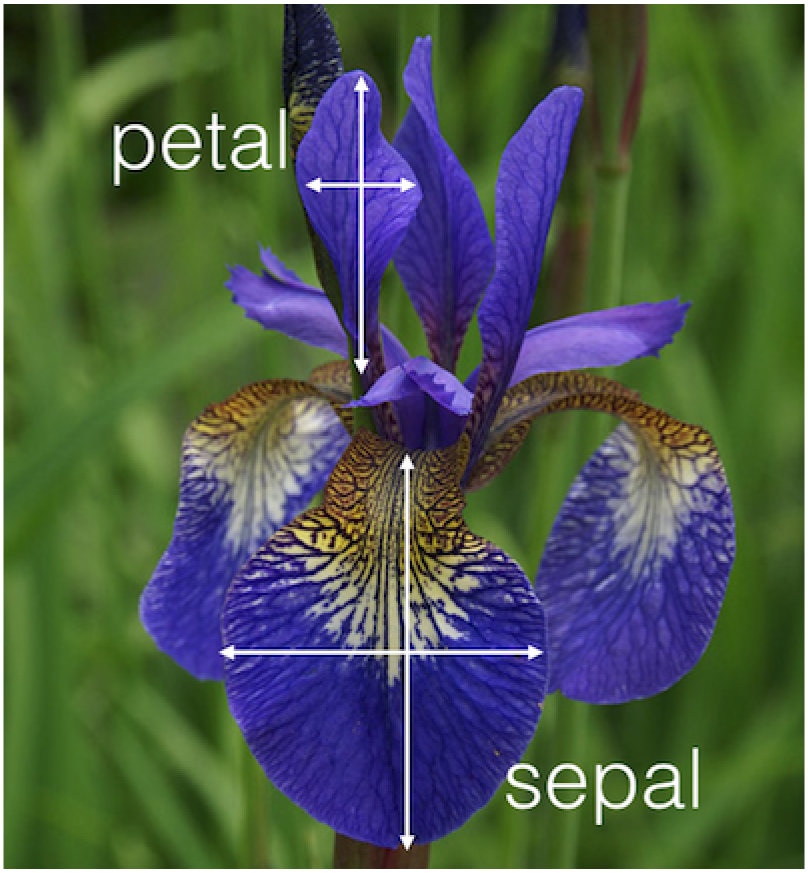
## Exercise for k means tutorial



1. Use iris flower dataset from sklearn library and try to form clusters of flowers using petal width and length features. Drop other two features for simplicity.
2. Figure out if any preprocessing such as scaling would help here
3. Draw elbow plot and from that figure out optimal value of k

In [1]:

```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
from sklearn.datasets import load_iris
%matplotlib inline
```

In [2]:

```
iris = load_iris()
```

In [3]:
```
df = pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

Out[3]:

| | 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 |

In [4]:
```
df['flower'] = iris.target
df.head()
```

Out[4]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | flower |
|---|---|---|---|---|---|
| **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 [5]:
```
df.drop(['sepal length (cm)', 'sepal width (cm)', 'flower'],axis='columns',inplac
```

In [6]:
```
df.head(3)
```

Out[6]:

| | petal length (cm) | petal width (cm) |
|---|---|---|
| **0** | 1.4 | 0.2 |
| **1** | 1.4 | 0.2 |
| **2** | 1.3 | 0.2 |

In [7]:
```
km = KMeans(n_clusters=3)
yp = km.fit_predict(df)
yp
```

Out[7]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [8]:
```
df['cluster'] = yp
df.head(2)
```

Out[8]:

| | petal length (cm) | petal width (cm) | cluster |
|---|---|---|---|
| **0** | 1.4 | 0.2 | 1 |
| **1** | 1.4 | 0.2 | 1 |

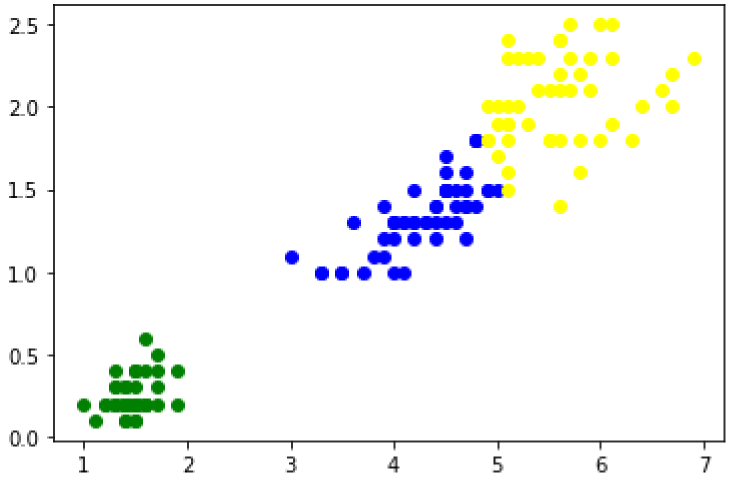
In [9]:
```
df.cluster.unique()
```

Out[9]:
```
array([1, 0, 2])
```

In [10]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
```

In [11]:
```
plt.scatter(df1['petal length (cm)'],df1['petal width (cm)'],color='blue')
plt.scatter(df2['petal length (cm)'],df2['petal width (cm)'],color='green')
plt.scatter(df3['petal length (cm)'],df3['petal width (cm)'],color='yellow')
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0xcb8bb80>
```



### Elbow Plot

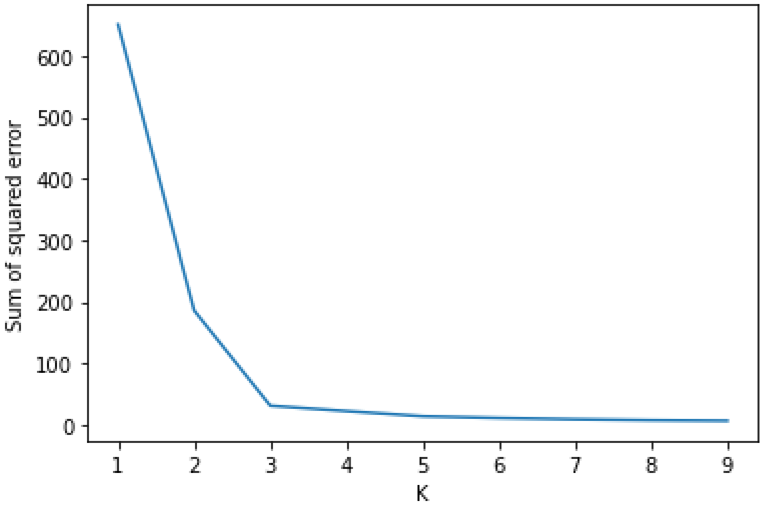
In [12]:
```
sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df)
    sse.append(km.inertia_)
```

In [13]:

```
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

Out[13]:

```
[<matplotlib.lines.Line2D at 0xd0b2b20>]
```



| Date | Author |
|---|---|
| 2021-10-03 | **Ehsan Zia** |