# Customer Segmentation in R

## Customer Segmentation

- Problem: we don't know if we have different types of customers and how to approach them
- Goals: We want to understand better our customers; We want to have clear criteria to segment our customers
- Why? To perform specific actions to improve the customer experience

## Techniques

### K-means

Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K center).

## Case

We consider the dataset: Wholesale customers Data Set. Abreu, N. (2011).

This *dataset* has the following attributes:

- FRESH: annual spending (m.u.) on fresh products (Continuous);
- MILK: annual spending (m.u.) on milk products (Continuous);
- GROCERY: annual spending (m.u.) on grocery products (Continuous);
- FROZEN: annual spending (m.u.) on frozen products (Continuous)
- DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous)
- DELICATESSEN: annual spending (m.u.) on and delicatessen products (Continuous);
- CHANNEL: customers Channel - Horeca (Hotel/Restaurant/Café) or Retail channel (Nominal)
- REGION: customers Region of Lisbon, Oporto or Other (Nominal)

```
# install.packages("NbClust")
# Load packages
library(NbClust)
```

```
# Load data
data <- read.csv('Wholesale_customers_data.csv', header = T,sep=',')
```

```
# Review data structure
str(data)
```

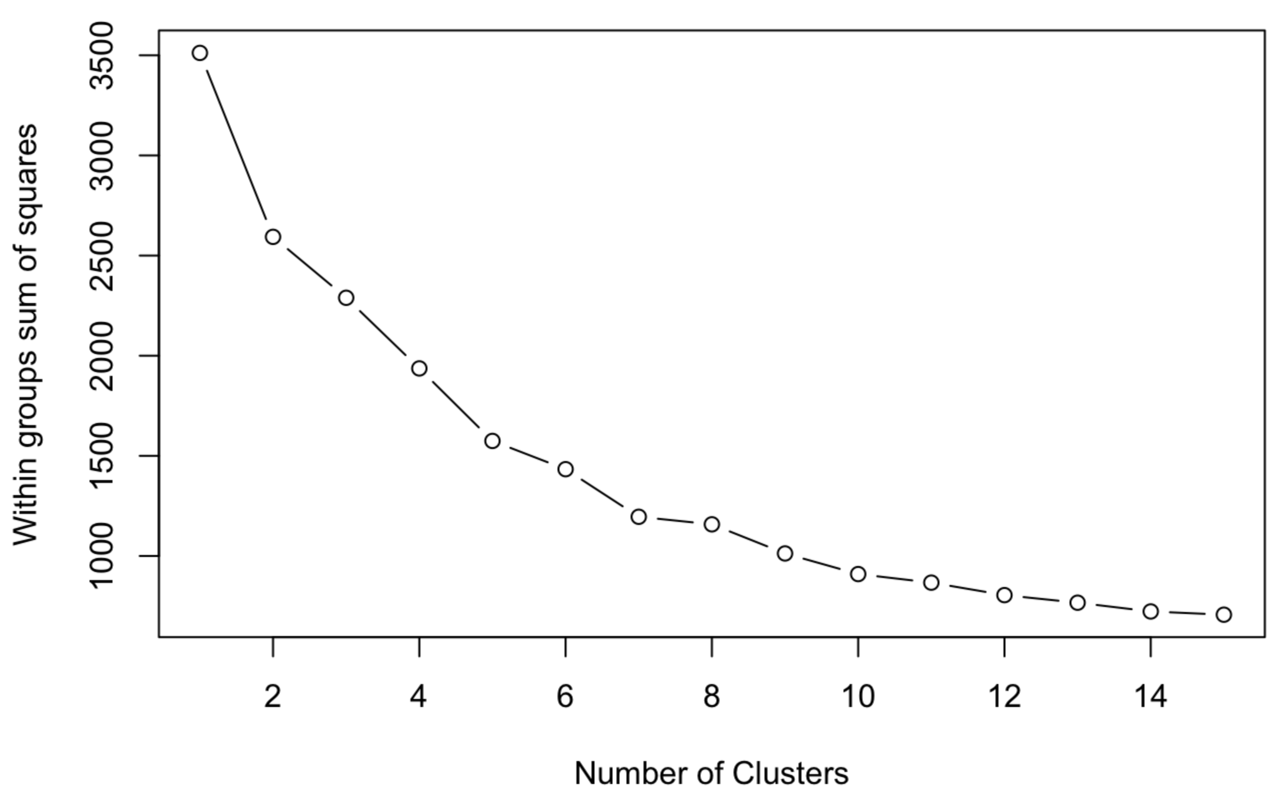
```
## 'data.frame':    440 obs. of  8 variables:
##  $ Channel         : int  2 2 2 1 2 2 2 2 1 2 ...
##  $ Region          : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Fresh           : int  12669 7057 6353 13265 22615 9413 12126 7579 5963 6006 ...
##  $ Milk            : int  9656 9810 8808 1196 5410 8259 3199 4956 3648 11093 ...
##  $ Grocery         : int  7561 9568 7684 4221 7198 5126 6975 9426 6192 18881 ...
##  $ Frozen          : int  214 1762 2405 6404 3915 666 480 1669 425 1159 ...
##  $ Detergents_Paper: int  2674 3293 3516 507 1777 1795 3140 3321 1716 7425 ...
##  $ Delicassen      : int  1338 1776 7844 1788 5185 1451 545 2566 750 2098 ...
```

```r
# Review data
summary(data)
```

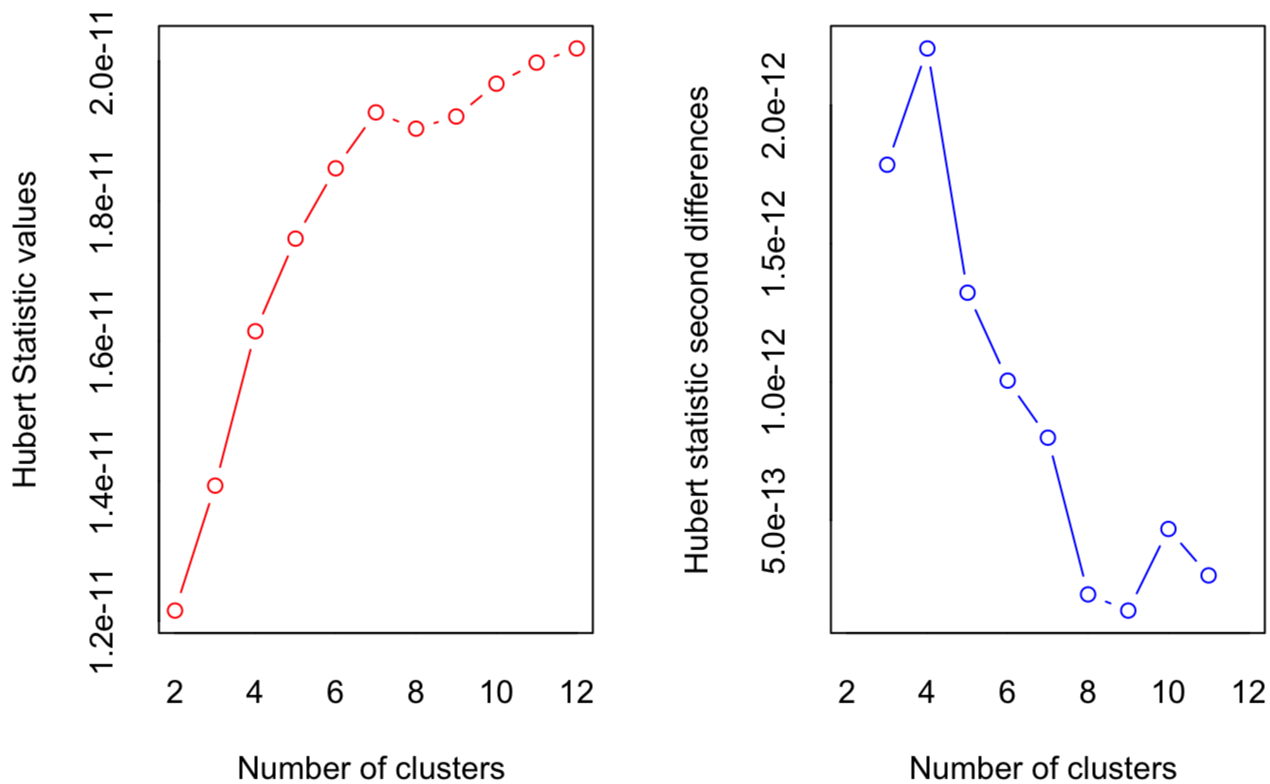
```
##     Channel          Region           Fresh              Milk
##  Min.   :1.000   Min.   :1.000   Min.   :     3   Min.   :   55
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3128   1st Qu.: 1533
##  Median :1.000   Median :3.000   Median :  8504   Median : 3627
##  Mean   :1.323   Mean   :2.543   Mean   : 12000   Mean   : 5796
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16934   3rd Qu.: 7190
##  Max.   :2.000   Max.   :3.000   Max.   :112151   Max.   :73498
##     Grocery          Frozen         Detergents_Paper    Delicassen
##  Min.   :    3   Min.   :   25.0   Min.   :    3.0   Min.   :    3.0
##  1st Qu.: 2153   1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2
##  Median : 4756   Median : 1526.0   Median :  816.5   Median :  965.5
##  Mean   : 7951   Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9
##  3rd Qu.:10656   3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2
##  Max.   :92780   Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
```

```r
# Scale data
testdata <- data
testdata <- scale(testdata)
```

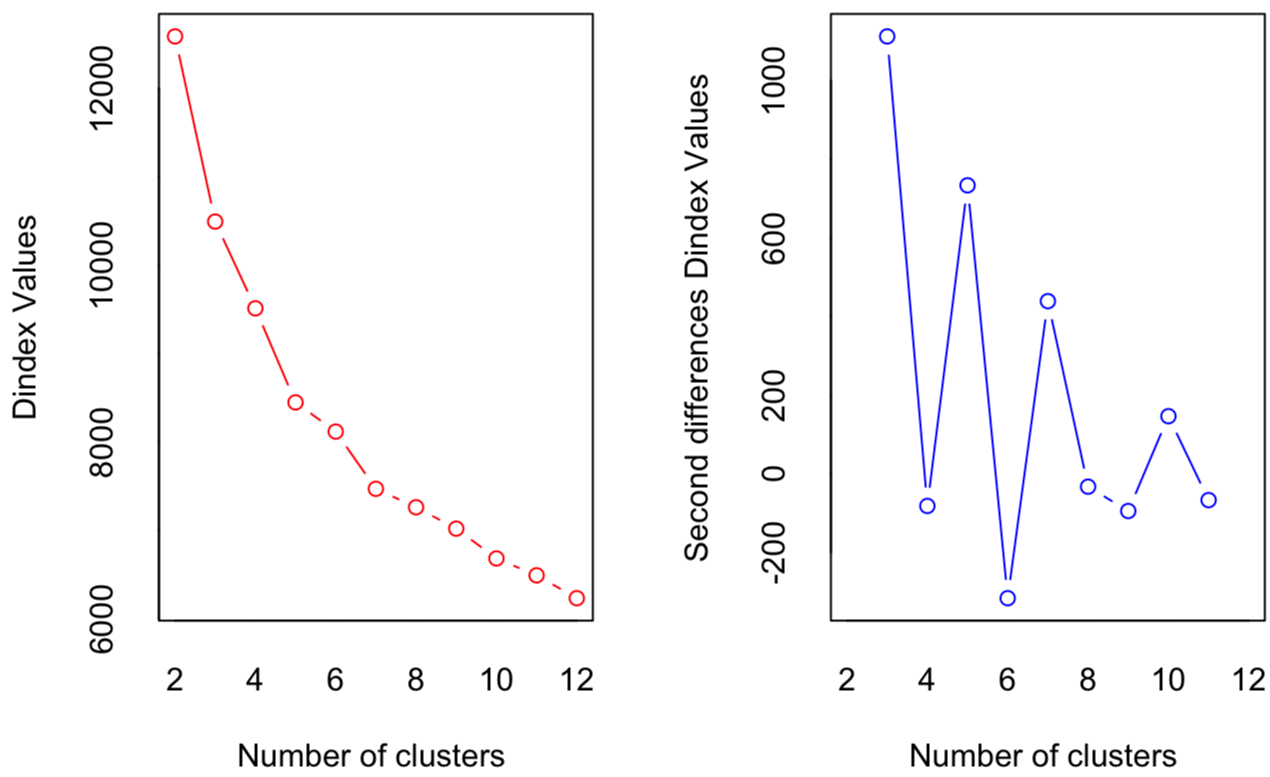
```r
# Determine number of clusters. Option 1: visual rule
wss <- (nrow(testdata)-1)*sum(apply(testdata,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(testdata,
                                     centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```



```
# Determine number of clusters. Option 2: more frequent optimal number
res <- NbClust(data, diss=NULL, distance = "euclidean", min.nc=2, max.nc=12,
               method = "kmeans", index = "all")
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hu
bert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Din
dex
##                 second differences plot) that corresponds to a significant increase of the valu
e of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 1 proposed 2 as the best number of clusters
## * 12 proposed 3 as the best number of clusters
## * 4 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 3 proposed 8 as the best number of clusters
## * 2 proposed 12 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

```
# More information
res$All.index
```

```
##        KL       CH Hartigan     CCC    Scott      Marriot       TrCovW
## 2  0.4228 139.3467 213.8768  3.6894 1792.821 1.795663e+64 6.362796e+20
## 3  2.6034 210.1526 104.2839  5.6955 2145.999 1.810542e+64 1.903724e+20
## 4  1.1518 207.8197  95.4616  3.2062 2624.545 1.084791e+64 1.390956e+20
## 5  2.1712 213.3661  54.7550  3.7014 2856.525 1.000448e+64 7.674669e+19
## 6  0.9702 202.6626  55.2290 -1.4515 3063.135 9.007938e+63 5.905103e+19
## 7  0.8032 199.1221  67.4358 -7.7967 3316.559 6.892582e+63 4.890789e+19
## 8  6.5121 206.4132  21.1059 -3.6800 3669.079 4.040332e+63 3.530384e+19
## 9  0.2840 191.6292  42.2033 -3.4975 3794.165 3.848200e+63 3.171553e+19
## 10 2.6559 191.2608  22.1807 -1.2339 3969.085 3.192426e+63 2.602246e+19
## 11 0.7139 182.8059  26.5497 -0.6531 4166.094 2.468608e+63 2.452977e+19
## 12 0.9332 178.4687  27.7134  0.4399 4270.643 2.316518e+63 2.093373e+19
##          TraceW Friedman  Rubin Cindex     DB Silhouette   Duda Pseudot2
## 2  119558984874  50.2162 2.2783 0.1410 1.4434     0.4560 0.6337 224.2598
## 3   80332414178  52.7588 3.3908 0.1268 1.1175     0.4784 1.2170 -26.9273
## 4   64855545846  62.2554 4.1999 0.1065 1.0629     0.3866 0.9585   5.3233
## 5   53206131638  65.8674 5.1195 0.0924 1.0817     0.3725 1.3944 -44.9731
## 6   47257645711  69.2034 5.7639 0.0997 1.1541     0.3186 1.4368 -27.3624
## 7   41922736034  74.7483 6.4974 0.0908 1.1485     0.3159 2.1220 -80.3708
## 8   36273470300  79.0365 7.5093 0.1033 1.0286     0.3195 1.7861 -51.0527
## 9   34583835537  83.3613 7.8762 0.0977 1.0728     0.3108 1.1325  -9.5946
## 10  31499429725  89.3755 8.6474 0.0967 1.1261     0.2766 1.4466 -29.6358
## 11  29954296827  91.9699 9.0935 0.0920 1.0917     0.2776 2.2725 -27.9981
## 12  28208544206  97.4402 9.6562 0.0866 1.0579     0.2665 1.2820 -16.9388
##      Beale Ratkowsky        Ball Ptbiserial    Frey McClain   Dunn Hubert
## 2   3.0467    0.2709 59779492437     0.4106 -0.0007  0.1641 0.0148      0
## 3  -0.9265    0.2838 26777471393     0.6010  3.3450  0.2588 0.0169      0
## 4   0.2247    0.2891 16213886461     0.5238  1.4382  0.4616 0.0155      0
## 5  -1.4801    0.2656 10641226328     0.4840  3.4153  0.6721 0.0145      0
## 6  -1.3390    0.2540  7876274285     0.4389  1.1288  0.8795 0.0146      0
## 7  -2.7311    0.2477  5988962291     0.4081 -0.1092  1.1064 0.0142      0
## 8  -2.2720    0.2499  4534183788     0.4093  0.9357  1.1024 0.0163      0
## 9  -0.6037    0.2404  3842648393     0.3935  8.0415  1.2351 0.0167      0
## 10 -1.6166    0.2293  3149942973     0.3531  0.6273  1.5827 0.0054      0
## 11 -2.8038    0.2231  2723117893     0.3427  0.5080  1.7139 0.0142      0
## 12 -1.1274    0.2147  2350712017     0.3327  0.9585  1.8470 0.0086      0
##    SDindex    Dindex   SDbw
## 2    5e-04 12586.767 1.4125
## 3    4e-04 10492.876 1.3148
## 4    5e-04  9511.550 1.4042
## 5    5e-04  8447.811 1.2716
## 6    6e-04  8117.653 1.3865
## 7    6e-04  7470.396 1.2558
## 8    6e-04  7261.920 1.2275
## 9    6e-04  7020.144 1.1501
## 10   6e-04  6683.115 1.0208
## 11   6e-04  6492.153 0.9631
## 12   6e-04  6233.716 0.8927
```

```
res$Best.nc
```

```
##                      KL       CH Hartigan    CCC    Scott      Marriot
## Number_clusters 8.0000   5.0000   3.0000 3.0000   4.0000 4.000000e+00
## Value_Index     6.5121 213.3661 109.5928 5.6955 478.5454 6.414079e+63
##                        TrCovW      TraceW Friedman  Rubin  Cindex     DB
## Number_clusters 3.000000e+00           3   4.0000  8.000 12.0000 8.0000
## Value_Index     4.459072e+20 23749702364   9.4966 -0.645  0.0866 1.0286
##                 Silhouette  Duda PseudoT2   Beale Ratkowsky        Ball
## Number_clusters     3.0000 3.000   3.0000  3.0000    4.0000           3
## Value_Index         0.4784 1.217 -26.9273 -0.9265    0.2891 33002021044
##                 PtBiserial Frey McClain   Dunn Hubert SDindex Dindex
## Number_clusters      3.000    1  2.0000 3.0000      0   3e+00      0
## Value_Index          0.601   NA  0.1641 0.0169      0   4e-04      0
##                    SDbw
## Number_clusters 12.0000
## Value_Index      0.8927
```

```
res$All.CriticalValues
```

```
##    CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
## 2          0.8424            72.5845       0.0020
## 3          0.7246            57.3888       1.0000
## 4          0.7213            47.5350       0.9864
## 5          0.7702            47.4294       1.0000
## 6          0.3471           169.2658       1.0000
## 7          0.6943            66.9210       1.0000
## 8          0.6918            51.6883       1.0000
## 9          0.6891            36.9930       1.0000
## 10         0.7758            27.7427       1.0000
## 11         0.5813            36.0200       1.0000
## 12         0.6603            39.6142       1.0000
```

```
res$Best.partition
```

```
##   [1] 3 3 3 3 1 3 3 3 3 2 3 3 1 3 1 3 3 3 3 3 3 3 1 2 1 3 3 3 2 1 3 3 3 1 3
##  [36] 3 1 3 2 1 1 3 3 2 3 2 2 2 3 2 3 3 1 3 1 3 2 3 3 3 3 2 3 3 3 2 3 3 3 3
##  [71] 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 2 2 1 3 1 3 3 2 3 3 3 3 3 3 3 3 3 3 1 3
## [106] 3 3 3 3 2 3 2 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3
## [141] 3 1 1 3 3 2 3 3 3 1 3 3 3 3 3 2 3 3 3 3 3 3 3 2 3 2 3 3 3 3 3 2 3 2 3
## [176] 3 1 3 3 3 3 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 2 2 1 3 3 2 3 3 3 2
## [211] 3 2 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 1 1 1 3 3 3
## [246] 3 3 3 3 3 3 2 3 1 3 1 3 3 1 1 3 3 1 3 3 2 2 3 2 3 3 3 3 1 3 3 1 3 3 3
## [281] 3 3 1 1 1 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 2 3 3 2 3 1 2 3 3
## [316] 3 3 3 3 2 3 3 3 3 1 1 3 3 3 3 3 2 3 2 3 1 3 3 3 3 3 3 3 2 3 3 3 1 3 2
## [351] 3 2 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 1 3 3 1 3 1 3 2
## [386] 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 1 1 3 3 1 2 3 3 3 3 3 3 3 3 3 3 3 2 3
## [421] 3 3 1 3 3 3 3 1 3 3 3 3 3 3 3 1 1 2 3 3
```

```
# K-Means Cluster Analysis (based on the proposed number by NbCluster)
fit <- kmeans(testdata, 3)
```

```
# Calculate average for each cluster
aggregate(data,by=list(fit$cluster),FUN=mean)
```

```
##   Group.1  Channel   Region     Fresh      Milk   Grocery    Frozen
## 1       1 1.015444 2.467181  9530.919  3005.668  3608.687  2583.039
## 2       2 1.065217 2.695652 36500.609  5879.804  6093.000 10373.848
## 3       3 2.000000 2.637037  8389.593 11121.615 16915.807  1521.822
##   Detergents_Paper Delicassen
## 1         785.7181   991.8958
## 2         870.1739  3832.6304
## 3        7587.6148  1761.0444
```

```
# Add segmentation to dataset
data <- data.frame(data, fit$cluster)
```

# Property of each cluster

Cluster 1: Customers are mostly from Hotel/Restaurant/Cafe channel. They spend relatively less than Cluster 2 and Cluster 3 customers.

Cluster 2: Customers are mostly from Hotel/Restaurant/Cafe channel. They spend mostly on Fresh and Frozen products.

Cluster 3: Customers are mostly from Retail channel. They spend mostly on Milk and Grocery products.

# Marketing strategies for the customer segments

Based on the 3 clusters, we could formulate marketing strategies relevant to each cluster:

- A typical strategy would focus on certain promotional efforts for the high value customers of Cluster 2 and Cluster 3.
- For Cluster 2: Customers are mostly from Hotel/Restaurant/Cafe channel. These customers tend to spend more on Fresh and Frozen products. There could be some discounted pricing on those products in order to increase the spend from this segment.
- For Cluster 3: Customers are mostly from Retail channel. These customers tend to spend more on Milk and Grocery products. There could be some discounted pricing based promotional campaigns for this group so as to retain them.