Name : Aniket Narbariya

UID : 2019130043

Subject : AIML

Experiment : 5

Aim : To train and test a machine learning model using K-Means algorithm

Theory:

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. It groups the unlabeled dataset into different clusters. Here K defines the number of pre- defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

How does K-Means algorithm work?

The working of the K-Means algorithm is explained in the below steps:
Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

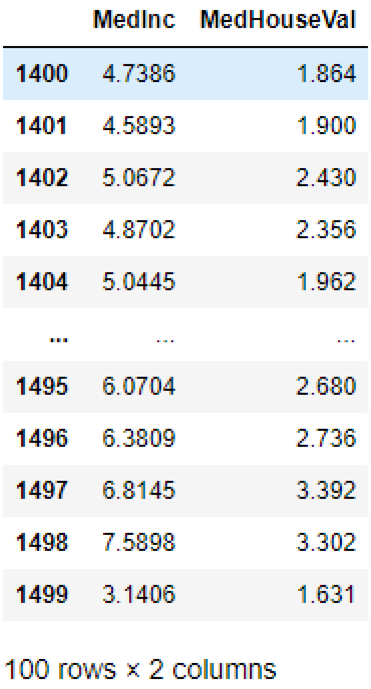
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

Program :

```
import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

from sklearn.cluster import KMeans


df = pd.read_csv('housing.csv')

df = df.iloc[1400:1500]

# df

data_for_clustering = df[["MedInc","MedHouseVal"]]
```

```
data_for_clustering
```

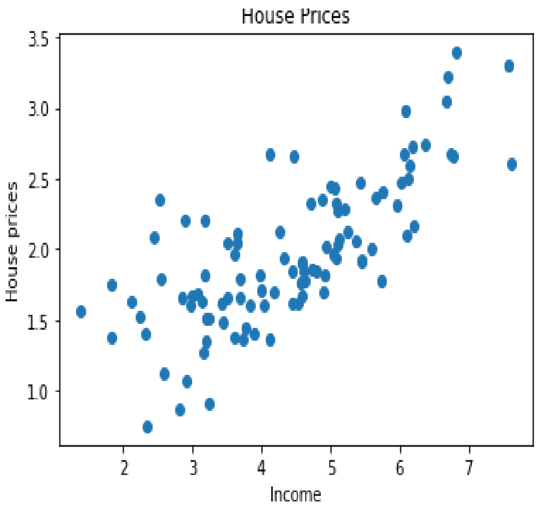


	MedInc	MedHouseVal
1400	4.7386	1.864
1401	4.5893	1.900
1402	5.0672	2.430
1403	4.8702	2.356
1404	5.0445	1.962
...	...	...
1495	6.0704	2.680
1496	6.3809	2.736
1497	6.8145	3.392
1498	7.5898	3.302
1499	3.1406	1.631

100 rows × 2 columns

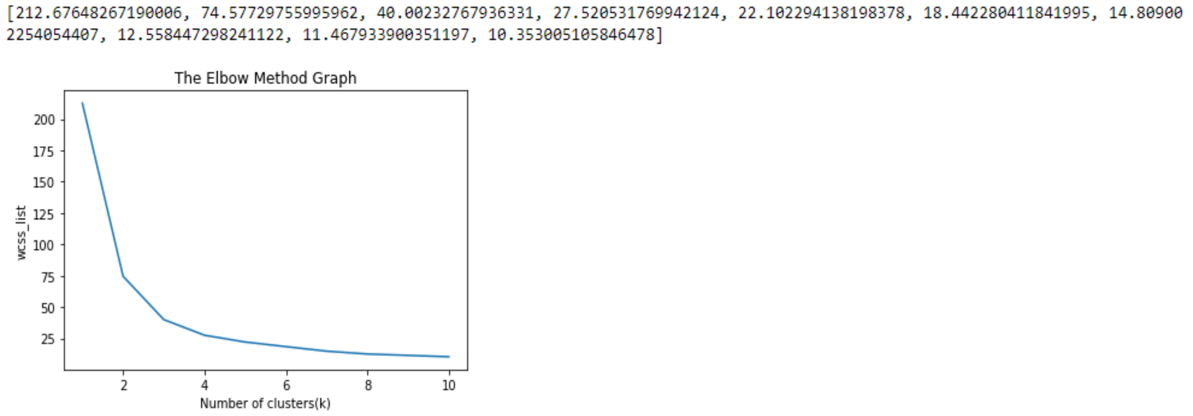
```
x = data_for_clustering.values
x
```

```
plt.scatter(data_for_clustering.MedInc.to_list() , data_for_clustering.MedHouseVal.to_list())
plt.title("House Prices")
plt.xlabel("Income")
plt.ylabel("House prices")
plt.show()
```



```
def get_wcss(X):
    wcss_list= []
    for i in range(1, 11):
        kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
        kmeans.fit(X)
        wcss_list.append(kmeans.inertia_)

    return wcss_list
```

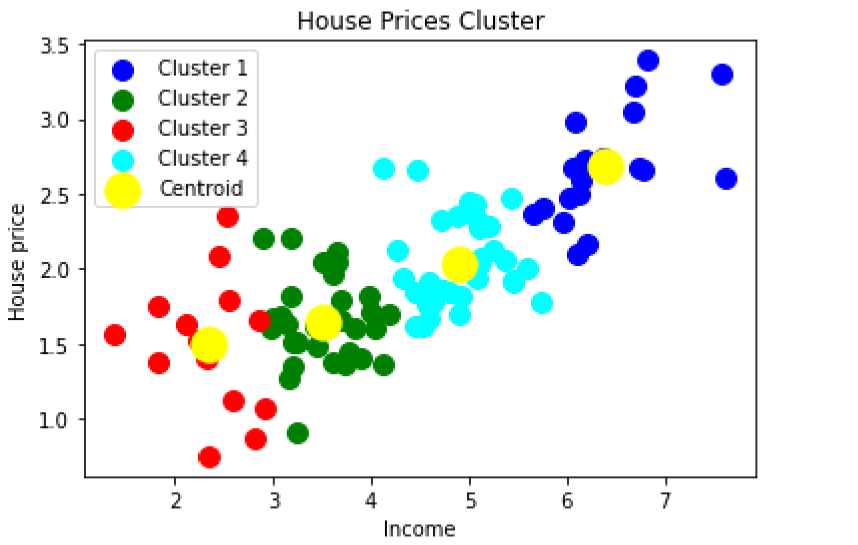
```
wcss = get_wcss(x)
print(wcss)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method Graph')
plt.xlabel('Number of clusters(k)')
plt.ylabel('wcss_list')
plt.show()
```



```
[212.67648267190006, 74.57729755995962, 40.00232767936331, 27.520531769942124, 22.102294138198378, 18.442280411841995, 14.80900
2254054407, 12.558447298241122, 11.467933900351197, 10.353005105846478]
```

```
def clustering_kmeans(X,k):
kmeans = KMeans(n_clusters=k, init='k-means++', random_state= 42)
y= kmeans.fit_predict(X)
return kmeans,y
```

```
# Using the Elbow method the optimal value of cluster(k) is 4 for the given dataset
k_means, y = clustering_kmeans(x, 4)

plt.scatter(x[y == 0, 0], x[y == 0, 1], s = 100, c = 'blue', label = 'Cluster 1') #for first cluster
plt.scatter(x[y == 1, 0], x[y == 1, 1], s = 100, c = 'green', label = 'Cluster 2') #for second cluster
plt.scatter(x[y== 2, 0], x[y == 2, 1], s = 100, c = 'red', label = 'Cluster 3') #for third cluster
plt.scatter(k_means.cluster_centers_[:, 0], k_means.cluster_centers_[:, 1], s = 300, c = 'yellow',
label = 'Centroid')
plt.title('House Prices Cluster')
plt.xlabel('Income')
plt.ylabel('House price')
plt.legend()
plt.show()
```



Conclusion:

From above experiment, I learned about basics of K-Means algorithm. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

It determines the best value for K center points or centroids by an iterative process and assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm. Accuracy of algorithm varies with number of clusters selected.