Name : Tavhare Ruchita Sharad

class : BE AI&DS

Roll No.: 61

Subject: Computer Laboratory-I(ML)

Title:Implement K-Means clustering on Iris.csv dataset. Determine the number of clustersusing the elbow method.

In [6]:
```
import pandas as pd # Pandas (version : 1.1.5)
import numpy as np # Numpy (version : 1.19.2)
import matplotlib.pyplot as plt # Matplotlib (version : 3.3.2)
from sklearn.cluster import KMeans # Scikit Learn (version : 0.23.2)
import seaborn as sns # Seaborn (version : 0.11.1)
```

In [8]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [10]:
```
data = pd.read_csv('iris.csv')
```

In [12]:
```
data
```

Out[12]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Speci
0	1	5.1	3.5	1.4	0.2	Ir setc
1	2	4.9	3.0	1.4	0.2	Ir setc
2	3	4.7	3.2	1.3	0.2	Ir setc
3	4	4.6	3.1	1.5	0.2	Ir setc
4	5	5.0	3.6	1.4	0.2	Ir setc
...	...	...	...	...	...	
145	146	6.7	3.0	5.2	2.3	Ir virgini
146	147	6.3	2.5	5.0	1.9	Ir virgini
147	148	6.5	3.0	5.2	2.0	Ir virgini
148	149	6.2	3.4	5.4	2.3	Ir virgini
149	150	5.9	3.0	5.1	1.8	Ir virgini

150 rows × 6 columns

In [14]:
```python
data.head()
```

Out[14]:

	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

In [16]:
```python
data.tail()
```

Out[16]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Speci
145	146	6.7	3.0	5.2	2.3	Ir virgini
146	147	6.3	2.5	5.0	1.9	Ir virgini
147	148	6.5	3.0	5.2	2.0	Ir virgini
148	149	6.2	3.4	5.4	2.3	Ir virgini
149	150	5.9	3.0	5.1	1.8	Ir virgini

In [18]:
```python
len(data)
```

Out[18]: 150

In [20]:
```python
data.shape
```

Out[20]: (150, 6)

In [22]:
```python
data.columns
```

Out[22]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthC
m',
       'Species'],
      dtype='object')
```

In [24]:
```python
for i,col in enumerate(data.columns):
    print(f'Column number {1+i} is {col}')
```

```
Column number 1 is Id
Column number 2 is SepalLengthCm
Column number 3 is SepalWidthCm
Column number 4 is PetalLengthCm
Column number 5 is PetalWidthCm
Column number 6 is Species
```

In [26]:
```
data.dtypes
```

Out[26]:
```
Id                 int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [28]:
```
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 [30]:
```
data.describe()
```

Out[30]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthC
count	150.000000	150.000000	150.000000	150.000000	150.0000
mean	75.500000	5.843333	3.054000	3.758667	1.1986
std	43.445368	0.828066	0.433594	1.764420	0.7631
min	1.000000	4.300000	2.000000	1.000000	0.1000
25%	38.250000	5.100000	2.800000	1.600000	0.3000
50%	75.500000	5.800000	3.000000	4.350000	1.3000
75%	112.750000	6.400000	3.300000	5.100000	1.8000
max	150.000000	7.900000	4.400000	6.900000	2.5000

In [32]:
```
data.isnull()
```

Out[32]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Spe
0	False	False	False	False	False	F
1	False	False	False	False	False	F
2	False	False	False	False	False	F
3	False	False	False	False	False	F
4	False	False	False	False	False	F
...	...	...	...	...	...	
145	False	False	False	False	False	F
146	False	False	False	False	False	F
147	False	False	False	False	False	F
148	False	False	False	False	False	F
149	False	False	False	False	False	F

150 rows × 6 columns

In [34]:
```
data.isnull().sum()
```

Out[34]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [36]:
```
data.drop('Id', axis=1, inplace=True)
data.head()
```

Out[36]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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

K - Means Clustering

In [39]:
```
data.isna().sum()
```

Out[39]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [41]:
```
data.head()
```

Out[41]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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

In [43]:
```
data['Species'].value_counts()
```

Out[43]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [45]:
```
target_data = data.iloc[:,4]
target_data.head()
```

Out[45]:
```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

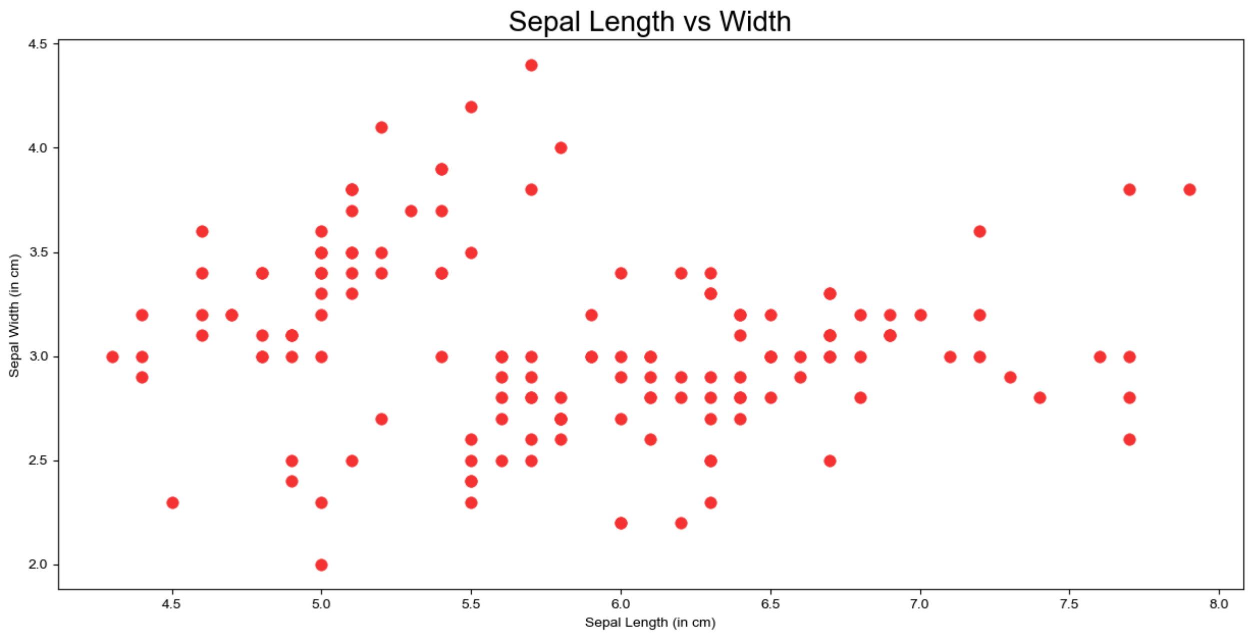
In [47]:
```
clustering_data = data.iloc[:,[0,1,2,3]]
clustering_data.head()
```

Out[47]:

	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

In [49]:
```
fig, ax = plt.subplots(figsize=(15,7))
```

```
sns.set(font_scale=1.5)
ax = sns.scatterplot(x=data['SepalLengthCm'],y=data['SepalWidthCm'], s=70, col
edgecolor='#f73434', linewidth=0.3)
ax.set_ylabel('Sepal Width (in cm)')
ax.set_xlabel('Sepal Length (in cm)')
plt.title('Sepal Length vs Width', fontsize = 20)
plt.show()
```

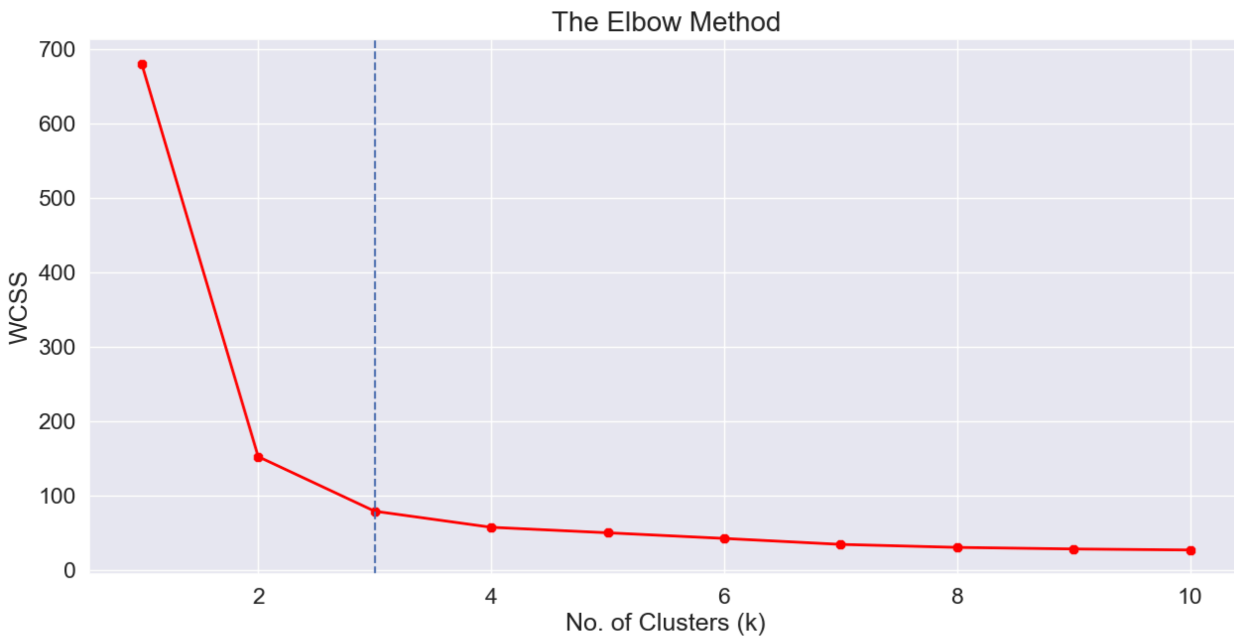


Determining No. of Clusters Required

In [52]:
```
from sklearn.cluster import KMeans
import numpy as np

wcss = []  # Within-Cluster Sum of Squares

for i in range(1, 11):
    km = KMeans(n_clusters=i, random_state=0)
    km.fit(clustering_data)
    wcss.append(km.inertia_)

print(np.array(wcss))
```

```
[680.8244     152.36870648  78.94506583  57.31787321  49.91714056
  42.31252156  34.31116759  30.27495426  28.1512578   26.77847392]
```

In [54]:
```
fig, ax = plt.subplots(figsize=(15,7))
ax = plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")
plt.axvline(x=3, ls='--')
plt.ylabel('WCSS')
plt.xlabel('No. of Clusters (k)')
plt.title('The Elbow Method', fontsize = 20)
plt.show()
```



Clustering

In [57]:
```
from sklearn.cluster import KMeans
kms = KMeans(n_clusters=3, init='k-means++')
kms.fit(clustering_data)
KMeans(n_clusters=3)
```

Out[57]:
```
▾ KMeans
KMeans(n_clusters=3)
```

In [59]:
```
clusters = clustering_data.copy()
clusters['Cluster_Prediction'] = kms.fit_predict(clustering_data)
clusters.head()
```

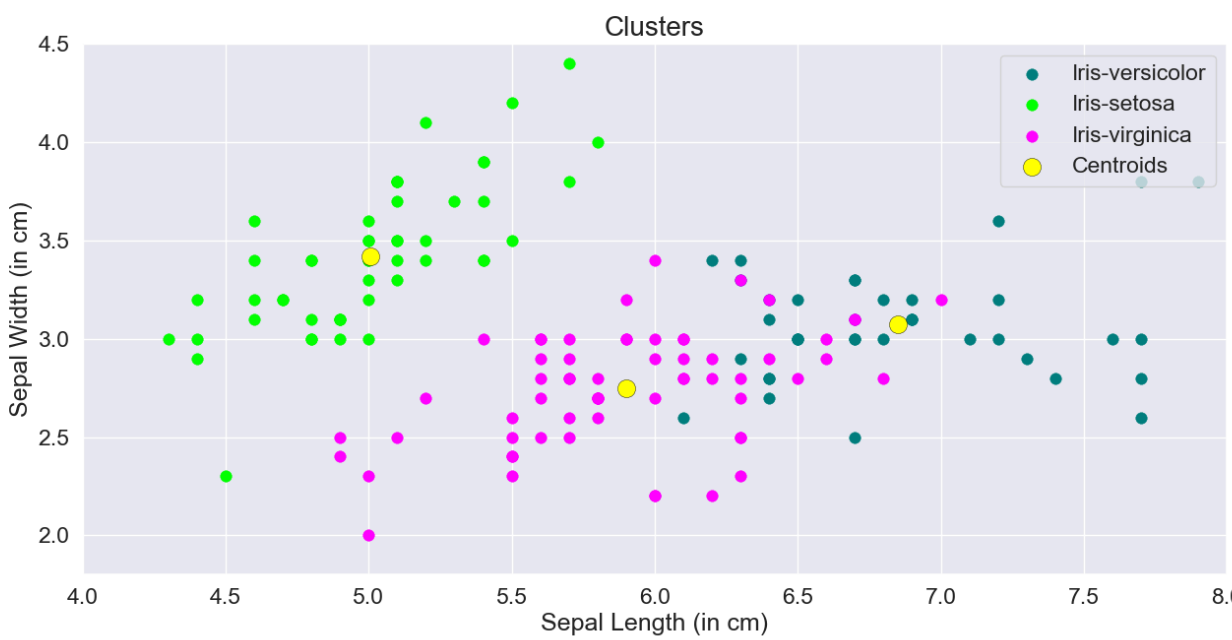
Out[59]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Cluster_Predi
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 [61]:
```
kms.cluster_centers_
```

Out[61]:
```
array([[6.85      , 3.07368421, 5.74210526, 2.07105263],
       [5.006     , 3.418     , 1.464     , 0.244     ],
       [5.9016129 , 2.7483871 , 4.39354839, 1.43387097]])
```

In [63]:
```
fig, ax = plt.subplots(figsize=(15,7))
plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 0]['SepalLengthCm'],
    y=clusters[clusters['Cluster_Prediction'] == 0]['SepalWidthCm'],
    s=70,edgecolor='teal', linewidth=0.3, c='teal', label='Iris-versicolor')
plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 1]['SepalLengthCm'],
    y=clusters[clusters['Cluster_Prediction'] == 1]['SepalWidthCm'],
    s=70,edgecolor='lime', linewidth=0.3, c='lime', label='Iris-setosa')
plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 2]['SepalLengthCm'],
    y=clusters[clusters['Cluster_Prediction'] == 2]['SepalWidthCm'],
    s=70,edgecolor='magenta', linewidth=0.3, c='magenta', label='Iris-virginic
plt.scatter(x=kms.cluster_centers_[:, 0], y=kms.cluster_centers_[:, 1], s = 17
'Centroids',edgecolor='black', linewidth=0.3)
plt.legend(loc='upper right')
plt.xlim(4,8)
plt.ylim(1.8,4.5)
ax.set_ylabel('Sepal Width (in cm)')
ax.set_xlabel('Sepal Length (in cm)')
plt.title('Clusters', fontsize = 20)
plt.show()
```



In [ ]: