Importing Data & Library

In [1]:
```
import pandas as pd
from sklearn.datasets import load_iris
iris=load_iris()
```

In [2]:
```
dir(iris)
```

Out[2]:
```
['DESCR',
 'data',
 'feature_names',
 'filename',
 'frame',
 'target',
 'target_names']
```

In [3]:
```
iris.feature_names
```

Out[3]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [5]:
```
df=pd.DataFrame(iris.data, columns=iris.feature_names)
df.head()
```

Out[5]:

	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 [6]:
```
df['target']=iris.target
df.head()
```

Out[6]:

	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

In [7]:
```
iris.target
```

Out[7]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [8]:
```
iris.target_names
```

Out[8]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [9]:
```
df[df.target==0].head()
```

Out[9]:

	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

In [12]:
```
df[df.target==1].head()
```

Out[12]:

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

In [13]:
```
df[df.target==2].head()
```

Out[13]:

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

Creating new column based on target

In [14]:
```
df['flower_name'] =df.target.apply(lambda x: iris.target_names[x])
df.head()
```

Out[14]:

	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 [15]:
```
df[45:55]
```

Out[15]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
45	4.8	3.0	1.4	0.3	0	setosa
46	5.1	3.8	1.6	0.2	0	setosa
47	4.6	3.2	1.4	0.2	0	setosa
48	5.3	3.7	1.5	0.2	0	setosa
49	5.0	3.3	1.4	0.2	0	setosa
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

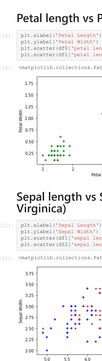
In [16]:
```
df0 = df[:50]
df1 = df[50:100]
df2 = df[100:]
```

In [17]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

Sepal length vs Sepal Width (Setosa vs Versicolor)

In [18]:
```
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.scatter(df0['sepal length (cm)'], df0['sepal width (cm)'],color="green",marker='+')
plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color="blue",marker='.')
```

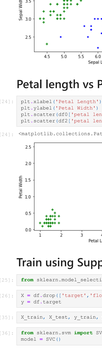
Out[18]:
```
<matplotlib.collections.PathCollection at 0x23210cf2a130>
```



Petal length vs Petal Width (Setosa vs Versicolor)

In [19]:
```
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marker='+')
plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker='.')
```

Out[19]:
```
<matplotlib.collections.PathCollection at 0x23210dffcd0>
```



Sepal length vs Sepal Width (Versicolor vs Virginica)

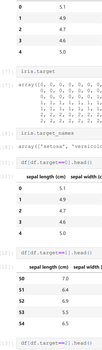
In [21]:
```
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color="blue",marker='+')
plt.scatter(df2['sepal length (cm)'], df2['sepal width (cm)'],color="red",marker='.')
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x23210df6fc10>
```

Petal length vs Petal Width (Versicolor vs Virginica)

In [22]:
```
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker='+')
plt.scatter(df2['petal length (cm)'], df2['petal width (cm)'],color="red",marker='.')
```

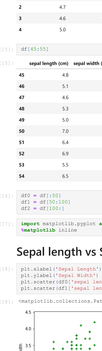
Out[22]:
```
<matplotlib.collections.PathCollection at 0x23210e27a430>
```



Sepal length vs Sepal Width (Setosa vs Verginica)

In [23]:
```
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.scatter(df0['sepal length (cm)'], df0['sepal width (cm)'],color="green",marker='+')
plt.scatter(df2['sepal length (cm)'], df2['sepal width (cm)'],color="blue",marker='.')
```

Out[23]:
```
<matplotlib.collections.PathCollection at 0x23210e87e520>
```



Petal length vs Petal Width (Setosa vs Virginica)

In [24]:
```
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marker='+')
plt.scatter(df2['petal length (cm)'], df2['petal width (cm)'],color="blue",marker='.')
```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x23210e6f1f0>
```


Train using Support Vector Machine (SVM)

In [25]:
```
from sklearn.model_selection import train_test_split
```

In [26]:
```
X = df.drop(['target','flower_name'], axis='columns')
y = df.target
```

In [27]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [28]:
```
from sklearn.svm import SVC
model = SVC()
```

In [29]:
```
model.fit(X_train, y_train)
```

Out[29]:
```
SVC()
```

In [30]:
```
model.score(X_test, y_test)
```

Out[30]:
```
1.0
```

In [39]:
```
model.predict([[4,3,0.1,5.0,3]])
```

Out[39]:
```
array([0])
```

Tuning Parameter

1. Regularization (C)

In [40]:
```
model_C = SVC(C=1)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)
```

Out[40]:
```
1.0
```

In [41]:
```
model_C = SVC(C=0.1)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)
```

Out[41]:
```
1.0
```

2. Gamma

In [42]:
```
model_g = SVC(gamma=10)
model_g.fit(X_train, y_train)
model_g.score(X_test, y_test)
```

Out[42]:
```
0.9666666666666667
```

3. Kernel

In [43]:
```
model_linear_kernal = SVC(kernel='linear')
model_linear_kernal.fit(X_train, y_train)
model_linear_kernal.score(X_test, y_test)
```

Out[43]:
```
1.0
```