IRIS

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sb
from matplotlib import pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn import metrics
# from yellowbrick.cluster import KElowVisualizer
from collections import Counter
from matplotlib.colors import ListedColormap
```

In [2]:
```
data= pd.read_csv("Iris.csv")
```

In [3]:
```
data
```

Out[3]:

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [5]:
```
# df = data.drop("SepalLengthCm", axis='columns')
data = data.drop(['Id','Species'],axis=1)
data
```

Out[5]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [6]:
```
data = data.drop(['PetalLengthCm','PetalWidthCm'],axis=1)
data
```

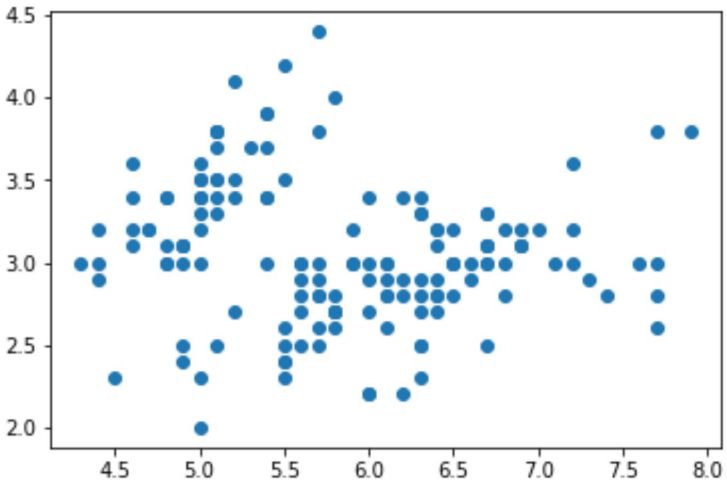
Out[6]:

	SepalLengthCm	SepalWidthCm
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6
...	...	...
145	6.7	3.0
146	6.3	2.5
147	6.5	3.0
148	6.2	3.4
149	5.9	3.0

150 rows × 2 columns

In [7]:
```
plt.scatter(data.SepalLengthCm,data.SepalWidthCm)
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x1bf81336a30>
```



In [8]:
```
s=MinMaxScaler()
data = s.fit_transform(data)
data
```

Out[8]:
```
array([[0.22222222, 0.625     ],
       [0.16666667, 0.41666667],
       [0.11111111, 0.5       ],
       [0.08333333, 0.45833333],
       [0.19444444, 0.66666667],
       [0.30555556, 0.79166667],
       [0.08333333, 0.58333333],
       [0.19444444, 0.58333333],
       [0.02777778, 0.375     ],
       [0.16666667, 0.45833333],
       [0.30555556, 0.70833333],
       [0.13888889, 0.58333333],
       [0.13888889, 0.41666667],
       [0.        , 0.41666667],
       [0.41666667, 0.83333333],
       [0.38888889, 1.        ],
       [0.30555556, 0.79166667],
       [0.22222222, 0.625     ],
       [0.38888889, 0.75      ],
```

In [9]:
```
data = pd.DataFrame(data,columns=['A','B'])
```

In [10]:
```
data
```

Out[10]:

	A	B
0	0.222222	0.625000
1	0.166667	0.416667
2	0.111111	0.500000
3	0.083333	0.458333
4	0.194444	0.666667
...	...	...
145	0.666667	0.416667
146	0.555556	0.208333
147	0.611111	0.416667
148	0.527778	0.583333
149	0.444444	0.416667

150 rows × 2 columns

In [11]:
```
model = KMeans(n_clusters=2)
model.fit(data)
y=model.fit_predict(data)
y
```

Out[11]:
```
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, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [12]:
```
data['cluster']=y
data
```

Out[12]:

	A	B	cluster
0	0.222222	0.625000	1
1	0.166667	0.416667	1
2	0.111111	0.500000	1
3	0.083333	0.458333	1
4	0.194444	0.666667	1
...	...	...	...
145	0.666667	0.416667	0
146	0.555556	0.208333	0
147	0.611111	0.416667	0
148	0.527778	0.583333	0
149	0.444444	0.416667	0

150 rows × 3 columns

In [13]:
```
cent = model.cluster_centers_
cent
```

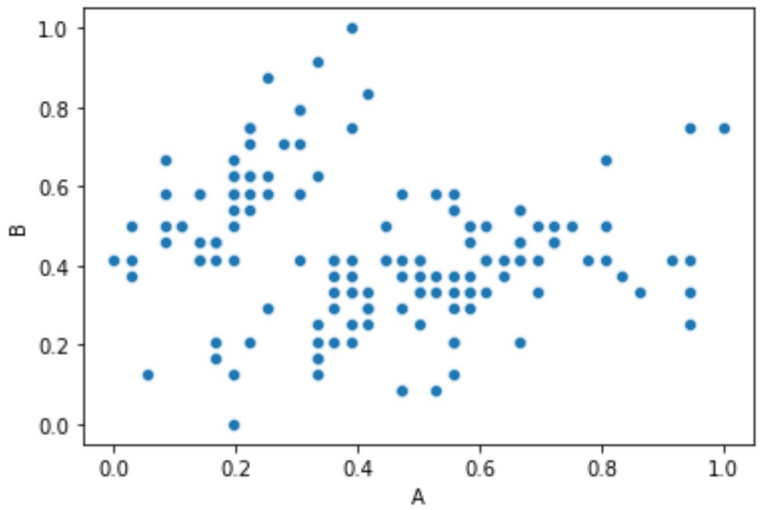
Out[13]:
```
array([[0.57035024, 0.37047101],
       [0.20402299, 0.54813218]])
```

In [14]:
```
Counter(model.labels_)
```

Out[14]:
```
Counter({1: 58, 0: 92})
```

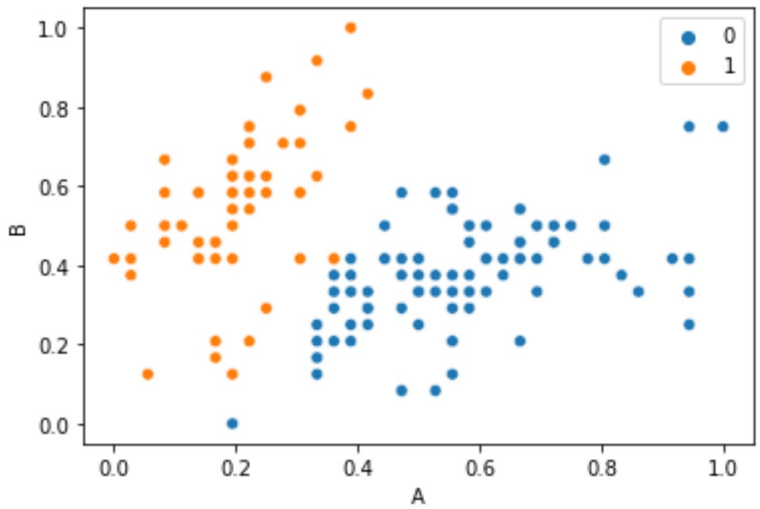
In [15]:
```
sb.scatterplot(data=data,x='A',y='B')
```

Out[15]: <AxesSubplot:xlabel='A', ylabel='B'>



In [16]:
```
sb.scatterplot(data=data,x='A',y='B',hue=model.labels_)
```

Out[16]: <AxesSubplot:xlabel='A', ylabel='B'>



In [17]:
```
k_rng = range(1,11)
sse=[]
for k in k_rng:
    km = KMeans(n_clusters = k)
    km.fit(data[['A','B']])
    sse.append(km.inertia_)
sse
```

Out[17]:
```
[12.746657664609057,
 6.850052259749757,
 4.131846697085131,
 3.2196860183926157,
 2.5444141077216686,
 2.0878705311214,
 1.8004674491345634,
 1.5399565162584332,
 1.2839316297269658,
 1.1060889654398527]
```

In [19]:
```
model = KMeans(n_clusters=3)
model.fit(data)
y=model.fit_predict(data)
y
```

Out[19]:
```
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, 2, 2, 2, 1, 1, 1, 2, 0, 2, 0, 1, 1, 1, 1, 1, 2,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 2, 2, 1,
       0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 2, 1, 2, 1, 2, 2, 0, 2, 1, 2,
       2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 2, 2,
       1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1])
```

In [20]:
```
data['cluster']=y
data
```

Out[20]:

	A	B	cluster
0	0.222222	0.625000	0
1	0.166667	0.416667	0
2	0.111111	0.500000	0
3	0.083333	0.458333	0
4	0.194444	0.666667	0
...	...	...	...
145	0.666667	0.416667	2
146	0.555556	0.208333	1
147	0.611111	0.416667	2
148	0.527778	0.583333	2
149	0.444444	0.416667	1

150 rows × 3 columns

In [21]:
```
cent = model.cluster_centers_
cent
```

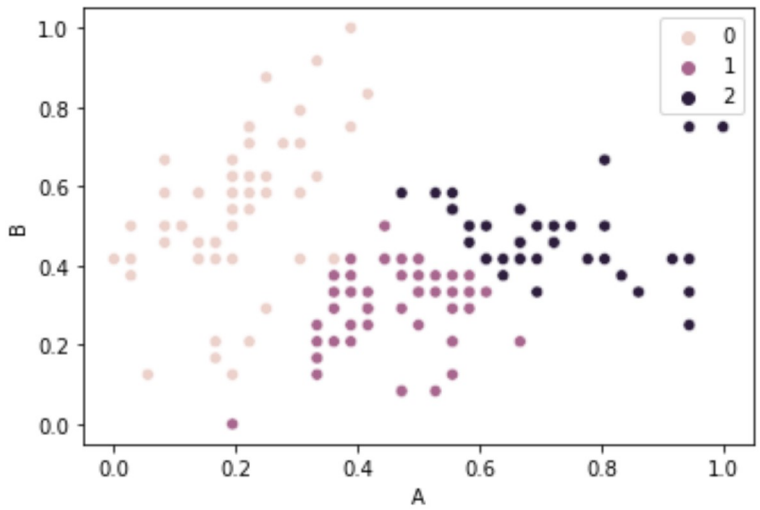
Out[21]:
```
array([[0.20402299, 0.54813218, 1.        ],
       [0.458061  , 0.28839869, 0.        ],
       [0.7100271 , 0.47256098, 0.        ]])
```

In [22]:
```
Counter(model.labels_)
```

Out[22]:
```
Counter({0: 58, 2: 41, 1: 51})
```

In [23]:
```
sb.scatterplot(data=data,x='A',y='B',hue=model.labels_)
```

Out[23]:
```
<AxesSubplot:xlabel='A', ylabel='B'>
```



In [24]:
```
plt.xlabel('k')
plt.ylabel('sse')
plt.plot(k_rng,sse)
```

Out[24]:
```
[<matplotlib.lines.Line2D at 0x1bf81b25910>]
```