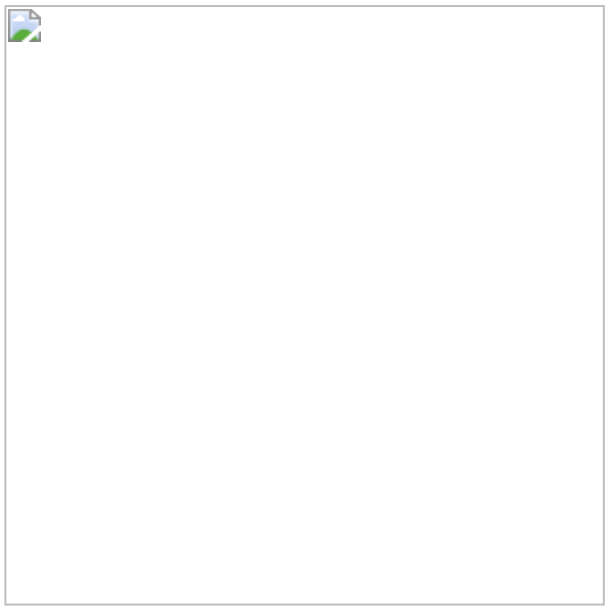
Name- Kshitij V Darwhekar

Roll No - TETB19

Batch -B2

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

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



```
iris.feature_names
```

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

```
iris.target_names
```

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

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

	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

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

	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

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

	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

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

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

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

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

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

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flowe
45	4.8	3.0	1.4	0.3	0	
46	5.1	3.8	1.6	0.2	0	
47	4.6	3.2	1.4	0.2	0	
48	5.3	3.7	1.5	0.2	0	
49	5.0	3.3	1.4	0.2	0	
50	7.0	3.2	4.7	1.4	1	v
51	6.4	3.2	4.5	1.5	1	v
52	6.9	3.1	4.9	1.5	1	v
53	5.5	2.3	4.0	1.3	1	v
54	6.5	2.8	4.6	1.5	1	v

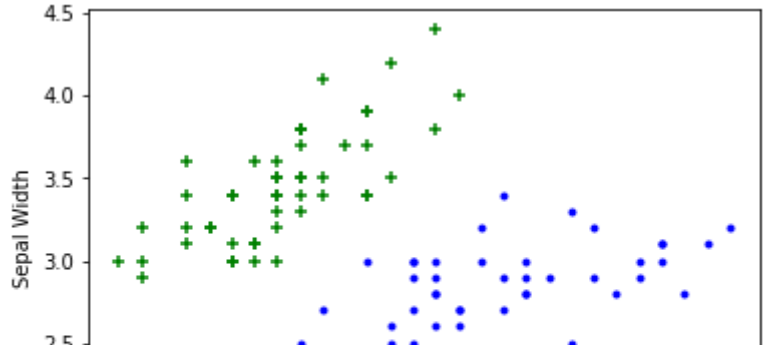
```
df0 = df[:50]
df1 = df[50:100]
df2 = df[100:]
```

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

Sepal length vs Sepal Width (Setosa vs Versicolor)

```
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='.')
```

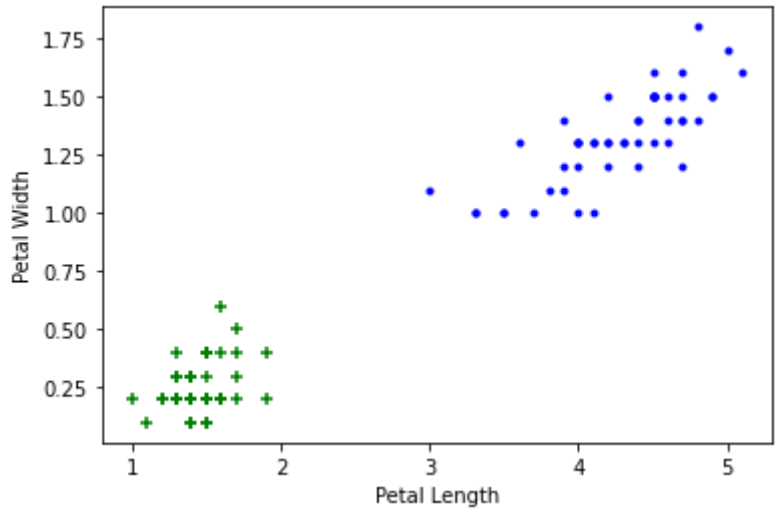
```
<matplotlib.collections.PathCollection at 0x7f8d07945f50>
```



Petal length vs Pepal Width (Setosa vs Versicolor)

```
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='.')
```

```
<matplotlib.collections.PathCollection at 0x7f8d07437910>
```



Train test split

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

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

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

```
len(X_train)
```

```
120
```

```
len(X_test)
```

```
30
```

Create KNN (K Neighrest Neighbour Classifier)

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

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

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

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

```
0.9666666666666667
```

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

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not h
  "X does not have valid feature names, but"
array([0])
```

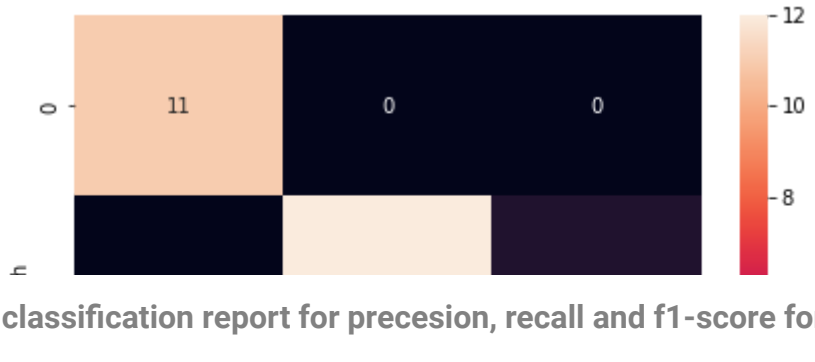
Plot Confusion Matrix

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

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

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

```
Text(42.0, 0.5, 'Truth')
```



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

```
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
```