ml4

November 3, 2024

Clustering Analysis: Implement K-Means clustering on Iris.csv dataset. Determine the number of clusters using the elbow method

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
df=pd.read_csv(r"C:\Users\dell\Desktop\DMV and ML\ML Datasets\Iris.csv")
```

[3]:
```
df.head()
```

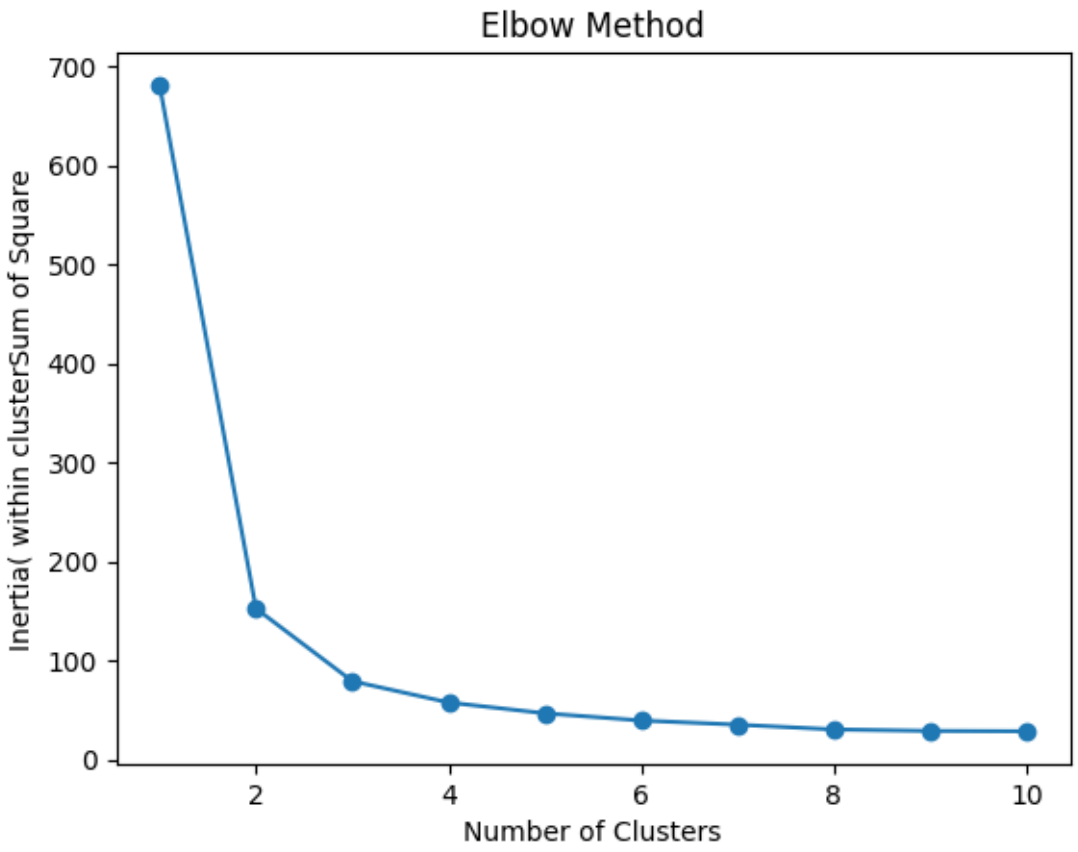
[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Class Label
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

[4]:
```
x=df.iloc[:,[1,2,3,4]]
# x=df.drop("Species",axis=1)   another way
```

[10]:
```
inertia=[]   #sum of squared distances to the nearest cluster center
for i in range(1,11):
    model= KMeans(n_clusters=i,max_iter=300,random_state=42)
    model.fit(x)
    inertia.append(model.inertia_)
```

[6]:
```
# plt.figure(figsize=(8,5))
plt.plot(range(1,11), inertia, marker="o")
plt.xlabel("Number of Clusters")
plt.ylabel("Inertia( within cluster Sum of Square")
plt.title("Elbow Method")
plt.show()
```



[7]:
```
optimal_clusters=3
model = KMeans(n_clusters=optimal_clusters,random_state=42)
cluster_label = model.fit_predict(x)
df["Cluster"] = cluster_label
```

[8]:
```
df
```

[8]:

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

```
        Class Label  Cluster
0       Iris-setosa        1
1       Iris-setosa        1
2       Iris-setosa        1
3       Iris-setosa        1
4       Iris-setosa        1
..              …      …
145  Iris-virginica        0
146  Iris-virginica        2
147  Iris-virginica        0
148  Iris-virginica        0
149  Iris-virginica        2

[150 rows x 7 columns]
```

[9]:
```
cluster_label
```

[9]:
```
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, 1, 0, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0,
       0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0, 0,
       0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2])
```

K-Means is an unsupervised machine learning algorithm commonly used for clustering. It partitions the data into 'k' distinct clusters by minimizing the variance within each cluster. The algorithm initializes 'k' centroids, assigns each data point to the nearest centroid, and recalculates the centroids based on the points assigned to each cluster. This process repeats until the centroids no longer change significantly, resulting in a stable clustering.

Inertia is the sum of squared distances of each data point to its closest cluster center.

The goal is to identify the point where the rate of decrease in WCSS sharply changes, indicating that adding more clusters (beyond this point) yields diminishing returns. This "elbow" point suggests the optimal number of clusters.