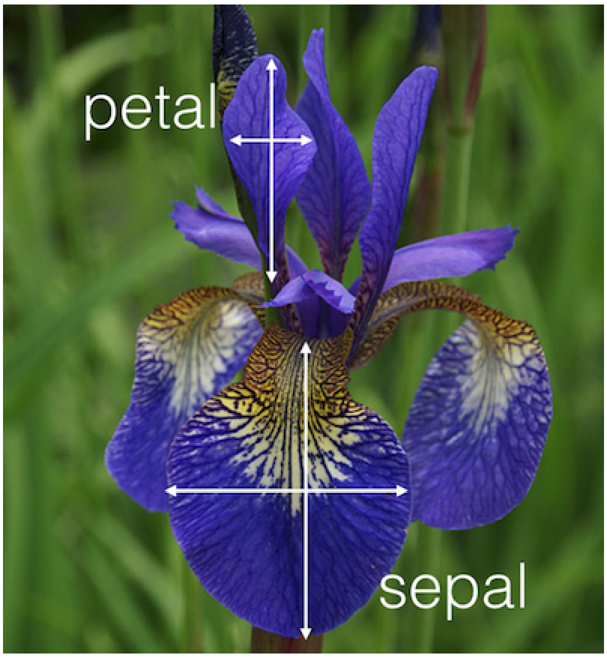
KNN (K Nearest Neighbors) Classification: Machine Tutorial Using Python Sklearn

In [15]:

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



In [16]:

```
iris.feature_names
```

Out[16]:

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

In [17]:

```
iris.target_names
```

Out[17]:

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

In [18]:

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

In [19]:

```
df
```

Out[19]:

	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
...	...	...	...	...
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 × 4 columns

In [20]:

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

Out[20]:

	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 [21]:

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

Out[21]:

	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 [22]:

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

Out[22]:

	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 [23]:

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

Out[23]:

	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 [24]:

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

Out[24]:

	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 [25]:

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

In [26]:

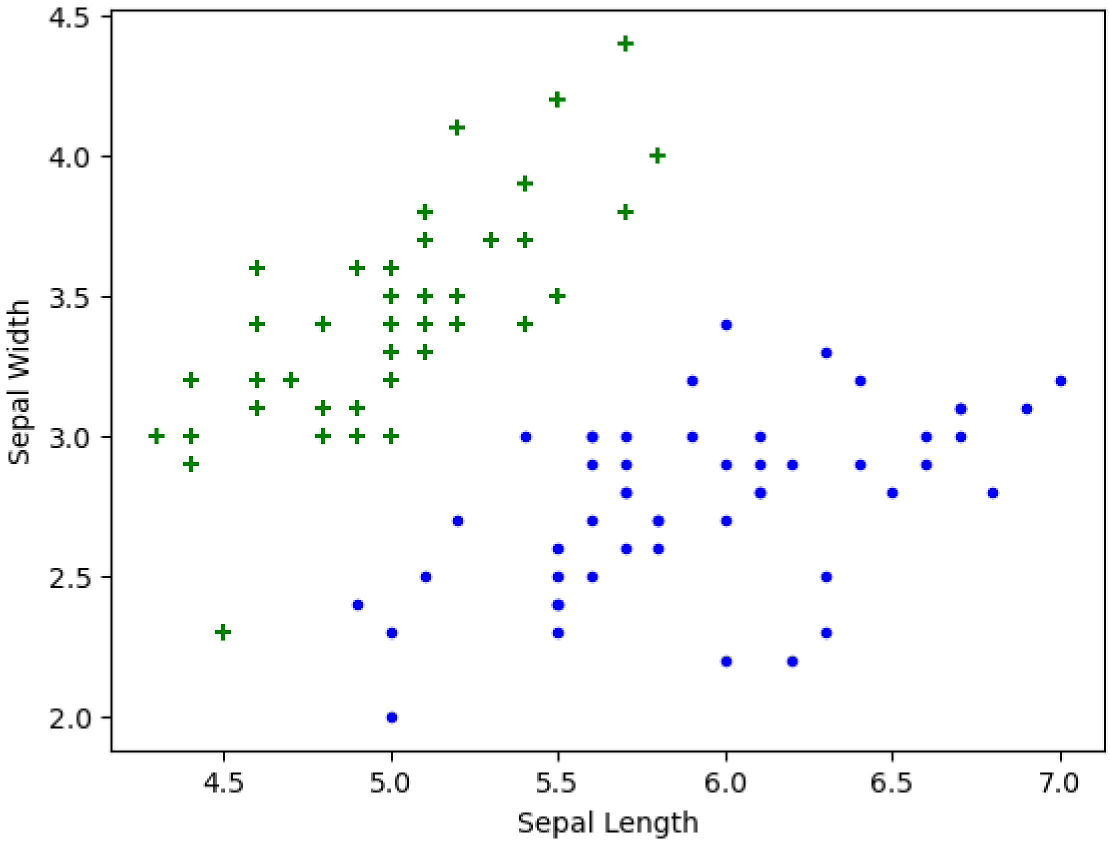
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [27]:

```
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[27]:

```
<matplotlib.collections.PathCollection at 0x1ae754a0b20>
```



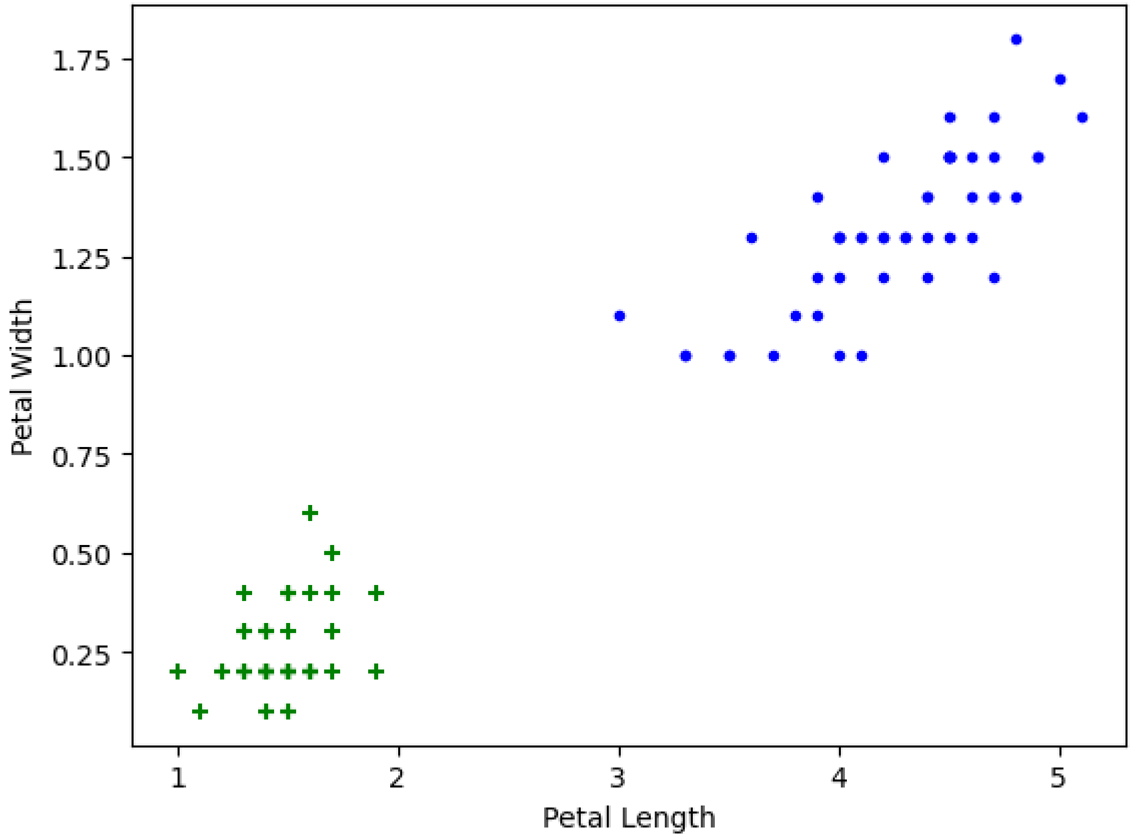
Petal length vs Pepal Width (Setosa vs Versicolor)

In [29]:

```
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[29]:

```
<matplotlib.collections.PathCollection at 0x1ae755cb670>
```



Train test split

In [30]:

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

In [32]:

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

In [33]:

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

In [34]:

```
len(X_train)
```

Out[34]:

```
120
```

In [35]:

```
len(X_test)
```

Out[35]:

```
30
```

Create KNN (K Neighrest Neighbour Classifier)

In [38]:

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

In [39]:

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

Out[39]:

```
▾        KNeighborsClassifier

KNeighborsClassifier(n_neighbors=10)
```

In [41]:

```
knn.score(X_test,y_test)
```

Out[41]:

```
0.9666666666666667
```

In [42]:

```
knn.predict([[4.8,3.0,1.5,0.3]])
```

```
C:\Users\baps\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarnin
g: X does not have valid feature names, but KNeighborsClassifier was fitte
d with feature names
  warnings.warn(
```

Out[42]:

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

Plot Confusion Matrix

In [45]:

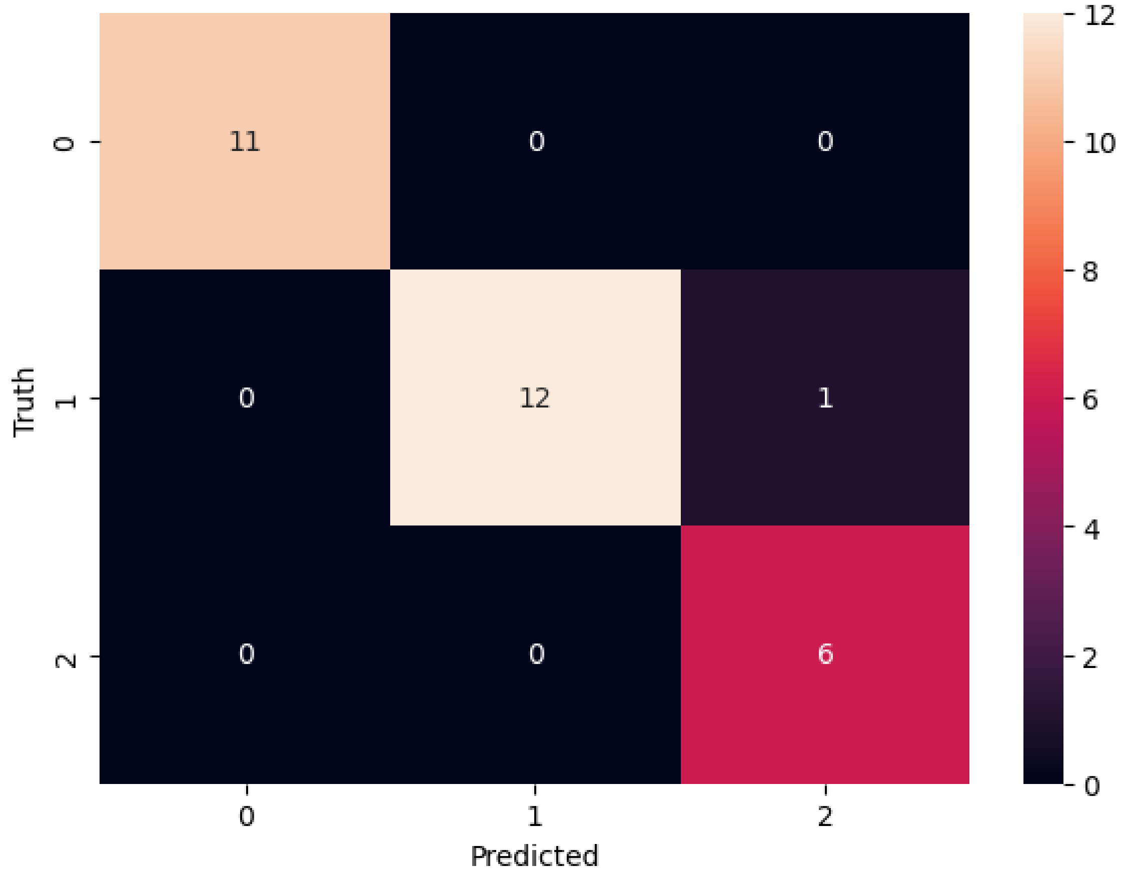
```
from sklearn.metrics import confusion_matrix
y_pred = knn.predict(X_test)
cm = confusion_matrix(y_test,y_pred)
cm
```

Out[45]:

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

In [51]:

```
%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')
plt.show()
```



Print classification report for precesion, recall and f1-score for each classes

In [52]:

```
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
```

In [ ]: