Author Name : SRAVAN KUMAR VASAM
R & D Project : Machine Learning Clustering models K-means and Hierarchical Clustering
Technologies : R version 4.0.2, Rstudio, Linux
Year of submission : November, 2020

Data source : Banking/credit limit data

Aim : Based on the annual income and spending score we discover the group of clusters, segmenting the potential, non potential and sensitive customers. By applying two k-means and HC algorithms.

Clustering machine learning model

Important points to be noted

1) Clustering is similar to classification, but the basis is different.
2) In Clustering you don't know what you are looking for, and you are trying to identify some segments or clusters in your data.
3) When you use clustering algorithms on your dataset, unexpected things can suddenly pop up like structures, clusters and groupings you would have never thought of otherwise.
4) Using the elbow method to find the optimal number of clusters

Following machine learning clustering models implementing.

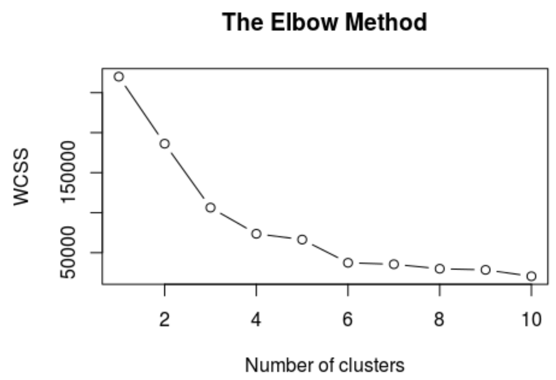
(1) K-Means Clustering, library(cluster), function :clusplot
(2) Hierarchical Clustering, library(cluster), functions : visualisation-clusplot, hclust

1. K-Means Clustering
   pros : Simple to understand, easily adaptable, works well on small or large datasets, fast, efficient and performant
   cons : Need to choose the number of clusters
   formula :
   --Fitting K-Means to the dataset
   set.seed(29)
   kmeans = kmeans(x = dataset, centers = 5)

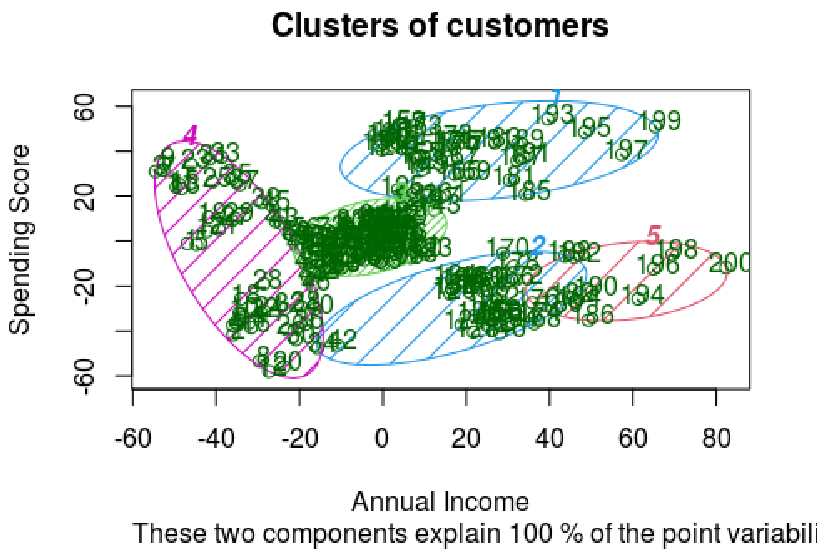
K-means clustering model variables output

Data	
dataset	200 obs. of 2 variables
kmeans	List of 9
Values	
i	10L
wcss	num [1:10] 269981 186207 106348 73680 66465 ...
y_kmeans	int [1:200] 4 4 4 4 4 4 4 4 4 4 ...

Elbow method



K-means clustering Rlot



2. Hierarchical Clustering

   pros : The optimal number of clusters can be obtained by the model itself, practical visualisation with the dendrogram.

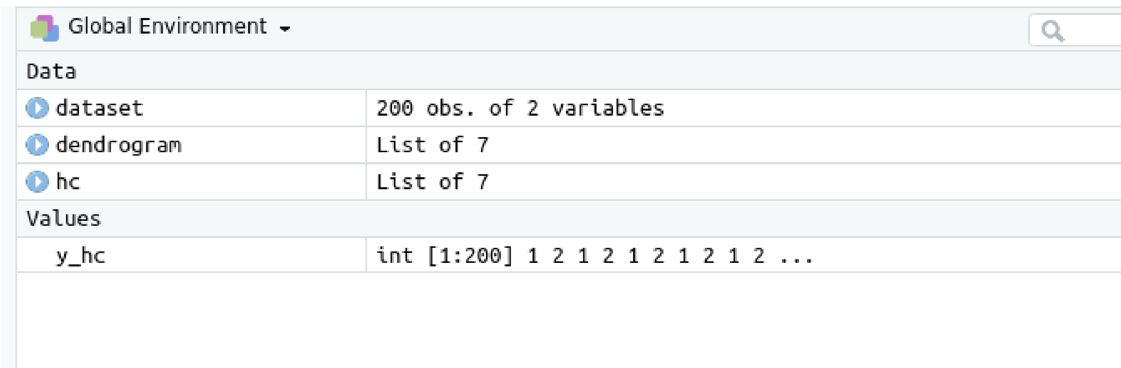
   Cons : Not appropriate for large datasets

   formula :

   dendrogram = hclust(d = dist(dataset, method = 'euclidean'), method = 'ward.D')

   plot(dendrogram, main = paste('Dendrogram'), xlab = 'Customers', ylab = 'Euclidean distances')

Hierarchical clustering variable output:



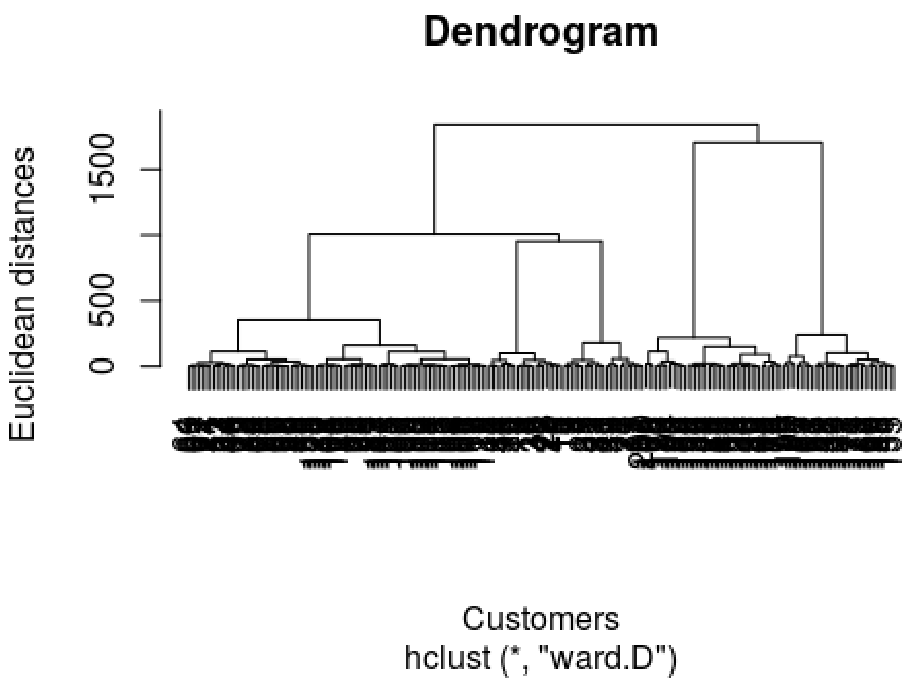
Global Environment

Data	
dataset	200 obs. of 2 variables
dendrogram	List of 7
hc	List of 7
Values	
y_hc	int [1:200] 1 2 1 2 1 2 1 2 1 2 ...

Hierarchical clustering variable output:

```
+       ylab = 'Euclidean distances')
> y_hc
  [1] 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
 [31] 1 2 1 2 1 2 1 2 1 2 1 2 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [61] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [91] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[121] 3 3 3 4 3 4 3 4 5 4 5 4 3 4 5 4 5 4 5 4 5 4 3 4 5 4 3 4 5 4
[151] 5 4 5 4 5 4 5 4 5 4 3 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4
[181] 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4
> 
```

Hierarchical clustering



super market mall data source

	CustomerID	Genre	Age	Annual.Income..k..	Spending.Score..1.100.
1	1	Male	19	15	39
2	2	Male	21	15	81
3	3	Female	20	16	6
4	4	Female	23	16	77
5	5	Female	31	17	40
6	6	Female	22	17	76
7	7	Female	35	18	6
8	8	Female	23	18	94
9	9	Male	64	19	3
10	10	Female	30	19	72
11	11	Male	67	19	14
12	12	Female	35	19	99

input data source

	Annual.Income..k..	Spending.Score..1.100.
1	15	39
2	15	81
3	16	6
4	16	77
5	17	40
6	17	76
7	18	6
8	18	94
9	19	3
10	19	72
11	19	14
12	19	99
13	20	15
14	20	77
15	20	13
16	20	79
17	21	35
18	21	66
19	23	29
20	23	98
21	24	35
22	24	73