internship-task-2

May 4, 2022

1 internship-task-2

Use the "Run" button to execute the code.

[45]:
```
!pip install jovian --upgrade --quiet
```

[46]:
```
import jovian
```

[47]:
```
# Execute this to save new versions of the notebook
jovian.commit(project="internship-task-2")
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "parishabhatia12/internship-task-2" on
https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/parishabhatia12/internship-
task-2
```

[47]:
```
'https://jovian.ai/parishabhatia12/internship-task-2'
```

[30]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
import pandas as pd
```

[33]:
```
iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data,columns = iris.feature_names)
iris_df
```

[33]:

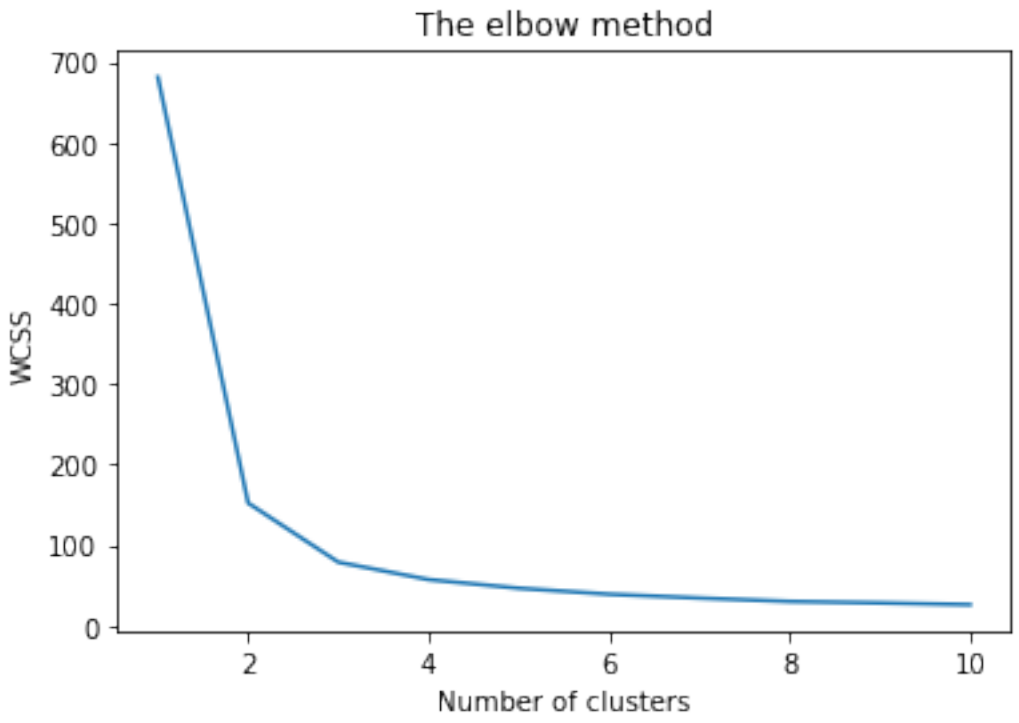
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
..	…	…	…	…
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0

```
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

[150 rows x 4 columns]
```

[40]:
```
x = iris_df.iloc[:,[0,1,2,3]].values
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters = i,init ='k-means++',
                max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show
```

[40]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



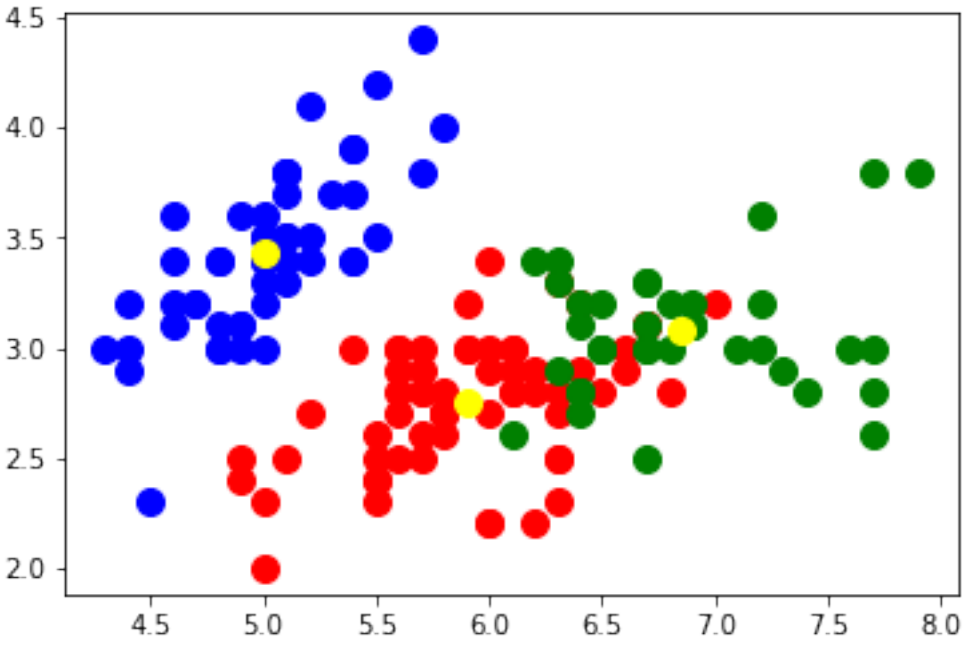
1.0.1 Optimum number of cluster is where the elbow occurs. In this case it is three.

[43]:
```
kmeans = KMeans(n_clusters = 3,init = 'k-means++',
                max_iter = 300, n_init = 10,random_state = 0)
y_kmeans = kmeans.fit_predict(x)
```

[44]:
```
plt.scatter(x[y_kmeans == 0,0], x[y_means == 0,1],
            s = 100, c = 'red', label = 'Iris-setosa')
plt.scatter(x[y_kmeans == 1,0], x[y_means == 1,1],
            s = 100, c = 'blue', label = 'Iris-versicolor')
plt.scatter(x[y_kmeans == 2,0], x[y_means == 2,1],
            s = 100, c = 'green', label = 'Iris-virginica')

plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],
            s=100, c='yellow', label='Centroids')
plt.legend
```

[44]:
```
<function matplotlib.pyplot.legend(*args, **kwargs)>
```



[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: