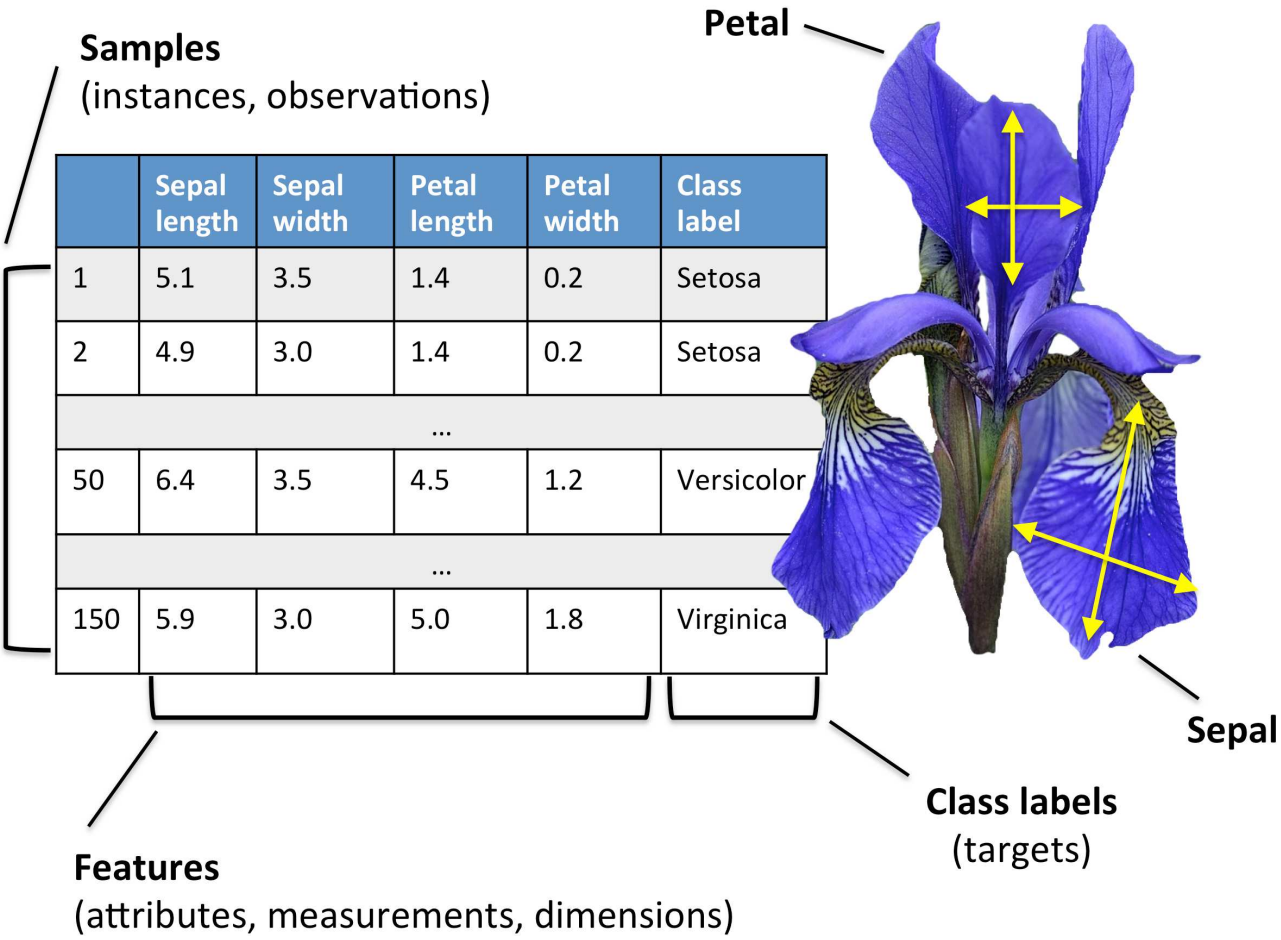
ဥပမာ-၂ k-NN Classification

ပြီးခဲ့သည့် ဥပမာ-၁ ရင်သားကင်ဆာ ဖြစ် မဖြစ်ကိုသာ ခန့်မှန်းသည့် k-NN Classification ဖြစ်သည်။ အခုအခါ ပန်းပွင့်များ၏ အချက်အလက်များ ဒေတာများကို သုံး၍ ပန်းပွင့်အမည်များကို ခန့်မှန်းချက် ထုတ်ပေးသည့် multiclass Classification ဖြစ်သည်။

Iris-setosa, Iris-versicolor နှင့် Iris-virginica တို့၏ ပွင့်ဖပ်အလျား (Sepal Length), ပွင့်ဖပ်အနံ (Sepal Width), ပွင့်ချပ်အလျား (Petal Length) နှင့် ပွင့်ချပ်အနံ (Petal Width) တို့၏ အတိုင်းအတာများကို ပါဝင်သည့် ဒေတာများဖြစ်သည်။

ထိုပန်းပွင့် (၃)မျိုး ဒေတာများဖြင့် k-NN multiclass classification model တည်ဆောက်ပြီး test ဒေတာများ ထည့်ကာ မည်သည့် ပန်းပွင့် အမျိုးအစား ဖြစ်မည်ကို ခန့်မှန်းကြမည်။



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

pandas နှင့် ပုံများဆွဲရန် matplotlib.pyplot ကို import လုပ်သည်။ sklearn.neighbors မှ KNeighborsClassifier ကို import လုပ်သည်။

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier as KNN
```

iris dataset ကို ဖတ်သည်။ ပထမဆုံး row (၅)ခုကို .head() ဖြင့် ကြည့်သည်။

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

Out[2]:

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

ပွင့်ဖပ်အလျား (Sepal Length), ပွင့်ဖပ်အနံ (Sepal Width), ပွင့်ချပ်အလျား (Petal Length) နှင့် ပွင့်ချပ်အနံ (Petal Width) လေးမျိုးအနက်မှ ပွင့်ဖပ်အလျား (Sepal Length) နှင့် ပွင့်ချပ်အနံ (Petal Width) ကို model တည်ဆောက်ရန် classification feature များ အဖြစ် အသုံးပြုသည်။

Iris-setosa အား အနီရောင် 'x' အဖြစ် လည်းကောင်း Iris-versicolor အား အပြာရောင် '*' အဖြစ် လည်းကောင်း Iris-virginica အား အနက်ရောင် 'o' အဖြစ် လည်းကောင်း သတ်မှတ်၍ ဂရပ်ပုံဆွဲသည်။

`plt.figure(figsize=(10, 7))` ဖြင့် ဂရပ်ပုံ အရွယ်အစားကို သတ်မှတ်ပေးသည်။

for name, group in df.groupby('Species'): ဖြင့်

`for name, group in df.groupby('Species'):` တွင် ပန်းပွင့်လေးမျိုးကို တူရာ တူရာများ စုထားသည့် `.groupby('  ')` ဖြင့် စု၍ ပန်းပွင့်နာမည်ဖြင့် တစ်မျိုးချင်းစီကို for loop ပတ်သည်။

`plt.scatter(group['Sepal Length'], group['Petal Width']` တွင် Sepal Length အတိုင်းအတာကို X ဝင်ရိုး တန်းဖိုးများ အဖြစ် နှင့် Petal Width အတိုင်းအတာကို Y ဝင်ရိုး တန်းဖိုးများ အဖြစ် ထား၍ scatter ဂရပ်ဆွဲသည်။

`plt.title('Species Classification Sepal Length vs Petal Width');` ဂရပ်နာမည်ရေးသည်။

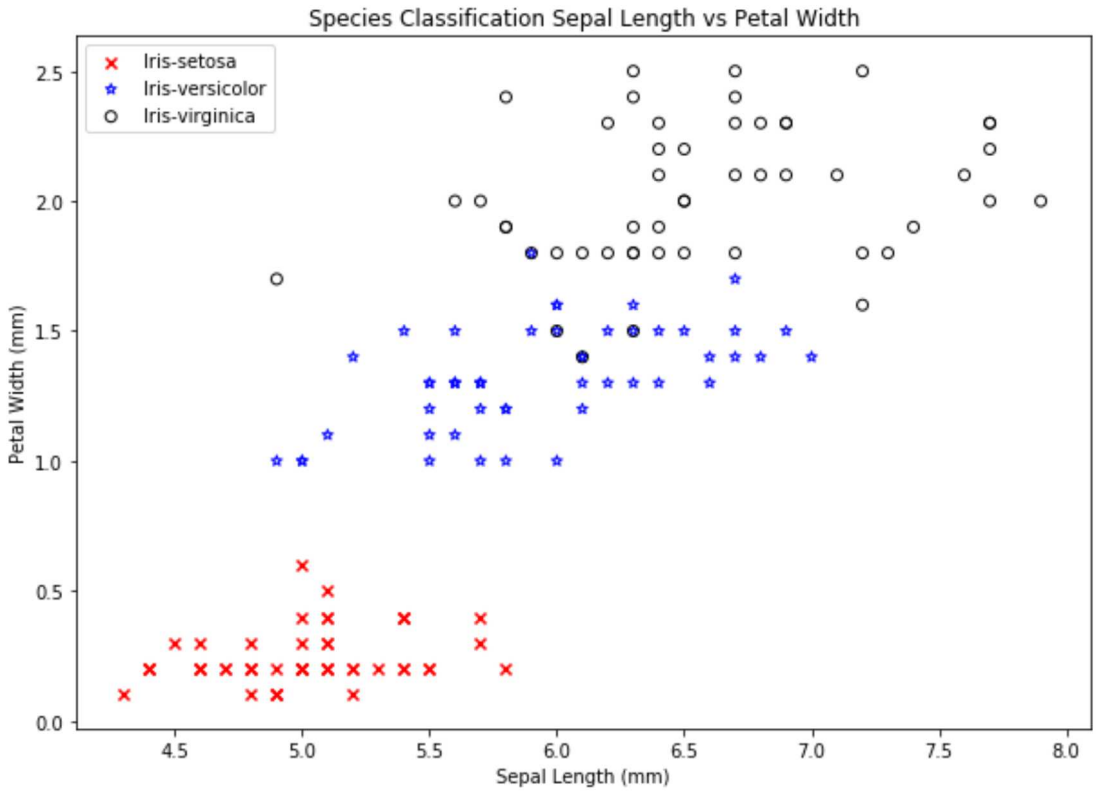
`plt.xlabel('Sepal Length (mm)');` X ဝင်ရိုး နာမည်ရေးသည်။

`plt.ylabel('Petal Width (mm)');` Y ဝင်ရိုး နာမည်ရေးသည်။

`plt.legend();` ပန်းပွင့်နာမည် များကို legend အဖြစ် ရေးသည်။

In [3]:
```
markers = {
    'Iris-setosa': {'marker': 'x', 'facecolor': 'r', 'edgecolor': 'r
'},
    'Iris-versicolor': {'marker': '*', 'facecolor': 'none', 'edgecol
or': 'b'},
    'Iris-virginica': {'marker': 'o', 'facecolor': 'none', 'edgecolo
r': 'k'},
}
plt.figure(figsize=(10, 7))
for name, group in df.groupby('Species'):
    plt.scatter(group['Sepal Length'], group['Petal Width'],
                label=name,
                marker=markers[name]['marker'],
                facecolors=markers[name]['facecolor'],
                edgecolor=markers[name]['edgecolor'])

plt.title('Species Classification Sepal Length vs Petal Width');
plt.xlabel('Sepal Length (mm)');
plt.ylabel('Petal Width (mm)');
plt.legend();
```



sample အမှတ် 134 ကို test point အဖြစ် သတ်မှတ်သည်။ ထို့ test point သည် အစုနှစ်ခုအကြား နယ်နိမိတ်(boundary of two classes) တွင် ရှိနေသောကြောင့် တမင်ရည်ရွယ်၍ ရွေးချယ်ခြင်းဖြစ်သည်။ ထို sample အမှတ် 134 ခုကို training data မှ ဖယ်ထုတ်သည်။

We are going to select sample 134 as the test point. This point was specifically chosen as it lies at the boundary of two classes. Lets remove sample 134 from the training data

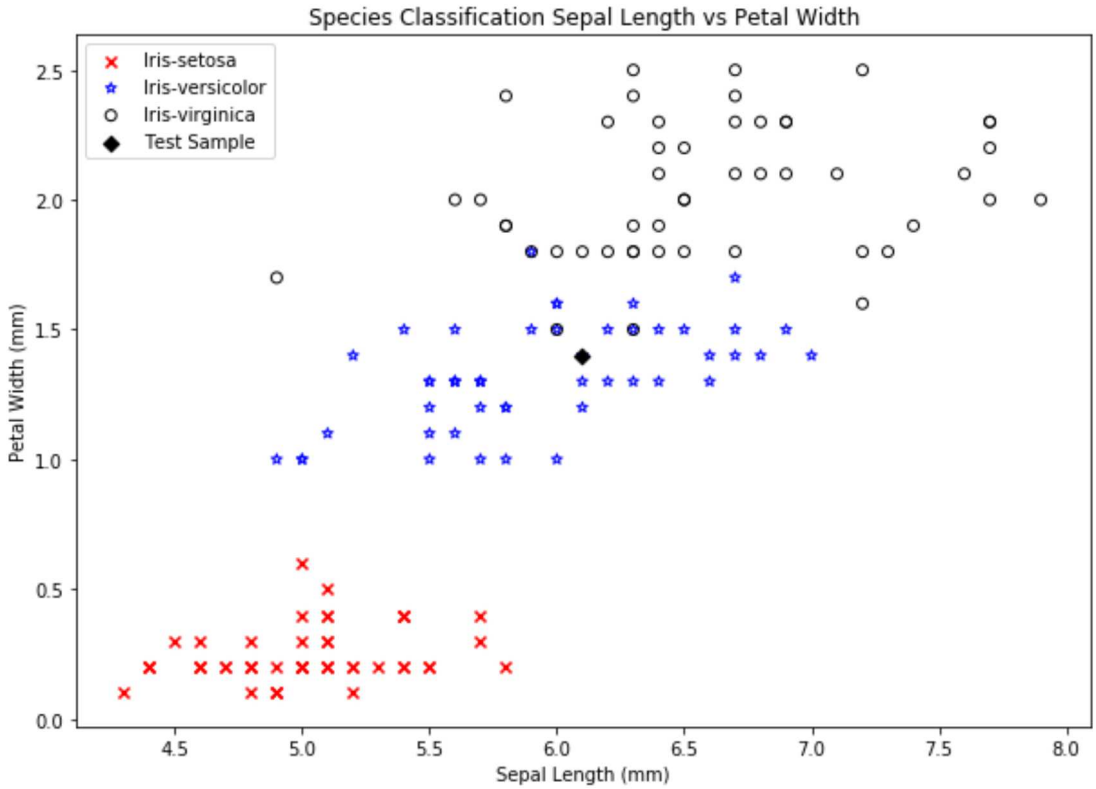
In [4]:
```
df_test = df.iloc[134]
df = df.drop([134]) # Remove the sample
df_test
```

Out[4]:
```
Sepal Length                6.1
Sepal Width                 2.6
Petal Length                5.6
Petal Width                 1.4
Species          Iris-virginica
Name: 134, dtype: object
```

ထို sample အမှတ် 134 ခုကို training data မှ ဖယ်ထုတ်ပြီးနောက် ဂရပ်ပုံ ဆွဲ၍ ပြန်ကြည့်သည်။

In [5]:
```
plt.figure(figsize=(10, 7))
for name, group in df.groupby('Species'):
    plt.scatter(group['Sepal Length'], group['Petal Width'],
                label=name,
                marker=markers[name]['marker'],
                facecolors=markers[name]['facecolor'],
                edgecolor=markers[name]['edgecolor'])

plt.scatter(df_test['Sepal Length'], df_test['Petal Width'], label='
Test Sample', c='k', marker='D')
plt.title('Species Classification Sepal Length vs Petal Width');
plt.xlabel('Sepal Length (mm)');
plt.ylabel('Petal Width (mm)');
plt.legend();
```



k အရေအတွက် (၃) (3 nearest neighbours) ဖြင့် KNN model တည်ဆောက်သည်။ `model = KNN(n_neighbors=3)` ဖြင့် KNN model တည်ဆောက်သည်။ K-NN classifier model များအား ထူးခြားသည့်အချက်မှာ encode လုပ်ပေးရန် မလိုခြင်း ဖြစ်သည်။

တည်ဆောက်ပြီးသည့် KNN model ထဲသို့ ဒေတာများ ထည့်၍ fit လုပ်ပေးရသည်။ fit လုပ်ရန် အတွက် X train data နှင့် y train data ထည့်ပေးရသည်။

`X=df[['Petal Width', 'Sepal Length']]` ဖြင့် X train data ထည့်ပေးသည်။ X သည် input train data ဖြစ်သည်။ ထို့ကြောင့် feature ဟုခေါ်သည့် input data နှစ်မျိုးကို 'Petal Width', 'Sepal Length' ထည့်ပေးသည်။ `y=df.Species` ဖြင့် y train data ထည့်ပေးသည်။

K-NN classifier model များအား ထူးခြားသည့်အချက်မှာ encode လုပ်ပေးရန် မလိုခြင်း ဖြစ်သည်။ (One of the great things about K-NN classifiers is that we do not need to encode the classes for the method to work.)

In [6]:
```
model = KNN(n_neighbors=3)
model.fit(X=df[['Petal Width', 'Sepal Length']], y=df.Species)
```

Out[6]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minko
wski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')
```

တည်ဆောက်ထားသည့် model မှန်ကန်မှုမည်မျှရှိသည်(score) ကို model.score() ဖြင့် ရှာသည်။

In [7]:
```
model.score(X=df[['Petal Width', 'Sepal Length']], y=df.Species)
```

Out[7]:
```
0.9731543624161074
```

test point တစ်ခုထည့်ပေးပြီး ထို test point ပါဝင်မည့် class ကို ခန့်မှန်းသည်။ တစ်နည်းအားဖြင့် ထို test point သည် မည်သည့် ပန်းပွင့်ဖြစ်မည်ကို ခန့်မှန်းသည်။ (Predict the class for the test point)

In [8]:
```
model.predict(df_test[['Petal Width', 'Sepal Length']].values.reshap
e((-1, 2)))[0]
```

Out[8]:
```
'Iris-versicolor'
```

လက်တွေ့ အဖြေမှန်နှင့် နှိုင်းယှဉ်သည်။ (Compare against the actual predictions)

In [9]:
```
df.iloc[134].Species
```

Out[9]:
```
'Iris-virginica'
```

model ၏ ခန့်မှန်းပေးချက် မမှန်ပါ။ အစုနှစ်ခုအကြားတွင် ကျရောက်နေသောကြောင့် ဖြစ်ပါသည်။ This prediction is incorrect, but given its position at the boundary this isn't necessarily surprising.