**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 [2]:

**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 [7]:

iris **=** load\_iris()

In [8]:

df **=** pd**.**DataFrame(iris**.**data,columns**=**iris**.**feature\_names)

df**.**head()

Out[8]:

|  | **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 [9]:

df['flower'] **=** iris**.**target

df**.**head()

Out[9]:

|  | **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 [23]:

df**.**drop(['sepal length (cm)', 'sepal width (cm)', 'flower'],axis**=**'columns',inplace**=True**)

In [24]:

df**.**head(3)

Out[24]:

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

In [28]:

km **=** KMeans(n\_clusters**=**3)

yp **=** km**.**fit\_predict(df)

yp

Out[28]:

array([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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2,

2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])

In [30]:

df['cluster'] **=** yp

df**.**head(2)

Out[30]:

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

In [31]:

df**.**cluster**.**unique()

Out[31]:

array([0, 2, 1], dtype=int64)

In [33]:

df1 **=** df[df**.**cluster**==**0]

df2 **=** df[df**.**cluster**==**1]

df3 **=** df[df**.**cluster**==**2]

In [34]:

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')

Chart, scatter chart

Description automatically generated

**Elbow Plot**

In [35]:

sse **=** []

k\_rng **=** range(1,10)

**for** k **in** k\_rng:

km **=** KMeans(n\_clusters**=**k)

km**.**fit(df)

sse**.**append(km**.**inertia\_)

In [36]:

plt**.**xlabel('K')

plt**.**ylabel('Sum of squared error')

plt**.**plot(k\_rng,sse)

Out[36]:

[<matplotlib.lines.Line2D at 0x124ce45fc88>]

Shape

Description automatically generated