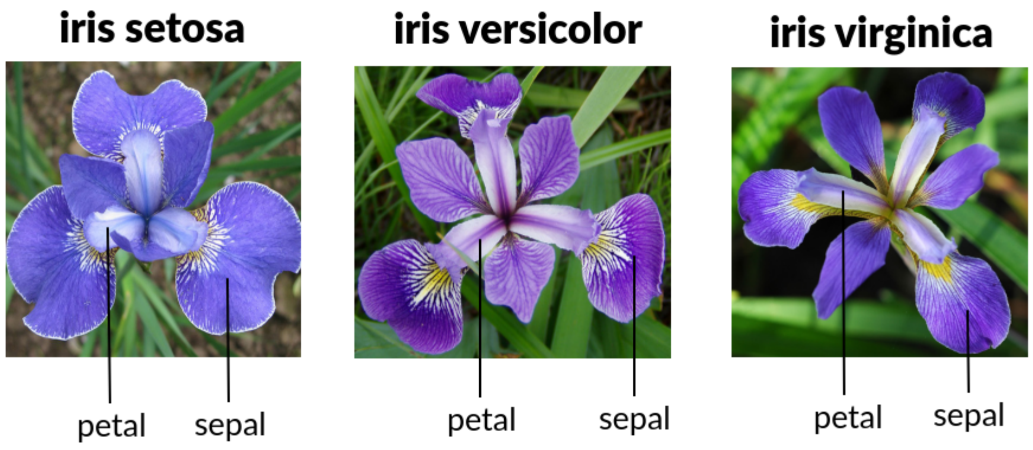
KNN (K Nearest Neighbors) Classification: Machine Learning Tutorial Using Sklearn



In [1]:

```
import pandas as pd
from sklearn.datasets import load_iris
iris = load_iris()
```

In [2]:

```
iris.feature_names
```

Out[2]:

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

In [3]:

```
iris.target_names
```

Out[3]:

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

In [4]:

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

Out[4]:

	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 [5]:

```
df.shape
```

Out[5]:

```
(150, 4)
```

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]:

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

Out[7]:

	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 [8]:

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

Out[8]:

	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

In [9]:

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

Out[9]:

	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 [10]:

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

Out[10]:

	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 [11]:

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

In [12]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

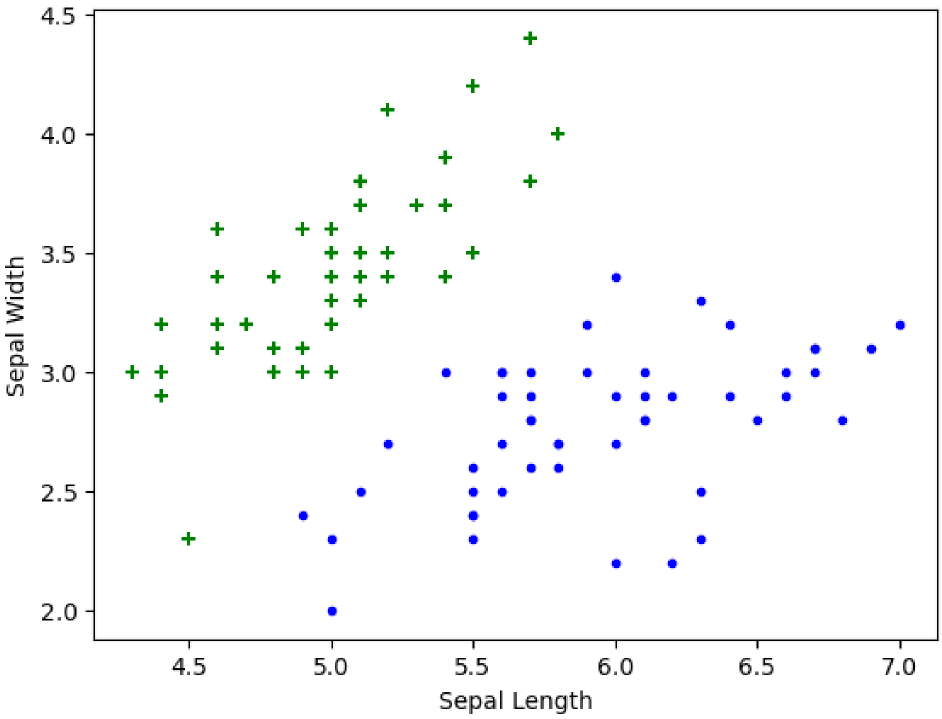
sepal length vs sepal width (Setosa vs Versicolor)

In [13]:

```
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[13]:

```
<matplotlib.collections.PathCollection at 0x211f4ca3b80>
```



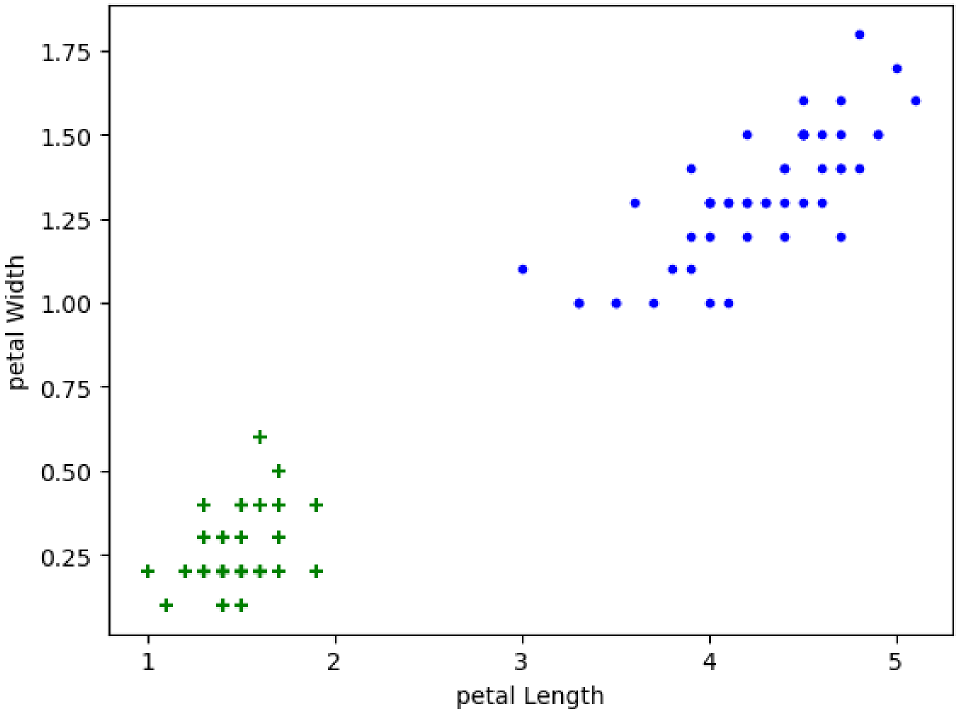
petal length vs petal width (Setosa vs Versicolor)

In [14]:

```
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[14]:

```
<matplotlib.collections.PathCollection at 0x211f55bcc10>
```



Train test split

In [15]:

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

In [16]:

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

In [17]:

```python
X_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
```

In [18]:

```python
len(X_train)
```

Out[18]:

```
120
```

In [19]:

```python
len(x_test)
```

Out[19]:

```
30
```

Create KNN (K Nearest Neighbour Classifier)

In [20]:

```python
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=10)
```

In [21]:

```python
knn.fit(X_train, y_train)
```

Out[21]:

```
▾        KNeighborsClassifier
KNeighborsClassifier(n_neighbors=10)
```

In [22]:

```python
knn.score(x_test, y_test)
```

Out[22]:

```
0.9666666666666667
```

In [23]:

```python
from sklearn.metrics import confusion_matrix
y_pred = knn.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[23]:

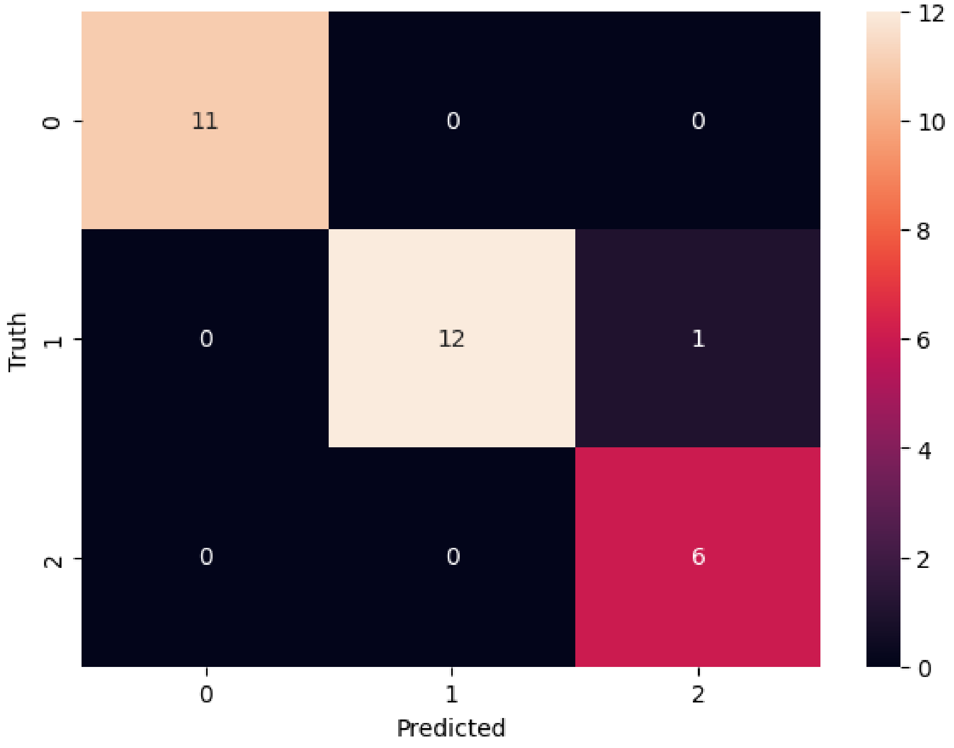
```
array([[11,  0,  0],
       [ 0, 12,  1],
       [ 0,  0,  6]], dtype=int64)
```

In [24]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(7,5))
sns.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[24]:

```
Text(58.222222222222214, 0.5, 'Truth')
```



In [25]:

```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       1.00      0.92      0.96        13
           2       0.86      1.00      0.92         6

    accuracy                           0.97        30
   macro avg       0.95      0.97      0.96        30
weighted avg       0.97      0.97      0.97        30
```

You can find this project on **Github.**