BIG DATA PROJECT – STORE FORMAT ANALYSIS

Malavika Andavilli

PROJECT 1

The company currently has 85 stores and planning to open 10 new stores. Currently, all stores use the same store format for selling their products. Up until now, the company has treated all stores similarly, shipping the same amount of product to each store. This is beginning to cause problems as stores are suffering from product surpluses in some product categories and shortages in others. We have been asked to provide analytical support to make decisions about store formats and inventory planning.

Task1: Determining Store Format

To remedy the product surplus and shortages, the company wants to introduce different store formats. Each store format will have a different product selection in order to better match local demand. The actual building sizes will not change, just the product selection and internal layouts.

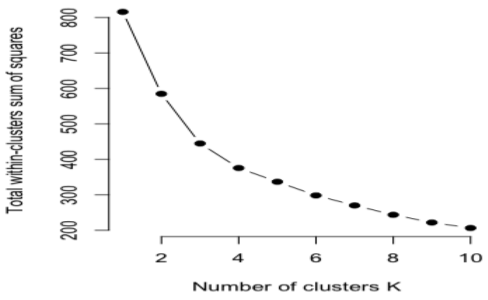
Storesalesdata – This file has the sales by product category for all existing stores for 2012, 2013, 2014 and 2015.

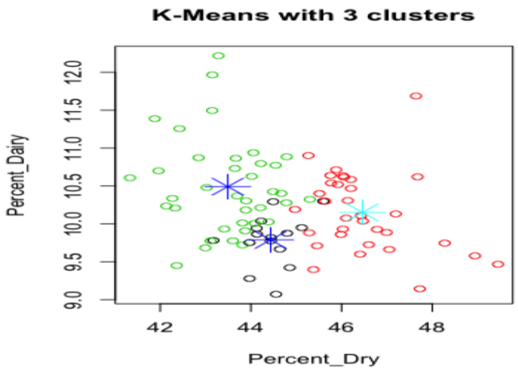
There are different product categories that each store sells.

- For each year we calculated for each category its sales as a percentage of total store sales.
- We then filtered the data for year 2015.
- Optimal number of store formats are 3. We arrived at this figure by looking at F-stat
- F stat for 3 clusters is **34.2238526443412.**

```
[1] "F-stat for 2 clusters is 32.8077742969771"
   > print(paste("F-stat for 3 clusters is", k3))
[1] "F-stat for 3 clusters is 34.2238526443412"
   > print(paste("F-stat for 4 clusters is", k4))
[1] "F-stat for 4 clusters is 31.6681718448581"
   > print(paste("F-stat for 5 clusters is", k5))
[1] "F-stat for 5 clusters is 28.4738235605986"
```

- This is the basis for clustering.





- The elbow is at 3. Hence The optimal number of clusters is 3.

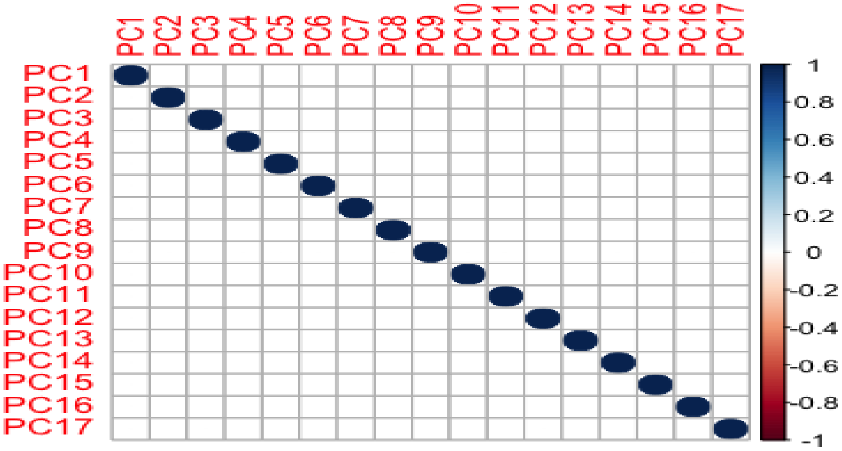
For 2 variables in k-means clustering we plotted the centers and the points.

We also calculated (seg.k3$betweenss/seg.k3$totss*100) which was 45%. This shows that there is more distance between the clusters.

Cluster	Number of Stores
1	15
2	33
3	37

TASK2- Applying machine learning algorithms to predict store format for 10 new stores.

We need to create a model for the existing stores based on the demographic data.
Storedemographicdata.xlsx – This file has the demographics of the stores.
From PART1 we have the clusters for each store. After merging the data we have a new file with StoreID, Demographics and Cluster.
We first do a PCA to reduce the number of columns from 46. This is because there can be many highly co related variables.



After PCA, we divided the existing stores into training set and test set. We used 80,20% rule.

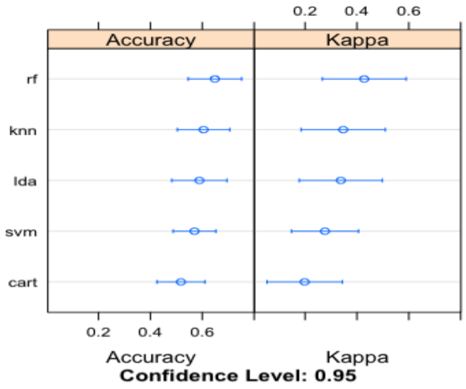
After this we performed various algorithms like SVM, Random Forest, KNN etc.
We got the highest accuracy percentage for Random Forest. We also see the spread of the accuracy.
Accuracy

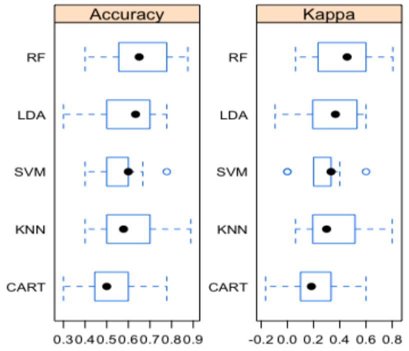
	Min	1st Qu.	Median	Mean
lda	0.2857143	0.4062500	0.6904762	0.6202381
cart	0.2857143	0.4285714	0.5000000	0.5095238
knn	0.4285714	0.5714286	0.6250000	0.6226190
svm	0.2857143	0.5000000	0.6190476	0.5994048
rf	0.2857143	0.5000000	**0.6696429**	0.6839286

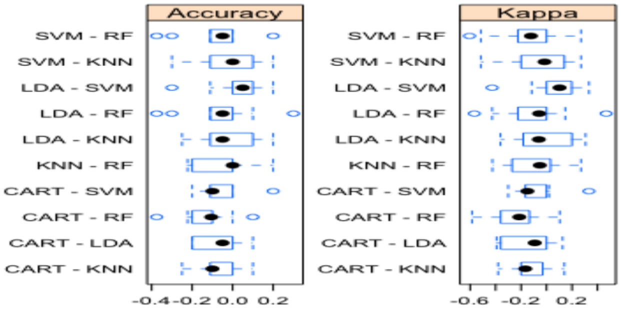
Kappa

	Min.	1st Qu.	Median	Mean

lda	0.25000000	0.12232558	0.4772727	0.3916813
cart	-0.09375000	0.01666667	0.1904762	0.2066205
knn	0.06666667	0.25000000	0.4073171	0.3702892
svm	-0.25000000	0.16202091	0.3522727	0.3311842
rf	-0.16666667	0.23463415	**0.4885671**	0.4969503







- Based on this we finalized on Random Forest.
- Accuracy is so less for all models is because we have very few observations in this data. This is the reason random forest has very less accuracy.
- We predict the cluster number for the rest 20% of the data using SVM,RF and CART.

SVM results:
For the test data it predicted the following. Confusion matrix shows an accuracy of 62%.

```
  > print(predictions)
 [1] Cluster3 Cluster3 Cluster1 Cluster3 Cluster3 Cluster3 Cluster2 Cluster2 Cluster2 Cluster3
Cluster2 Cluster2 Cluster3 Cluster2
[15] Cluster1 Cluster3
Levels: Cluster1 Cluster2 Cluster3
> confusionMatrix(as.factor(predictions), as.factor(validation$Cluster))
Confusion Matrix and Statistics

          Reference
```

```
Prediction Cluster1 Cluster2 Cluster3
  Cluster1        1        1        0
  Cluster2        2        3        1
  Cluster3        0        2        6

Overall Statistics

               Accuracy : 0.625
                 95% CI : (0.3543, 0.848)
    No Information Rate : 0.4375
    P-Value [Acc > NIR] : 0.1043

                  Kappa : 0.3924
 Mcnemar's Test P-Value : NA
```

RF results:
For the test data it predicted the following. Confusion matrix shows an accuracy of 62%.

```
> print(predictions)
 [1] Cluster3 Cluster2 Cluster1 Cluster3 Cluster3 Cluster3 Cluster2 Cluster2 Cluster2 Cluster3
Cluster1 Cluster2 Cluster3 Cluster2
[15] Cluster1 Cluster3
Levels: Cluster1 Cluster2 Cluster3
> confusionMatrix(as.factor(predictions), as.factor(validation$Cluster))
Confusion Matrix and Statistics

          Reference
Prediction Cluster1 Cluster2 Cluster3
  Cluster1        1        2        0
  Cluster2        2        3        1
  Cluster3        0        1        6

Overall Statistics

               Accuracy : 0.625
                 95% CI : (0.3543, 0.848)
    No Information Rate : 0.4375
    P-Value [Acc > NIR] : 0.1043

                  Kappa : 0.4074
 Mcnemar's Test P-Value : NA
```

CART results:
For the test data it predicted the following. Confusion matrix shows an accuracy of 68%.

```
> print(predictions)
```

```
 [1] Cluster3 Cluster2 Cluster2 Cluster3 Cluster3 Cluster3 Cluster2 Cluster2 Cluster2 Cluster3
Cluster2 Cluster2 Cluster3 Cluster2
[15] Cluster2 Cluster3
> confusionMatrix(as.factor(predictions), as.factor(validation$Cluster))
Confusion Matrix and Statistics

          Reference
Prediction Cluster1 Cluster2 Cluster3
  Cluster1        0        0        0
  Cluster2        3        5        1
  Cluster3        0        1        6

Overall Statistics

               Accuracy : 0.6875
                 95% CI : (0.4134, 0.8898)
    No Information Rate : 0.4375
    P-Value [Acc > NIR] : 0.03915

                  Kappa : 0.4771
 Mcnemar's Test P-Value : NA
```

Hence RF is the best followed by SVM.

We used RF fit to predict the cluster number for the 10 new stores.

It predicted the following for the 10 new stores.

```
> print(predictions)
 [1] Cluster2 Cluster3 Cluster2 Cluster3 Cluster3 Cluster2 Cluster3 Cluster1 Cluster3 Cluster3
Levels: Cluster1 Cluster2 Cluster3
```

Limitations:
This way of predicting the clusters for 10 new stores has the following limitations:

- Since number of rows are different for existing stores and new stores, PCA returns different number of consolidated columns. This leads to difference in the way prediction can happen.
- Hence, for performing PCA, 10 new stores are appended with the existing stores. Random numbers have been assigned for cluster column for 10 new stores. After PCA, existing stores are separated from new stores. 80,20 rule is then applied to the existing stores.
- Since there are 46 columns and 85 rows for existing stores, this leads to a problem for performing analysis. More data is hence preferred.