In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('Iris.csv')
df.head()
```

Out[2]:

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| **0** | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **3** | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **4** | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

In [3]:
```
df.shape
```

Out[3]:
```
(150, 6)
```

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [5]:
```
df.duplicated().sum()
```

Out[5]:
```
0
```

In [6]:
```
df.drop_duplicates(inplace=True)
df.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
df.shape
```

Out[7]:
```
(150, 6)
```

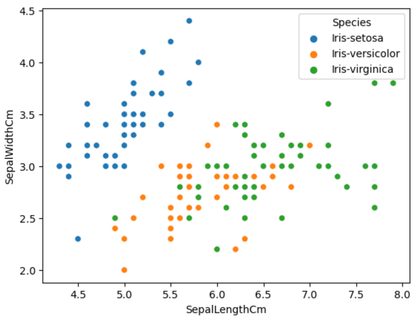
In [8]:
```
df['Species'].value_counts()
```

Out[8]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [9]:
```
x = df.iloc[:,:-1]
# x = df.drop('label',axis=1)
# x = df[['sepal_length','sepal_width','petal_length','petal_width']]
y = df.iloc[:,-1]
# y = df['Species']
print(x.shape)
print(y.shape)
print(type(x))
print(type(y))
```

```
(150, 5)
(150,)
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.series.Series'>
```

In [10]:
```
sns.scatterplot(x=df['SepalLengthCm'],y=df['SepalWidthCm'],hue=df['Species'])
plt.show()
```




In [11]:
```
from sklearn.model_selection import train_test_split
```

In [12]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(112, 5)
(38, 5)
(112,)
(38,)
```

In [13]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [14]:
```
m1 = KNeighborsClassifier(n_neighbors=11)
m1.fit(x_train,y_train)
```

Out[14]:
```
KNeighborsClassifier(n_neighbors=11)
```

In [15]:
```
# Accuracy
print('Training score',m1.score(x_train,y_train))
print('Testing score',m1.score(x_test,y_test))
```

```
Training score 1.0
Testing score 1.0
```

In [16]:
```
ypred = m1.predict(x_test)
print(ypred)
```

```
['Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-versicolor' 'Iris-virginica']
```

In [17]:
```
from sklearn.metrics import confusion_matrix,classification_report
```

In [18]:
```
cm = confusion_matrix(y_test,ypred)
print(cm)
print(classification_report(y_test,ypred))
```

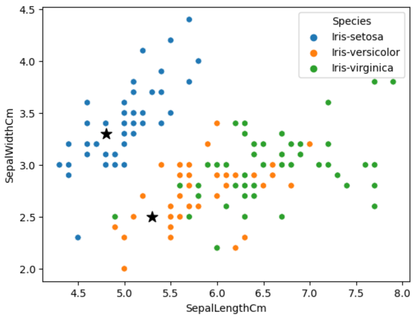
```
[[15  0  0]
 [ 0 13  0]
 [ 0  0 10]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        15
Iris-versicolor       1.00      1.00      1.00        13
 Iris-virginica       1.00      1.00      1.00        10

       accuracy                           1.00        38
      macro avg       1.00      1.00      1.00        38
   weighted avg       1.00      1.00      1.00        38
```

In [19]:
```
x_train.head()
```

Out[19]:

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm |
|---|---|---|---|---|---|
| **3** | 4 | 4.6 | 3.1 | 1.5 | 0.2 |
| **77** | 78 | 6.7 | 3.0 | 5.0 | 1.7 |
| **137** | 138 | 6.4 | 3.1 | 5.5 | 1.8 |
| **131** | 132 | 7.9 | 3.8 | 6.4 | 2.0 |
| **117** | 118 | 7.7 | 3.8 | 6.7 | 2.2 |

In [20]:
```
sns.scatterplot(x=df['SepalLengthCm'],y=df['SepalWidthCm'],hue=df['Species'])
plt.scatter([4.8,5.3],[3.3,2.5],color='black',marker='*',s=120)
plt.show()
```




In [ ]:
```
```