users\iamri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:409:
UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
```

Out[77]:

```
array([1])
```

In [78]:

```
model.predict([[6.3,3.3,6.0,2.5]]) #Values from iris DataFrame with index=100
```

```
C:\Users\iamri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:409:
UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
```

Out[78]:

```
array([2])
```

Score of Model

In [79]:

```
model.score(X_train, y_train)
```

Out[79]:

```
0.9916666666666667
```