American International University - Bangladesh (AIUB)

INTRODUCTION TO DATA SCIENCE [E]

Name: Shanto Kumar Basak

ID: 20-42945-1

Faculty Name: Tohedul Islam

Date of Submission: 30th April 2023

Final Term Project (Applying K-means)

Introduction: The straightforward and widely used unsupervised machine learning approach K-means clustering. Unsupervised algorithms often draw conclusions from datasets using just the input vectors and no knowledge of the known, or labeled, results. Household Living Cost dataset collected from https://www.stats.govt.nz/large-datasets/csv-files-for-download/ this site.

1) Observing the Dataset

mydata <- read.csv("D:/Shanto IDS Project/Household-living - costs.csv",header=TRUE,sep=",")

mydata

```
> mydata <- read.csv("D:/Shanto IDS Project/Household-living-costs.csv",header=TRUE,sep=",")
> mydata
   year tot_hhs     own own_wm own_prop own_wm_prop prop_hhs  age size income expenditure eqv_income eqv_exp
1  2008 1560859 1087580 574406     69.7        36.8    100.0 35.9  2.7  46704       42394      26869   25132
2  2008  185965   71256  39405     38.3        21.2     11.9 29.9  2.6  23404       25270      14258   15824
3  2008  312376  191470  48424     61.3        15.5     20.0 40.0  2.3  16747       21145      13402   14408
4  2008  312333  196203  84171     62.8        26.9     20.0 34.7  2.8  31308       29855      18917   18266
5  2008  312240  217657 141318     69.7        45.3     20.0 31.5  3.0  49106       46561      26870   24672
6  2008  312336  229014 147658     73.3        47.3     20.0 35.3  2.6  61674       52776      36691   31958
7  2008  311574  253235 152835     81.3        49.1     20.0 39.3  2.5  96861       72822      55637   42932
8  2008  312761  194358  49448     62.1        15.8     20.0 38.7  2.5  23680       16413      15190   11015
9  2008  311973  206342  86390     66.1        27.7     20.0 36.1  2.7  34155       29085      20357   18121
10 2008  311840  194361 108065     62.3        34.7     20.0 33.0  2.8  49771       42662      27203   25132
11 2008  312257  231612 149007     74.2        47.7     20.0 35.1  2.7  60863       59015      34547   34167
12 2008  312028  260907 181496     83.6        58.2     20.0 36.7  2.5  77434       89053      46269   51550
13 2008  253018  119963  77076     47.4        30.5     16.2 28.9  3.2  42885       35312      23096   19797
14 2008  300243  263054  15406     87.6         5.1     19.2 70.3  1.6  22367       21538      17203   17211
15 2011 1607228 1048164 523698     65.2        32.6    100.0 36.3  2.6  53103       46098      30833   27335
16 2011  197237   56665  27129     28.7        13.8     12.3 28.0  2.7  25902       27605      16097   16685
17 2011  321848  166355  49952     51.7        15.5     20.0 36.3  2.4  19787       24224      15414   16221
18 2011  321751  187275  77561     58.2        24.1     20.0 35.0  2.9  37370       34200      21998   20586
19 2011  321372  204957 119746     63.8        37.3     20.0 33.4  2.9  54894       49431      30833   28130
20 2011  321507  226916 133454     70.6        41.5     20.0 36.8  2.6  69183       55569      42084   33019
21 2011  320751  262660 142986     81.9        44.6     20.0 40.9  2.4 106227       71815      63106   44712
22 2011  321611  173327  35941     53.9        11.2     20.0 37.3  2.6  27501       18877      17612   13077
23 2011  321894  179200  77025     55.7        23.9     20.0 35.1  2.7  38932       32790      22895   20168
24 2011  321367  211728 108496     65.9        33.8     20.0 35.3  2.8  56117       46651      32053   27335
```

2) Standarized the Data

mydata1 <- scale (mydata[,2:5])

head(mydata1)

set.seed(1)

```
> mydata1 <- scale (mydata[,2:5])
> head(mydata1)
        tot_hhs        own      own_wm    own_prop
[1,]  3.2889138  3.4319910  3.45779744  0.40899179
[2,] -0.6488650 -0.8289141 -0.70151966 -1.66426456
[3,] -0.2868163 -0.3249208 -0.63140226 -0.14563730
[4,] -0.2869395 -0.3050779 -0.35349043 -0.04659639
[5,] -0.2872058 -0.2151327  0.09079378  0.40899179
[6,] -0.2869309 -0.1675188  0.14008354  0.64668997
```

3) Clustering Result

kR<- pam(mydata1,k=4)

summary(kR)

```
> kR<- pam(mydata1,k=4)

> summary(kR)
Medoids:
     ID    tot_hhs         own      own_wm   own_prop
[1,] 29  3.5138743  3.44494990  3.28811317  0.1977045
[2,] 31 -0.2410687 -0.41014949 -0.59822885 -0.7530882
[3,] 33 -0.2433113 -0.20652974 -0.08486124  0.2373209
[4,] 35 -0.2423547  0.01023339  0.16808696  1.2739491
Clustering vector:
 [1] 1 2 2 3 3 3 4 3 3 3 3 4 2 4 1 2 2 2 3 3 4 2 2 3 3 4 2 4 1 2 2 2 3 3 4 2 2 3 3 4 2 4 1 2 2 2 3 3 4 2 2 3 3 4 2 4 1 2 2 2 3 3 4
[64] 2 2 3 3 4 2 4
Objective function:
    build      swap
0.4596551 0.4545288

Numerical information per cluster:
     size  max_diss   av_diss  diameter separation
[1,]    5 0.4362275 0.2620728 0.6989996 5.58139556
[2,]   27 2.1207135 0.6080233 2.6938439 0.05478447
[3,]   23 0.6419911 0.3199364 1.1508055 0.05478447
[4,]   15 1.1061074 0.4487657 1.3990203 0.46209104

Isolated clusters:
 L-clusters: character(0)
 L*-clusters: [1] 1

Average silhouette width per cluster:
[1] 0.9281013 0.2982502 0.5544197 0.4651107
Average silhouette width of total data set:
[1] 0.4631654

2415 dissimilarities, summarized :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.03288 0.68462 1.34750 2.02860 2.31620 8.29160
Metric :  euclidean
Number of objects : 70

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"  "clusinfo"   "silinfo"    "diss"       "call"
[10] "data"
```

4) Cluster Structure

mydata2 <-data.frame(mydata,kR$clustering)

head(mydata2)

set.seed(1)

kR2 <- kmeans(mydata1,4)

kR2$cluster

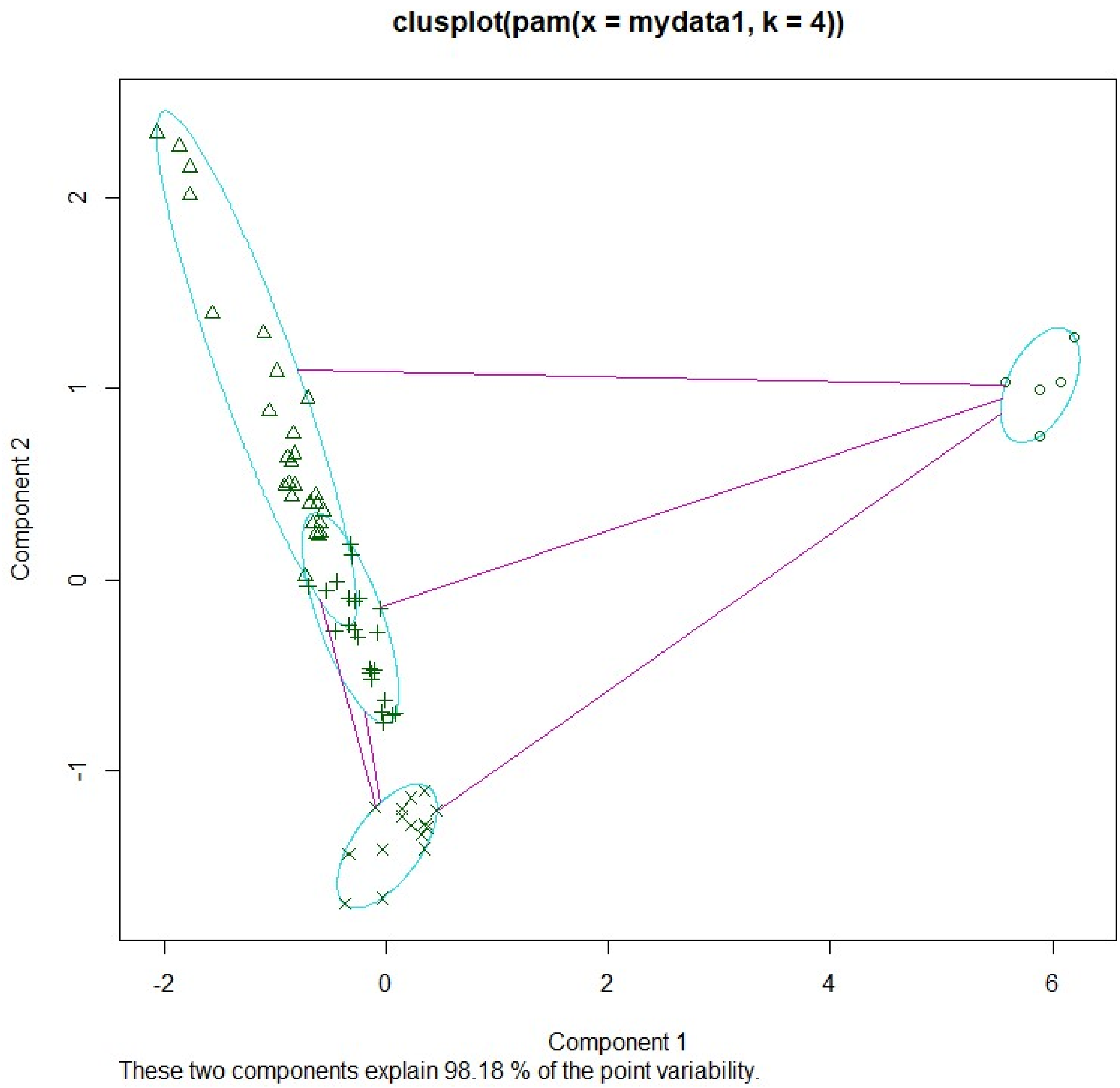
kR2$centers

```
> mydata2 <-data.frame(mydata,kR$clustering)
> head(mydata2)
  year tot_hhs     own own_wm own_prop own_wm_prop prop_hhs  age size income expenditure eqv_income eqv_exp kR.clustering
1 2008 1560859 1087580 574406     69.7        36.8    100.0 35.9  2.7  46704       42394      26869   25132             1
2 2008  185965   71256  39405     38.3        21.2     11.9 29.9  2.6  23404       25270      14258   15824             2
3 2008  312376  191470  48424     61.3        15.5     20.0 40.0  2.3  16747       21145      13402   14408             2
4 2008  312333  196203  84171     62.8        26.9     20.0 34.7  2.8  31308       29855      18917   18266             3
5 2008  312240  217657 141318     69.7        45.3     20.0 31.5  3.0  49106       46561      26870   24672             3
6 2008  312336  229014 147658     73.3        47.3     20.0 35.3  2.6  61674       52776      36691   31958             3
> set.seed(1)
> kR2 <- kmeans(mydata1,4)
> kR2$cluster
 [1] 3 2 4 4 1 1 1 4 4 4 1 1 4 1 3 2 4 4 4 1 1 4 4 4 1 1 2 1 3 2 4 4 4 1 1 4 4 4 1 1 4 1 3 2 4 4 4 1 1 4 4 4 1 1 4 1 3 2 4 4 4 4 1
[64] 4 4 4 4 1 4 1
> kR2$centers
     tot_hhs         own        own_wm   own_prop
1 -0.2306044 -0.02606027  0.0002860428  1.0257966
2 -0.6133278 -0.86571416 -0.7304611993 -2.1869805
3  3.5466827  3.45891920  3.3197177008  0.1871402
4 -0.2433983 -0.32785327 -0.3492196096 -0.3552267
> kR2$size
[1] 24  6  5 35
```

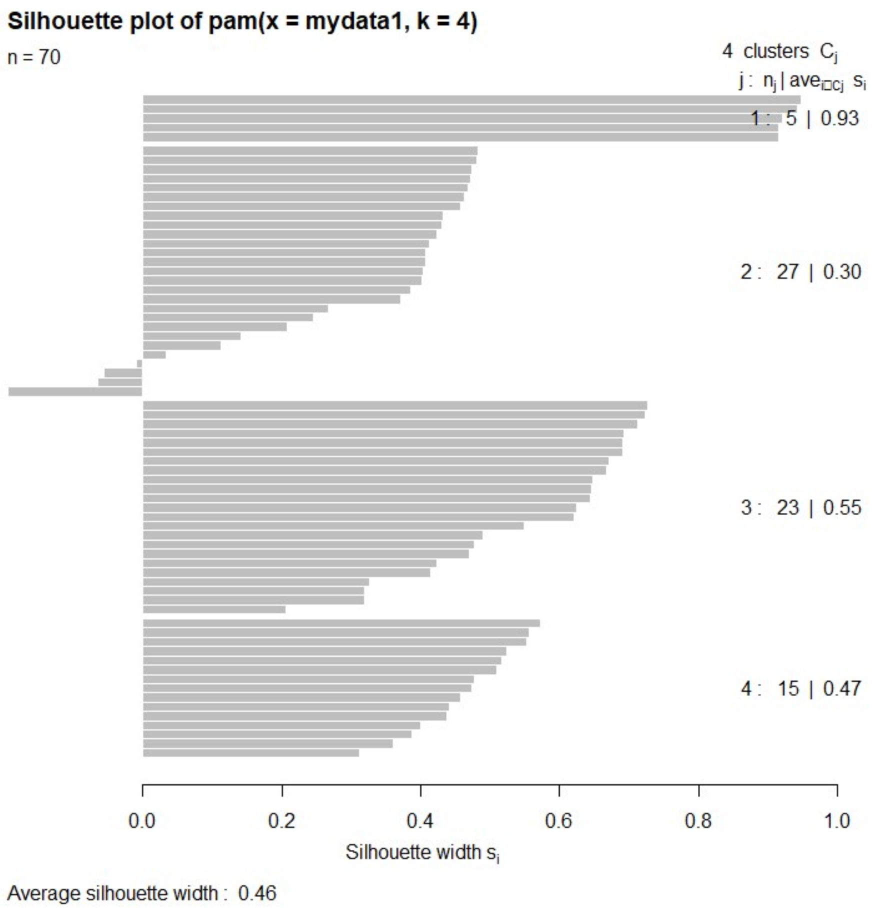
5) **Cluster and Silhouette Plot**

plot(kR)

Cluster Plot:



Silhouette Plot:



Conclusion: K-means clustering is an unsupervised machine learning method that is a component of a vast array of data approaches and operations in the field of data science. Data points are categorized using kmeans into unique, non-overlapping groupings. It is very easy to put into practice. Cluster generalization for various sizes and forms.

References:

[1] https://www.stats.govt.nz/large-datasets/csv-files-for-download/
[2] https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/
[3] https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
[4] https://www.geeksforgeeks.org/k-means-clustering-introduction/
[5] https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning
[6] https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/