Credit Card Segmentation Case Study

Niranjan

24 December 2018

```
# Setting Working Directory
setwd("C:/Users/Niranjan/Documents/Case studies/Segmentation/CC/")

# Importing the file
cc <- fread("CC GENERAL.csv")
```

Creating new measures

```
cc$AVG_PURCHASES <- cc$PURCHASES/12
cc$AVG_CASH_ADVANCE <- cc$CASH_ADVANCE/12
cc$PURCHASE_TYPE <- ifelse(cc$ONEOFF_PURCHASES == 0 & cc$INSTALLMENTS_PURCHASES > 0,"Installm
ents",
                           ifelse(cc$ONEOFF_PURCHASES > 0 & cc$INSTALLMENTS_PURCHASES == 0,"O
ne-off","Cash Advance"))
cc$LIMIT_USAGE <- cc$BALANCE/cc$CREDIT_LIMIT
cc$PAYM_MIN_PAYM <- cc$PAYMENTS/cc$MINIMUM_PAYMENTS
```

UDF for extracting numeric and categorical variables

Understanding more about the numeric variables

```
nums <- names(cc)[sapply(cc, is.numeric)]
abt_nums <- as.data.frame(t(sapply(cc[,..nums],about)))
abt_rest <- as.data.frame(t(sapply(cc[,!..nums],about_rest)))
table(cc$PURCHASE_TYPE)
```

```
##
## Cash Advance Installments      One-off
##         4816         2260         1874
```

```
fwrite(abt_nums,"abt_nums.csv",row.names = T)
```

In the above the missing values are less than 5 percent, so those observations are removed

```
cc <- cc[complete.cases(cc),]
```

Capping outliers

```
cc$BALANCE <- ifelse(cc$BALANCE>quantile(cc$BALANCE,p=0.99),quantile(cc$BALANCE,p=0.99),cc$BALANCE)
cc$BALANCE <- ifelse(cc$BALANCE<quantile(cc$BALANCE,p=0.01),quantile(cc$BALANCE,p=0.01),cc$BALANCE)
cc$PURCHASES <- ifelse(cc$PURCHASES>quantile(cc$PURCHASES,p=0.99),quantile(cc$PURCHASES,p=0.99),cc$PURCHASES)
cc$PURCHASES <- ifelse(cc$PURCHASES<quantile(cc$PURCHASES,p=0.01),quantile(cc$PURCHASES,p=0.01),cc$PURCHASES)
cc$ONEOFF_PURCHASES <- ifelse(cc$ONEOFF_PURCHASES>quantile(cc$ONEOFF_PURCHASES,p=0.99),quantile(cc$ONEOFF_PURCHASES,p=0.99),cc$ONEOFF_PURCHASES)
cc$ONEOFF_PURCHASES <- ifelse(cc$ONEOFF_PURCHASES<quantile(cc$ONEOFF_PURCHASES,p=0.01),quantile(cc$ONEOFF_PURCHASES,p=0.01),cc$ONEOFF_PURCHASES)
cc$INSTALLMENTS_PURCHASES <- ifelse(cc$INSTALLMENTS_PURCHASES>quantile(cc$INSTALLMENTS_PURCHASES,p=0.99),quantile(cc$INSTALLMENTS_PURCHASES,p=0.99),cc$INSTALLMENTS_PURCHASES)
cc$INSTALLMENTS_PURCHASES <- ifelse(cc$INSTALLMENTS_PURCHASES<quantile(cc$INSTALLMENTS_PURCHASES,p=0.01),quantile(cc$INSTALLMENTS_PURCHASES,p=0.01),cc$INSTALLMENTS_PURCHASES)
cc$CASH_ADVANCE <- ifelse(cc$CASH_ADVANCE>quantile(cc$CASH_ADVANCE,p=0.99),quantile(cc$CASH_ADVANCE,p=0.99),cc$CASH_ADVANCE)
cc$CASH_ADVANCE <- ifelse(cc$CASH_ADVANCE<quantile(cc$CASH_ADVANCE,p=0.01),quantile(cc$CASH_ADVANCE,p=0.01),cc$CASH_ADVANCE)
cc$CASH_ADVANCE_FREQUENCY <- ifelse(cc$CASH_ADVANCE_FREQUENCY>quantile(cc$CASH_ADVANCE_FREQUENCY,p=0.99),quantile(cc$CASH_ADVANCE_FREQUENCY,p=0.99),cc$CASH_ADVANCE_FREQUENCY)
cc$CASH_ADVANCE_FREQUENCY <- ifelse(cc$CASH_ADVANCE_FREQUENCY<quantile(cc$CASH_ADVANCE_FREQUENCY,p=0.01),quantile(cc$CASH_ADVANCE_FREQUENCY,p=0.01),cc$CASH_ADVANCE_FREQUENCY)
cc$CASH_ADVANCE_TRX <- ifelse(cc$CASH_ADVANCE_TRX>quantile(cc$CASH_ADVANCE_TRX,p=0.99),quantile(cc$CASH_ADVANCE_TRX,p=0.99),cc$CASH_ADVANCE_TRX)
cc$CASH_ADVANCE_TRX <- ifelse(cc$CASH_ADVANCE_TRX<quantile(cc$CASH_ADVANCE_TRX,p=0.01),quantile(cc$CASH_ADVANCE_TRX,p=0.01),cc$CASH_ADVANCE_TRX)
cc$PURCHASES_TRX <- ifelse(cc$PURCHASES_TRX>quantile(cc$PURCHASES_TRX,p=0.99),quantile(cc$PURCHASES_TRX,p=0.99),cc$PURCHASES_TRX)
cc$PURCHASES_TRX <- ifelse(cc$PURCHASES_TRX<quantile(cc$PURCHASES_TRX,p=0.01),quantile(cc$PURCHASES_TRX,p=0.01),cc$PURCHASES_TRX)
cc$CREDIT_LIMIT <- ifelse(cc$CREDIT_LIMIT>quantile(cc$CREDIT_LIMIT,p=0.99),quantile(cc$CREDIT_LIMIT,p=0.99),cc$CREDIT_LIMIT)
cc$CREDIT_LIMIT <- ifelse(cc$CREDIT_LIMIT<quantile(cc$CREDIT_LIMIT,p=0.01),quantile(cc$CREDIT_LIMIT,p=0.01),cc$CREDIT_LIMIT)
cc$PAYMENTS <- ifelse(cc$PAYMENTS>quantile(cc$PAYMENTS,p=0.99),quantile(cc$PAYMENTS,p=0.99),cc$PAYMENTS)
cc$PAYMENTS <- ifelse(cc$PAYMENTS<quantile(cc$PAYMENTS,p=0.01),quantile(cc$PAYMENTS,p=0.01),cc$PAYMENTS)
cc$MINIMUM_PAYMENTS <- ifelse(cc$MINIMUM_PAYMENTS>quantile(cc$MINIMUM_PAYMENTS,p=0.99),quantile(cc$MINIMUM_PAYMENTS,p=0.99),cc$MINIMUM_PAYMENTS)
cc$MINIMUM_PAYMENTS <- ifelse(cc$MINIMUM_PAYMENTS<quantile(cc$MINIMUM_PAYMENTS,p=0.01),quantile(cc$MINIMUM_PAYMENTS,p=0.01),cc$MINIMUM_PAYMENTS)
cc$AVG_PURCHASES <- ifelse(cc$AVG_PURCHASES>quantile(cc$AVG_PURCHASES,p=0.99),quantile(cc$AVG_PURCHASES,p=0.99),cc$AVG_PURCHASES)
cc$AVG_PURCHASES <- ifelse(cc$AVG_PURCHASES<quantile(cc$AVG_PURCHASES,p=0.01),quantile(cc$AVG_PURCHASES,p=0.01),cc$AVG_PURCHASES)
cc$AVG_CASH_ADVANCE <- ifelse(cc$AVG_CASH_ADVANCE>quantile(cc$AVG_CASH_ADVANCE,p=0.99),quantile(cc$AVG_CASH_ADVANCE,p=0.99),cc$AVG_CASH_ADVANCE)
cc$AVG_CASH_ADVANCE <- ifelse(cc$AVG_CASH_ADVANCE<quantile(cc$AVG_CASH_ADVANCE,p=0.01),quantile(cc$AVG_CASH_ADVANCE,p=0.01),cc$AVG_CASH_ADVANCE)
cc$LIMIT_USAGE <- ifelse(cc$LIMIT_USAGE>quantile(cc$LIMIT_USAGE,p=0.99),quantile(cc$LIMIT_USAGE,p=0.99),cc$LIMIT_USAGE)
cc$LIMIT_USAGE <- ifelse(cc$LIMIT_USAGE<quantile(cc$LIMIT_USAGE,p=0.01),quantile(cc$LIMIT_USAGE,p=0.01),cc$LIMIT_USAGE)
cc$PAYM_MIN_PAYM <- ifelse(cc$PAYM_MIN_PAYM>quantile(cc$PAYM_MIN_PAYM,p=0.99),quantile(cc$PAY
```

```
M_MIN_PAYM,p=0.99),cc$PAYM_MIN_PAYM)
cc$PAYM_MIN_PAYM <- ifelse(cc$PAYM_MIN_PAYM<quantile(cc$PAYM_MIN_PAYM,p=0.01),quantile(cc$PAY
M_MIN_PAYM,p=0.01),cc$PAYM_MIN_PAYM)
```

Identifying relationships

It can be done only on numeric variables

```
cc$CUST_ID <- NULL # Cust id is removed from analysis
cc <- dummy_cols(cc, remove_first_dummy = F) # Dummy vars are created for purchase type
cc$PURCHASE_TYPE <- NULL

cc_numeric <- cc
```

A correlation Matrix is built

```
corrm<- as.data.frame(cor(cc_numeric))
fwrite(corrm,"cor_mat.csv",row.names = T)
```

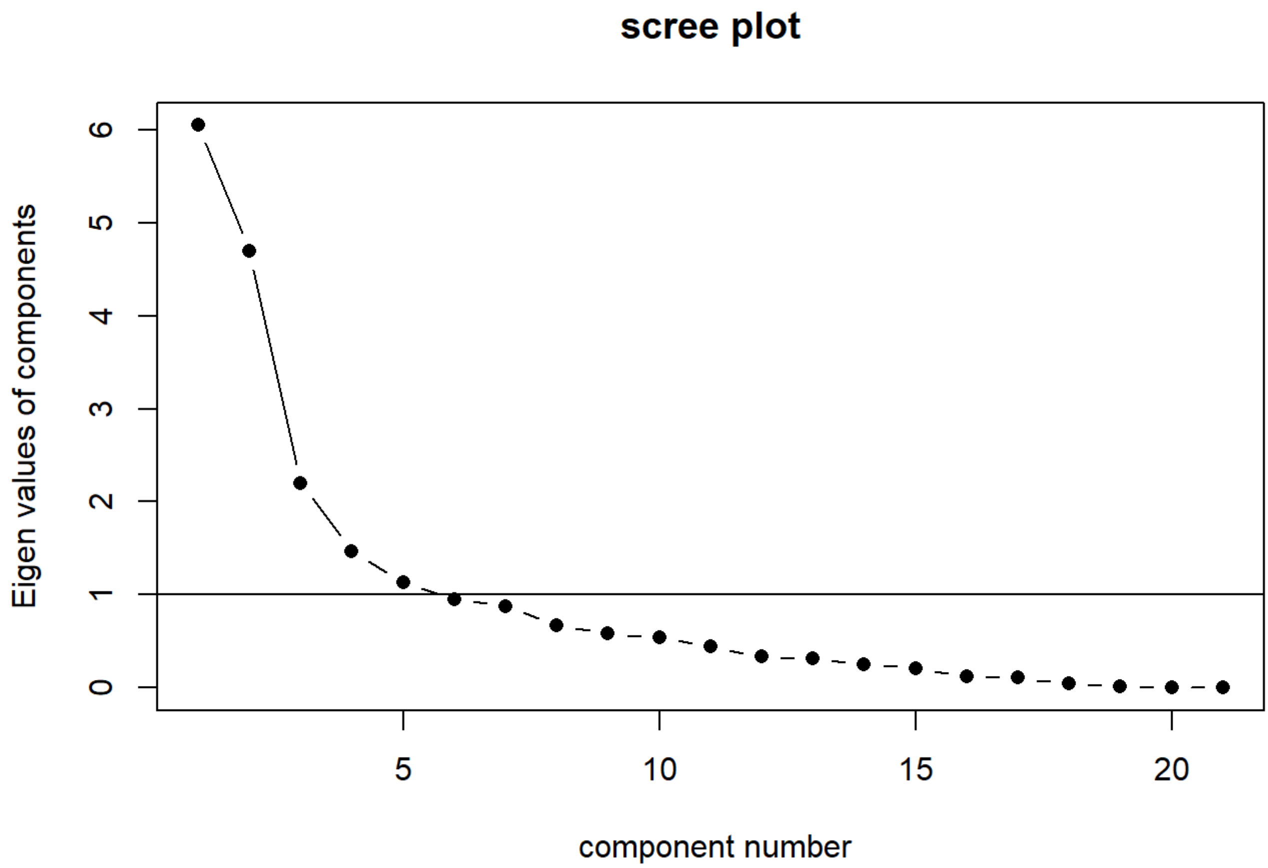
There does not seem to be any kind of strong relationships between the services

Factor analysis is conducted

```
cc_numeric$PURCHASE_TYPE_Installments <- NULL
cc_numeric$`PURCHASE_TYPE_Cash Advance` <- NULL
cc_numeric$`PURCHASE_TYPE_One-off` <- NULL
corrm <- cor(cc_numeric)
```

Relationship of dummy variables are removed as they are important for deriving cluster insigts

```
scree(corrm, factors=F, pc=T, main="scree plot", hline=NULL, add=FALSE)
```




```
(eigen_values <- mutate(data.frame(eigen(corrm)$values)
                       ,cum_sum_eigen=cumsum(eigen.corrm..values)
                       , pct_var=eigen.corrm..values/sum(eigen.corrm..values)
                       , cum_pct_var=cum_sum_eigen/sum(eigen.corrm..values)))
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
##           eigen.corrm..values cum_sum_eigen                  pct_var
## 1   6.049148597902156865302459      6.049149  0.2880546951381979248196785
## 2   4.696401203722193606893143     10.745550  0.2236381525581996598806001
## 3   2.204024833342390188306581     12.949575  0.1049535634924947530288719
## 4   1.468193998172079295372328     14.417769  0.0699139999129561456880921
## 5   1.134584693373142494365879     15.552353  0.0540278425415782054264113
## 6   0.951868928648689105465053     16.504222  0.0453270918404137593271663
## 7   0.878658190002797656603661     17.382880  0.0418408661906094075932927
## 8   0.668326636469557144870635     18.051207  0.0318250779271217615340461
## 9   0.578639148266722469493573     18.629846  0.0275542451555582067201922
## 10  0.541997443721224825274874     19.171844  0.0258094020819630842744630
## 11  0.440245186707924673896741     19.612089  0.0209640565099011716432287
## 12  0.332940976168823010716835     19.945030  0.0158543321985153799758006
## 13  0.310518775143510616931763     20.255549  0.0147866083401671701696989
## 14  0.249826769799840586738782     20.505375  0.0118965128476114546940412
## 15  0.206703248072029943660510     20.712079  0.0098430118129538054366767
## 16  0.123152336362879755160549     20.835231  0.0058643969696609401023868
## 17  0.105003736097206520905267     20.940235  0.0050001779093907855119894
## 18  0.042220923578735074233048     20.982456  0.0020105201704159556680418
## 19  0.017544374448098972590904     21.000000  0.0008354464022904270958589
## 20  0.000000000000000519013479     21.000000  0.0000000000000000247149276
## 21 -0.000000000000000005535331     21.000000 -0.0000000000000000002635872
##    cum_pct_var
## 1    0.2880547
## 2    0.5116928
## 3    0.6166464
## 4    0.6865604
## 5    0.7405883
## 6    0.7859153
## 7    0.8277562
## 8    0.8595813
## 9    0.8871355
## 10   0.9129449
## 11   0.9339090
## 12   0.9497633
## 13   0.9645499
## 14   0.9764464
## 15   0.9862895
## 16   0.9921539
## 17   0.9971540
## 18   0.9991646
## 19   1.0000000
## 20   1.0000000
## 21   1.0000000
```

Eigen value of 1 is taken as a cutoff, so 5 factors are taken into consideration

Running the factor analysis

```
FA <- NULL
FA<-fa(r=corrm,5, rotate="varimax", fm="pa",SMC = F)
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
## The estimated weights for the factor scores are probably incorrect.  Try a different facto
r extraction method.
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
FA_SORT<-fa.sort(FA)
Loadings<-data.frame(FA_SORT$loadings[1:ncol(cc_numeric),])

fwrite(Loadings,"factor_loadings.csv",row.names = T)
```

Note: as optimization cannot be done the factor extraction method that is adopted here is “pa” or principal factor solution

Even this method has a few errors, but it is ignored as this is the only method that yields results.

Clusturing

Selecting the final variables for segmentation

```
vars <- c("ONEOFF_PURCHASES","PURCHASES","CASH_ADVANCE","INSTALLMENTS_PURCHASES","LIMIT_USAG
E","BALANCE",
          "PURCHASE_TYPE_Installments","PURCHASE_TYPE_Cash Advance","PURCHASE_TYPE_One-off")
```

Factor analysis was conducted and variables were selected. The amount/value based variables for each purchase type were selected because in most factors they had higher loadings and were preferred over the other variables to keep uniformity in variable selection.

Conducting the clusturing analysis

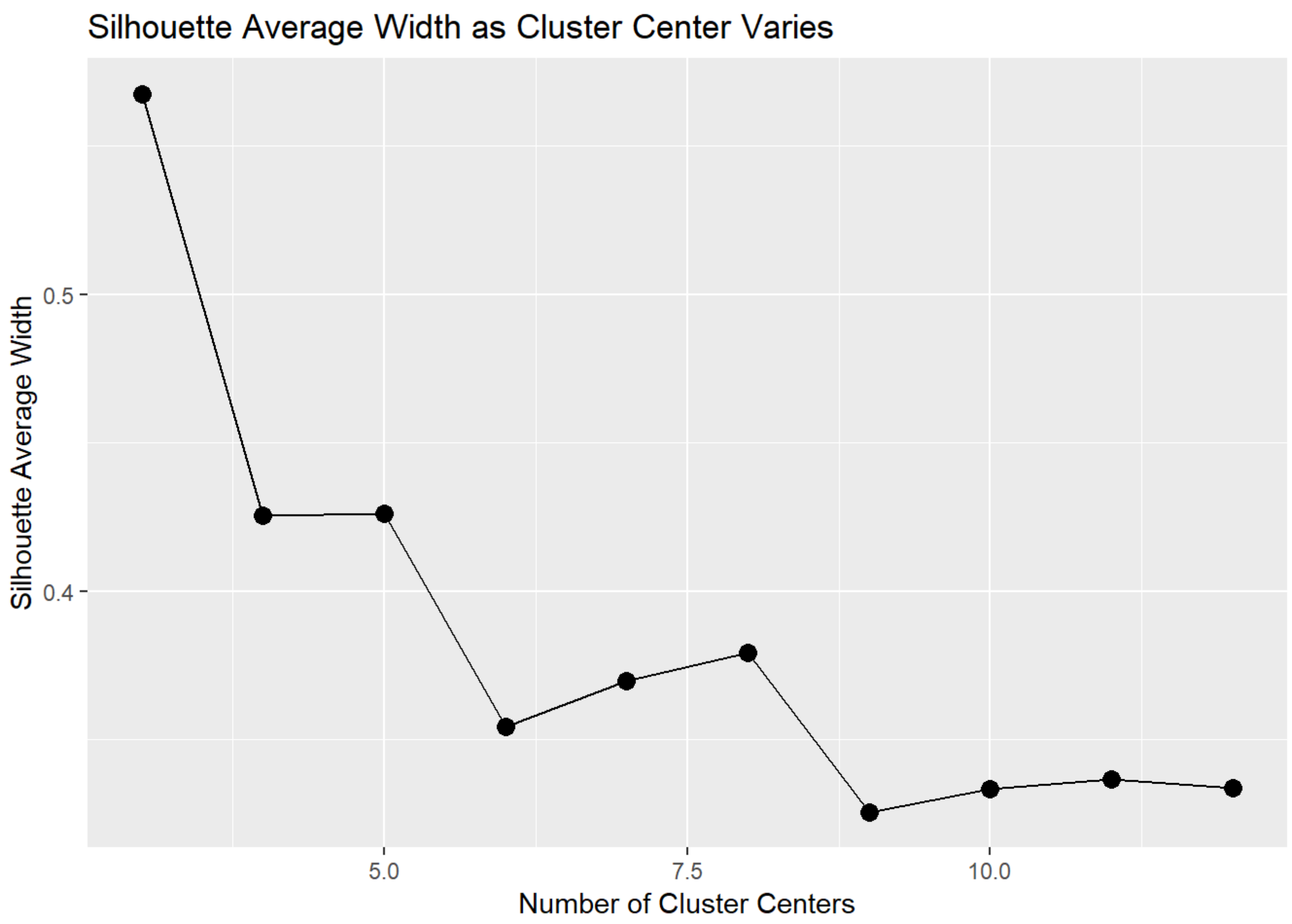
```
inputdata_final <- cc[,vars,with = F]

km.out <- list()
sil.out <- list()
x <- vector()
y <- vector()

minClust <- 3
maxClust <- 12

for (centr in minClust:maxClust) {
  i <- centr-(minClust-1)
  set.seed(11)
  km.out[i] <- list(kmeans(inputdata_final, centers = centr))
  sil.out[i] <- list(silhouette(km.out[[i]][[1]], dist(inputdata_final)))
  x[i] = centr
  y[i] = summary(sil.out[[i]])[[4]]
}


ggplot(data = data.frame(x, y), aes(x, y)) +
  geom_point(size=3) +
  geom_line() +
  xlab("Number of Cluster Centers") +
  ylab("Silhouette Average Width") +
  ggtitle("Silhouette Average Width as Cluster Center Varies")
```



cluster three, four and five are meaningful to be considered here

Profiling for the select clusters alone

```
cluster_three <- kmeans(inputdata_final,3)
cluster_four <- kmeans(inputdata_final,4)
cluster_five <- kmeans(inputdata_final,5)

cc_seg<-cbind(cc,clust_3=cluster_three$cluster,clust_4=cluster_four$cluster,clust_5=cluster_f
ive$cluster)

cc_seg$clust_3 <- as.factor(cc_seg$clust_3)
cc_seg$clust_4 <- as.factor(cc_seg$clust_4)
cc_seg$clust_5 <- as.factor(cc_seg$clust_5)

profiles <- tabular(1+ONEOFF_PURCHASES+PURCHASES+CASH_ADVANCE+INSTALLMENTS_PURCHASES+LIMIT_US
AGE+BALANCE+PURCHASE_TYPE_Installments+`PURCHASE_TYPE_Cash Advance`+`PURCHASE_TYPE_One-off`~m
ean+(mean*clust_3)+(mean*clust_4)+(mean*clust_5),data = cc_seg)
profiles1 <- as.data.table(as.matrix(profiles))

profiles<-tabular(1~length+(length*clust_3)+(length*clust_4)+(length*clust_5),data=cc_seg)
profiles2<-data.table(as.matrix(profiles))

fwrite(profiles1,"seg_profiles_1.csv",row.names = T)
fwrite(profiles2,"seg_profiles_2.csv",row.names = T)
```

The 4 segment is used to explain the profiles as there is more explaination given by that compared to 3 segment, 5 Segment is not considered because it explains behavioural pattern of the same groups as 4 segment and adds nothing new. Even in the 4 segment, profile there is a lot of similarities between two segments.

The segments identified are

1. The low spending segment that makes more installment purchase than overall installmet purchase proportion and have low balance and limit usage ratio than the overall average

2. The high spending segment that spends a lot more in average purchase than the overall average and make a lot of cash advance purchase but in less value than overall average but has the highest proportion of purchase in cash advance. They also have better than average balance.

3. The cash-advance purchase mid value spenders who make a lot of advance purchases in value and in proportion than the overall average and also maintain better balance and limit ratio

4. The cash-advance purchase high value spenders. They are the same as the previous group but have a way higher amount spent of cash advance purchases

Strategic Insights

From the segments created, targeting efforts can be done in a better way by the company. For the first segment, continuous email marketing efforts promoting cash advance purchases can be done. Special discounts for two or three cash advance purchases can be done.

The second segment can also be given special discounts in amount of their total order value for cash advance transactions is more than 950 (The current overall average)

The thrid and fourth segments are almost similar except for the magnitude of the value. The third segment is the one that make continuous purchases and make 19% of the total hence some retention initiatives in form of loyalty purchase programs can be initiated. They also have good balance, so their credit limit can also be increased. Some incentives can be given for installment purchases.

The fourth segment is the high value spenders that are the most important of all. Retention measures like discounts and loyalty programs can be used for them too.