KNN (K Nearest Neighbors) Classification

June 21, 2023

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

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

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

[7]:
```
df=pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

[7]:

	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

[9]:
```
df['target']=iris.target
df.head()
```

[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

```
2       0
3       0
4       0
```

[10]:
```
df[df.target==1].head()
```

[10]:
```
    sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
50                7.0               3.2                4.7               1.4
51                6.4               3.2                4.5               1.5
52                6.9               3.1                4.9               1.5
53                5.5               2.3                4.0               1.3
54                6.5               2.8                4.6               1.5

    target
50       1
51       1
52       1
53       1
54       1
```

[11]:
```
df[df.target==2].head()
```

[11]:
```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
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

     target
100       2
101       2
102       2
103       2
104       2
```

[13]:
```
df['flower_name']=df.target.apply(lambda x :iris.target_names[x])
df[51:54]
```

[13]:
```
    sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
51                6.4               3.2                4.5               1.5
52                6.9               3.1                4.9               1.5
53                5.5               2.3                4.0               1.3

    target flower_name
51       1  versicolor
52       1  versicolor
```

```
53       1  versicolor
```

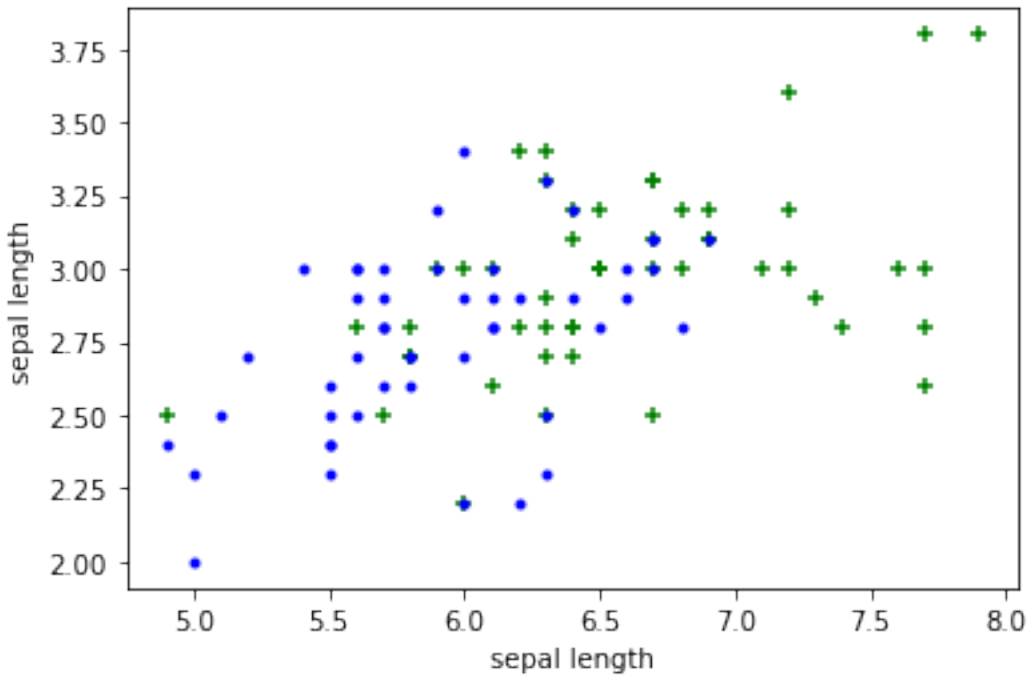
[14]:
```
df0=df[:50]
df1=df[51:100]
df0=df[100:]
```

[18]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

1 Sepal length vs Sepal Width (Setosa vs Versicolor)

[19]:
```
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='.')
plt.xlabel('sepal length')
plt.ylabel('sepal length')
```

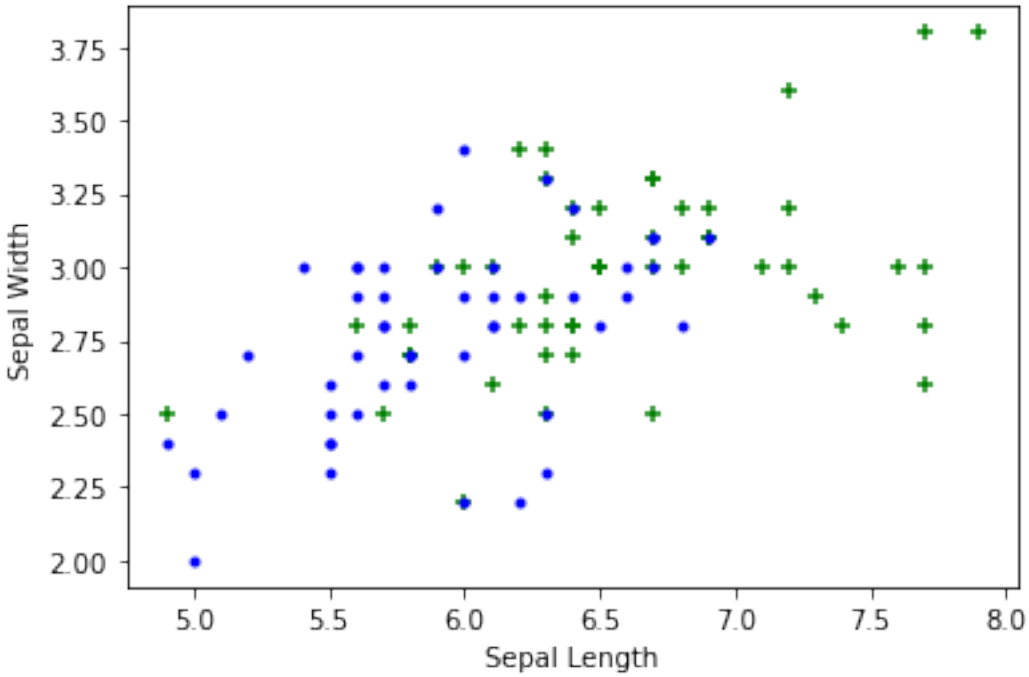
[19]:
```
Text(0, 0.5, 'sepal length')
```



[20]:
```

```

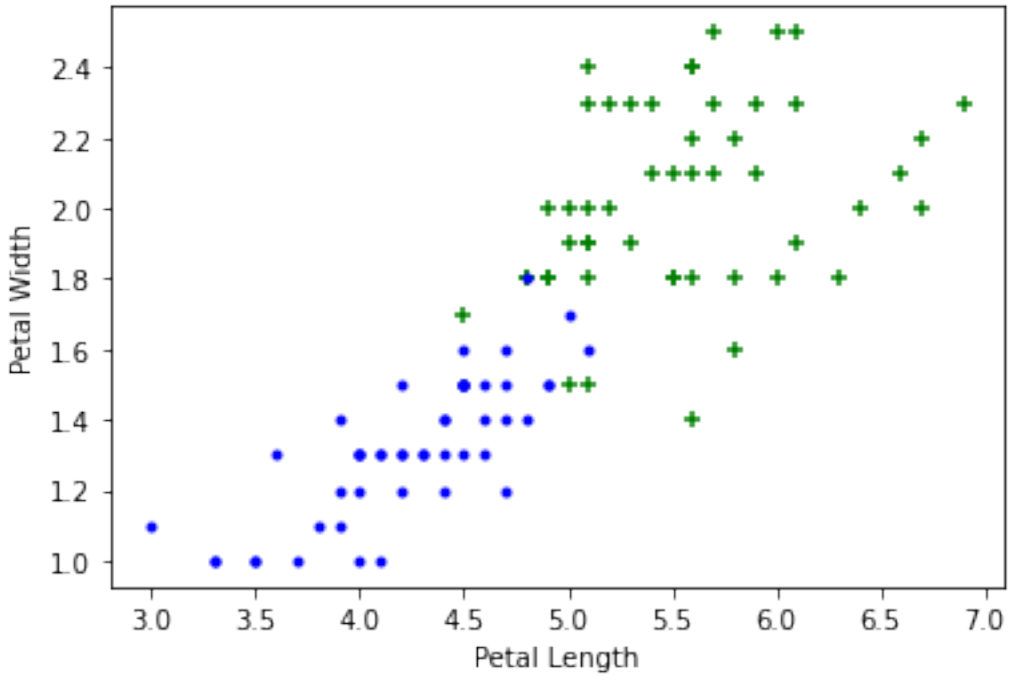
[20]:
```
<matplotlib.collections.PathCollection at 0x1f8000a7400>
```



2 Petal length vs Pepal Width (Setosa vs Versicolor)

[21]:
```
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='.')
```

[21]:
```
<matplotlib.collections.PathCollection at 0x1f800120730>
```



[22]:
```
X = df.drop(['target','flower_name'], axis='columns')
y = df.target
```

[23]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
  ↪2,random_state=1)
```

[24]:
```
len(X_train)
```

[24]: 120

[25]:
```
len(X_test)
```

[25]: 30

3 Create KNN (K Neighrest Neighbour Classifier)

[29]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=10)
```

[30]:
```
knn.fit(X_train, y_train)
```

[30]: KNeighborsClassifier(n_neighbors=10)

[31]:
```
knn.score(X_test, y_test)
```

[31]:
```
0.9666666666666667
```

[32]:
```
knn.predict([[4.8,3.0,1.5,0.3]])
```

```
C:\Users\Rakesh\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X
does not have valid feature names, but KNeighborsClassifier was fitted with
feature names
  warnings.warn(
```

[32]:
```
array([0])
```

4 Plot Confusion Matrix

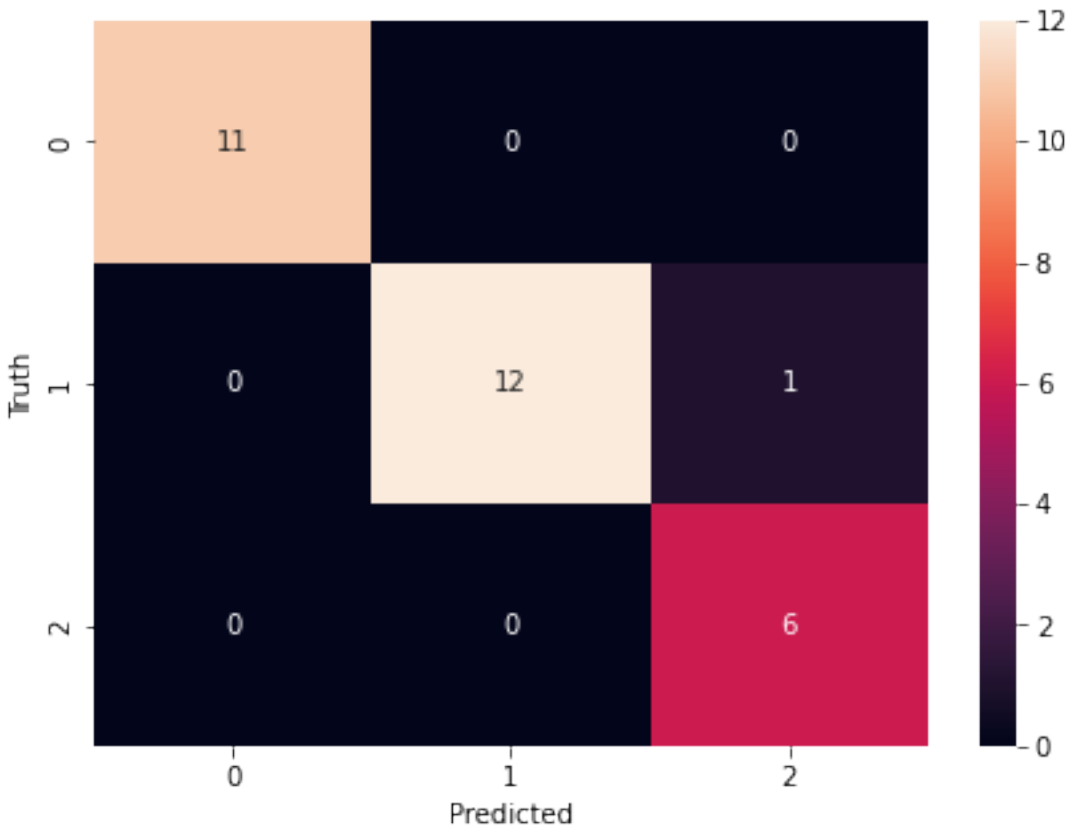
[33]:
```
from sklearn.metrics import confusion_matrix
y_pred=knn.predict(X_test)
```

[35]:
```
cm=confusion_matrix(y_test,y_pred)
cm
```

[35]:
```
array([[11,  0,  0],
       [ 0, 12,  1],
       [ 0,  0,  6]], dtype=int64)
```

[38]:
```
import seaborn as sn
plt.figure(figsize=(7,5))
sn.heatmap(cm,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

[38]:
```
Text(42.0, 0.5, 'Truth')
```



[39]:
```
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
```

[ ]: