Ex.no:10 **DHAMODARAN.B**

Date:06.10.2020 **1832016**

K-Means Clustering

Problem Statement:

Though the Indian GDP had a heavy fall in the year 2020, the industrial sectors such as the agricultural, forestry and fishery had a significant rise of about 3% in their contribution to GDP. Fishery contributions to the Indian GDP had remained consistent over a long period of time. Thus, understanding the behavioral nature of the fishes and their health can help us increase the productivity. Clusters can help us understand the behavioral nature of the fishes by grouping them into clusters. This will also help us to understand how similar the different species are and understand the health conditions of the fishes. Our Motto lies at building a K-Means clustering model which can be used to understand the nature of the fishes and help us take necessary actions.

Problem Description:

The different variables that provide information about the fish health are the 'weight', 'length1', 'length2', 'lenght3', 'height' and 'width'. All the features were continuous in nature. The dataset had about 159 data instances.

Initially, the data was checked for null values. Then, the data are scaled using the standard scaling technique. The K-Means clustering model is implemented using the scikit's learn' s KMeans class. Later the most appropriate value for k is chose using the elbow method. The performance of the model are also evaluated under different metrics.

Sample Dataset:

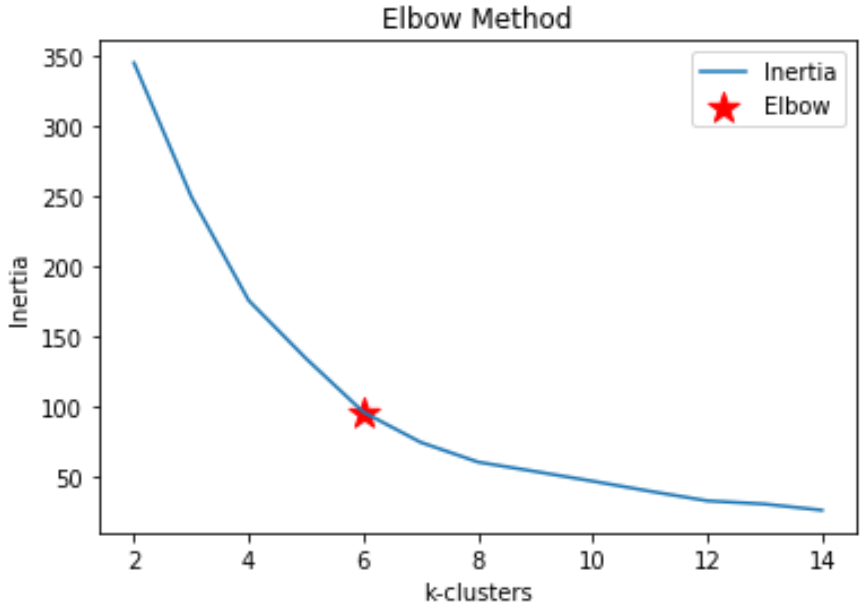
	Weight	Length1	Length2	Length3	Height	Width
0	242.0	23.2	25.4	30.0	11.5200	4.0200
1	290.0	24.0	26.3	31.2	12.4800	4.3056
2	340.0	23.9	26.5	31.1	12.3778	4.6961
3	363.0	26.3	29.0	33.5	12.7300	4.4555
4	430.0	26.5	29.0	34.0	12.4440	5.1340
5	450.0	26.8	29.7	34.7	13.6024	4.9274
6	500.0	26.8	29.7	34.5	14.1795	5.2785
7	390.0	27.6	30.0	35.0	12.6700	4.6900
8	450.0	27.6	30.0	35.1	14.0049	4.8438
9	500.0	28.5	30.7	36.2	14.2266	4.9594
10	475.0	28.4	31.0	36.2	14.2628	5.1042
11	500.0	28.7	31.0	36.2	14.3714	4.8146
12	500.0	29.1	31.5	36.4	13.7592	4.3680

Code:

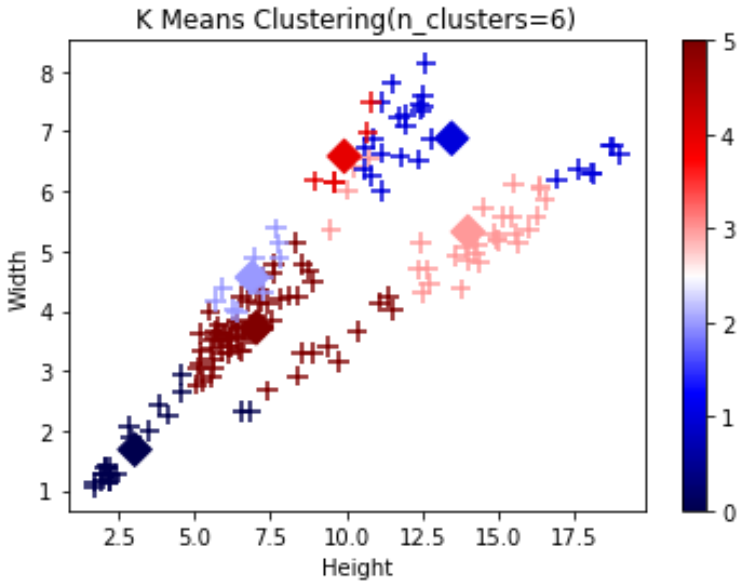
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
dataset=pd.read_csv('../7.KNN Classifier/Fish.csv')
dataframe=dataset.iloc[:,1:].copy()
dataframe
dataframe.isnull().sum()
dataframe.plot.scatter(x='Height',y='Width',title = 'Height VS Width')
from sklearn.preprocessing import StandardScaler
from kneed import KneeLocator
scaler=StandardScaler()
x = scaler.fit_transform(dataframe)
from sklearn.cluster import KMeans
inertia={}
for clusters in range(2,15):
    kmeans_model = KMeans(n_clusters=clusters,n_jobs=-1)
    kmeans_model.fit(x)
    inertia[clusters]=(kmeans_model.inertia_)
inertia
kl=KneeLocator(range(2,15),list(inertia.values()),direction='decreasing',curve='convex')
elbow = kl.elbow
plt.plot(inertia.keys(),inertia.values(),label='Inertia')
plt.scatter(elbow,inertia[elbow],marker = '*',s=200,c='r',label='Elbow')
plt.legend()
plt.xlabel('Clusters')
plt.title('Elbow Method')
plt.ylabel('Inertia')
kmeans_model=KMeans(n_clusters=elbow,n_jobs=-1)
kmeans_model.fit(x)
c_labels = kmeans_model.predict(x)
c_centroids = scaler.inverse_transform(kmeans_model.cluster_centers_)
```

```
clusters = np.append(dataframe,c_labels.reshape(-1,1),axis=1)

clusters = pd.DataFrame(data=clusters)

columns=list(dataframe.columns)+['c_labels']

clusters.columns=columns

clusters.columns

fig,axes = plt.subplots(1,1)

fig =
axes.scatter(x=dataframe['Height'],y=dataframe['Width'],marker='+',cmap='seismic',c=clusters['c_labels'],s=90)

axes.scatter(c_centroids[:,4],c_centroids[:,5],s=120,marker='D',cmap='seismic',c=(range(len(c_centroids))))

axes.set_title('K Means Clustering(n_clusters={})'.format(elbow))

axes.set_xlabel('Height')

axes.set_ylabel('Width')

plt.colorbar(fig,ax=axes)

plt.show()

from sklearn.metrics import silhouette_score

from sklearn.metrics import calinski_harabaz_score

print("Silhouette score : ",silhouette_score(x,c_labels))

print("Calinski Harabaz Score : ",calinski_harabaz_score(x,c_labels))
```

OUTPUT:



```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=6, n_init=10, n_jobs=-1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```




```
Silhouette score :  0.4977064173652521

Calinski Harabaz Score :  271.7904954372581
```

Inference:

The inertia i.e. within cluster sum of squares decreases as the k value increases The elbow for the decreasing convex inertia curve occurs at k=6. This can be interpreted as that the inertia is optimal when number of clusters are 6. The model with k=6 clusters has an inertia of 96.53. Silhouette score is the mean Silhouette coefficient of all the samples. It explains how similar an instance is to its own cluster compared to other clusters. Silhouette score of 0.49 is good as it doesn't assign any data points to a wrong cluster and also it doesn't produce any overlapping clusters. Calinski Harabaz Score is the ratio of the between-clusters dispersion and the within-cluster dispersion. Calinski Harabaz score of 271.79 states that the clusters are dense and are well separated.

Conclusion:

The optimal number of clusters(k) in k-Means clustering method is 6 which is determined using the elbow method considering the intertia. The K-Means(k=6) clustering model had a Silhouette score of 0.49 and Calinski Harabaz score of 271.79 which indicates that the clusters are dense and also well separated from each other. Thus, the built K-Means(k=6) clustering model can be used to group the fishes based on their health conditions and take the necessary actions to increase the productivity.