

```
In [39]: import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [40]: iris = load_iris()
```

```
In [41]: dir(iris)
```

```
Out[41]: ['DESCR',
          'data',
          'feature_names',
          'filename',
          'frame',
          'target',
          'target_names']
```

```
In [83]: df = pd.DataFrame( iris.data, columns = iris.feature_names)
df.head()
```

```
Out[83]:
```

	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 [84]: df = df.drop(['petal length (cm)', 'petal width (cm)'], axis = 'columns')
df.head()
```

```
Out[84]:
```

	sepal length (cm)	sepal width (cm)
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

```
In [85]: km = KMeans(n_clusters = 3)
y_predicted = km.fit_predict(df[['sepal length (cm)', 'sepal width (cm)']])
y_predicted
```

```
Out[85]: 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, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1,
        2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1,
        1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1,
        1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2])
```

```
In [86]: df['clusters'] = y_predicted
df.head()
```

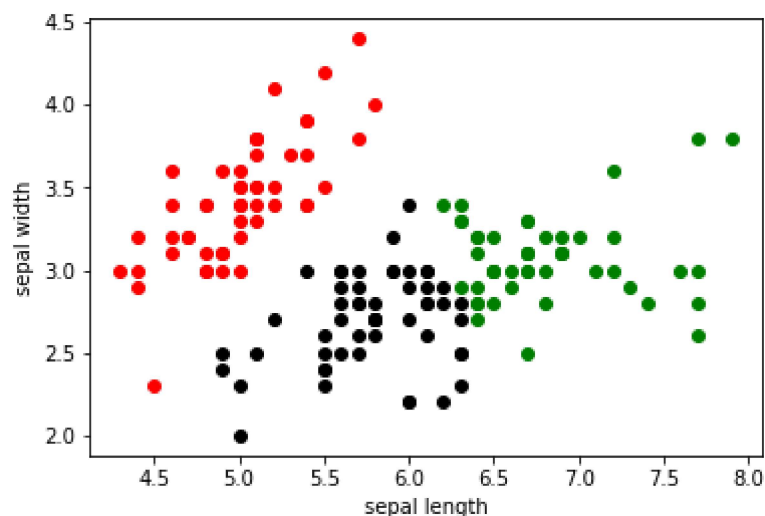
```
Out[86]:
```

	sepal length (cm)	sepal width (cm)	clusters
0	5.1	3.5	0
1	4.9	3.0	0
2	4.7	3.2	0
3	4.6	3.1	0
4	5.0	3.6	0

```
In [87]: df1 = df[df.clusters == 0]
df2 = df[df.clusters == 1]
df3 = df[df.clusters == 2]
```

```
In [88]: plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'], color = 'red')
plt.scatter(df2['sepal length (cm)'], df2['sepal width (cm)'], color = 'green')
plt.scatter(df3['sepal length (cm)'], df3['sepal width (cm)'], color = 'black')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
```

```
Out[88]: Text(0, 0.5, 'sepal width')
```



In [ ]:

In [89]: km.cluster\_centers\_

```
Out[89]: array([[5.006      , 3.428      ],
                [6.81276596, 3.07446809],
                [5.77358491, 2.69245283]])
```

In [90]: scaler = MinMaxScaler()

```
In [91]: scaler.fit(df[['sepal length (cm)']])
df['sepal length (cm)'] = scaler.transform(df[['sepal length (cm)']])
scaler.fit(df[['sepal width (cm)']])
df['sepal width (cm)'] = scaler.transform(df[['sepal width (cm)']])
```

```
In [62]: km = KMeans(n_clusters = 3)
y_predicted = km.fit_predict(df[['sepal length (cm)', 'sepal width (cm)']])
y_predicted
```

```
Out[62]: 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, 2, 0, 0,
                0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1,
                2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1,
                1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1,
                1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2])
```

```
In [93]: df['clusters'] = y_predicted
df.head()
```

Out[93]:

	sepal length (cm)	sepal width (cm)	clusters
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

In [101]: km.cluster\_centers\_

```
Out[101]: array([[5.006      , 3.428      ],
                [6.81276596, 3.07446809],
                [5.77358491, 2.69245283]])
```

```
In [102]: df1 = df[df.clusters == 0]
df2 = df[df.clusters == 1]
df3 = df[df.clusters == 2]
df1.head()
```

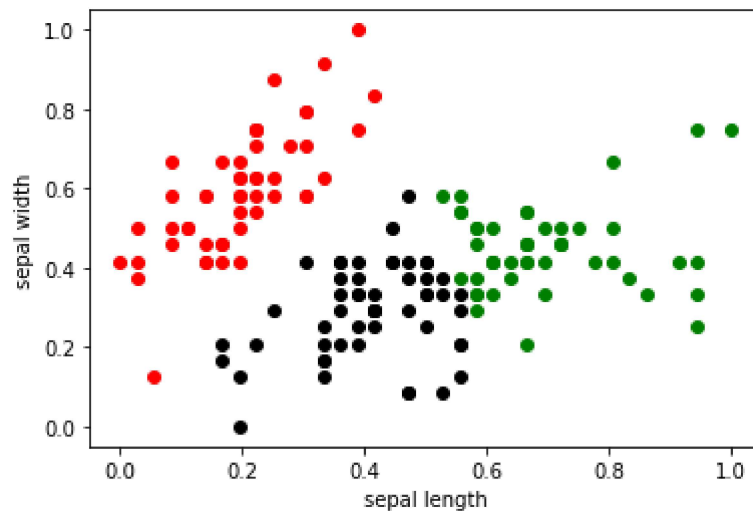
Out[102]:

	sepal length (cm)	sepal width (cm)	clusters
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

```
In [107]: plt.scatter(df1['sepal length (cm)'],df1['sepal width (cm)'] , color = 'red' )
plt.scatter(df2['sepal length (cm)'],df2['sepal width (cm)'] , color = 'green' )
plt.scatter(df3['sepal length (cm)'],df3['sepal width (cm)'] , color = 'black' )

plt.xlabel('sepal length')
plt.ylabel('sepal width')
```

Out[107]: Text(0, 0.5, 'sepal width')

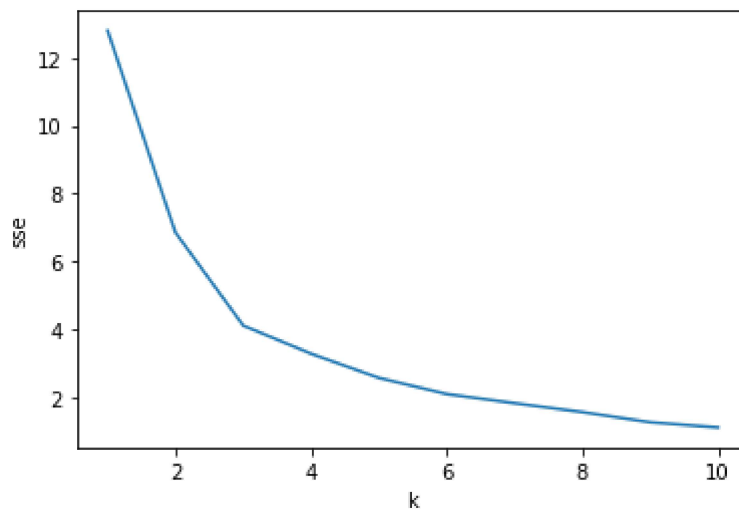


```
In [110]: sse = []  
          k_rng = range(1,11)  
          for k in k_rng:  
              km = KMeans(n_clusters = k)  
              km.fit(df[['sepal length (cm)', 'sepal width (cm)']])  
              sse.append(km.inertia_)
```

D:\ashwa\ana\lib\site-packages\sklearn\cluster\\_kmeans.py:881: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP\_NUM\_THREADS=1.  
 warnings.warn(

```
In [111]: plt.plot(k_rng, sse)  
          plt.xlabel('k')  
          plt.ylabel('sse')
```

Out[111]: Text(0, 0.5, 'sse')



In [ ]: