CLV - Customer Strategy

Mehdi Farhangian

12/20/2019

R Markdown

Marketers often ask how they can evaluate the effect of a new campaign or product on the retention of their customers. Is a new marketing campaign making our customers purchase more often over time? Or this marketing campaign is the root cause of our recent faster rate of customers' fade?

The way to deal with these questions is a contentious topic around Marketers. As we know from experience when customers make their first purchase, some of these customers turn into repeated buyers, some customers stop purchasing altogether or some customers purchase with less frequency. It is impossible to turn all of our customers into loyal and repeated customers. However, there are some ways to improve our customer relationships and allow them to become them more loyal. Traditionally we group customers and see their frequency in each segmentation. Often this segmentation can be based on the lifetime value of customers. Frequency segmentation is a simple but powerful approach that helps us to develop smart strategies for dealing with a different group of customers.

In order to know the effect of a new product on the retention of customers, we should look for differences. Analysing differences in terms of acquisition channels or demographic characteristics (such as gender, age or geography) can help us to implement customized strategies with different groups of customers. For instance, we may consider having different promotions for different groups of customers and develop a smart promotion system.

Visualisation of Customer Segmentation

Through using the R programming language, we've prepared an example to visualize customer segmentation and compare the conventional modelling with uplift modelling.

In this example, we have a dataset of the customers who visited our website and purchased a product with a promotion. This dataset contains four columns: Customer ID, Promotion, Device and Revenue. In order to demonstrate and compare the effect of a promotion on revenue for different customer segmentation, we selected two segmentations which are device and frequency of purchase.

The following code in R adds a new column, to the frequency of purchase for each customer. Then, we label each customer based on the frequency of purchase as Low, Medium, High and Very High.

```r
#Generate a data set
N <- 1000
df <- data.frame(CustomerID = sample(1:500, size = N, replace = TRUE),
Promotion = sample(c("Cruise","Flight","Hotel","Package"), size = N, replace = TRUE),
Device = sample(c("Desktop","Mobile","Tablet"), size = N, replace = TRUE),
Revenue = sample(100:500, size = N, replace = TRUE))
```

For the sake of simplicity, we only consider the first promotion and the first device in the customers' journeys.

```r
# Group customers based on their Customer ID and determine their first device, first promotion and the
library (dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

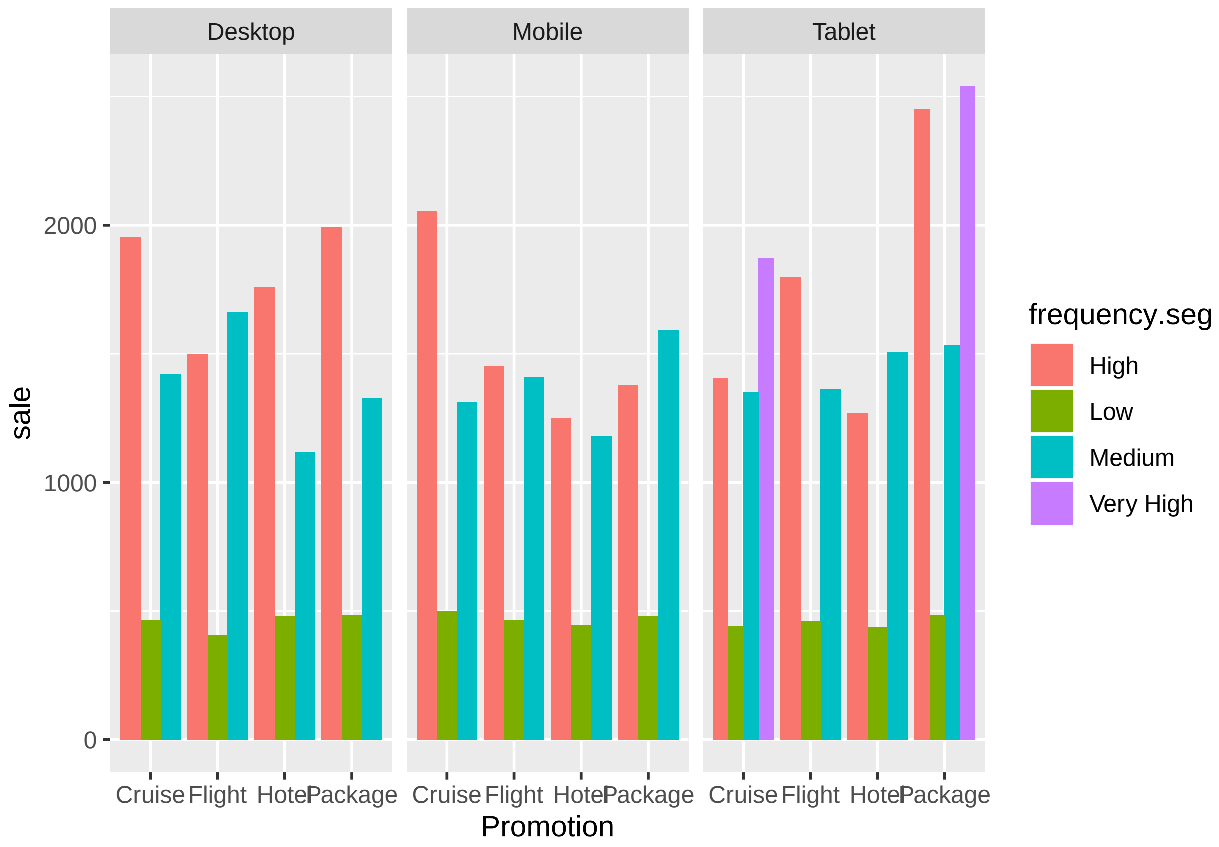
```
df2<- df %>% group_by(CustomerID ) %>%
summarise(frequency =n(), sale = sum(Revenue),Promotion= first (Promotion), Device = first (Device) )
```

We then group the customers based on the frequency of purchase into four categories which are Low, Medium, High and Very High.

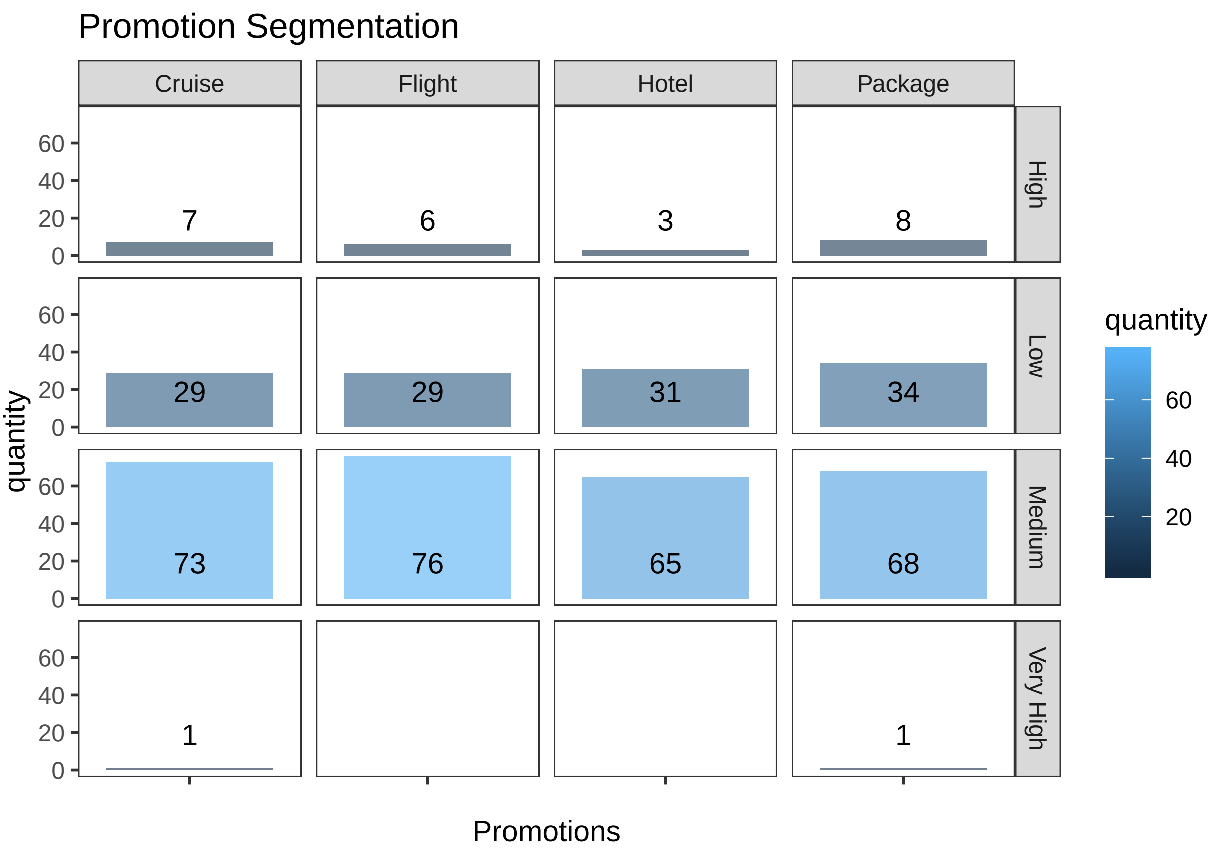
```
df3 <- df2 %>% mutate(frequency.seg=ifelse(between(frequency, 1, 1), "Low",ifelse(between(frequency, 2,
```

Customers' retention can be caught effectively with data visualization. Here we can see the behaviour of a different group of customers, and the potential for the future marketing strategies, as well as budget allocation by some simple but effective visualisation approach. In this article, we show you how you can visualise your customers' responses to a promotion or a device usage in a few ways. One way is using a bar chart:

```
library(ggplot2)
ggplot(df3, aes(x = Promotion, y = sale, fill = frequency.seg)) + geom_bar(stat = 'identity', position
```



```
df4 <- df3 %>% group_by(Promotion, frequency.seg) %>% summarise(quantity=n()) %>% mutate(Promotions='')
ungroup()
ggplot(df4, aes(x=Promotions, y=quantity, fill=quantity)) + theme_bw() + theme(panel.grid = element_blan
label=quantity), size=4) + facet_grid(frequency.seg ~ Promotion) + ggtitle("Promotion Segmentation")
```



That resulted in the following chart:

In order have a better understanding of different segmentation of our customers, we then add recency of purchase for each customer. After adding this we compare the different segments of customers, based on their frequency and recency. After comparing them we can visualize how each segment responded differently to different promotions. This technique also shows if the device category plays a role in these differences. You can also add a column to your new dataset which measures the recency of your customers and can be based on the difference between today's date and the past purchase of your customers. This column has three values, "0-10 days", "11-30 days" and "More than 30 days". Name this new dataset as df4.

Have a closer look at the above figure. How many marketing questions can you see being answered by this simple visualization?

##Uplift modelling

Although this traditional approach is insightful, it cannot foresee the behaviour of the customers in the future. Regarding this issue, uplift modelling has been widely used to predict the customers' next moves. By using uplift modelling, we can answer the following questions:

Which marketing actions should a company design? Which groups of customers should be exposed to the marketing activities? What is the optimal price to be charged? Uplift modelling predicts the behaviour of the customers, by looking at two groups of people. There is one group of people which are subject to the marketing actions and are called the treatment group, and the other group of people who were not subject to marketing actions who are called the control group. Uplift modelling identify the differences in behaviour between these two groups and allow us to find patterns in the variation.

Although uplift modelling can be applied in multiple areas, the biggest success of this approach comes to the area of customer retention that predicts which customers are at the risk of churn.

Here we will investigate if a promotion affects the customer's retention or not.

In order to apply the uplift modelling on the above-mentioned example, we need to add two columns to this dataset. Firstly we need to add a response variable. In our example, a response variable is a binary variable, where 1 shows that the return to the website after 60 days and 0 shows that the customer didn't return. Secondly, we add a binary variable that is called treatment in which 1 indicates whether a promotion is offered and 0 indicates that no promotion is offered to the customer.

Uplift package in R

In the Uplift package, five methods are implemented which are:

- Causal conditional inference forests (ccif)
- Uplift random forests (upliftRF)
- Modified covariate method (tian_transf)
- Modified outcome method (rvtu)
- Uplift k-nearest neighbor (upliftKNN)

In the following example, we modify our data set with the two additional columns (as mentioned before) and then in Uplift using the Uplift random forest, it determines which type of customers are more likely to respond positively to an offer. With the following code, we can predict the likelihood of a purchase for a customer who is exposed to a treatment (e.g. an advertisement).

```
N <- 1000
df <- data.frame(CustomerID = sample(1:500, size = N, replace = TRUE),
 Promotion = sample(c("Cruise","Flight","Hotel","Package"), size = N, replace = TRUE), Device = sample(
library(uplift)
```

```
## Warning: package 'uplift' was built under R version 3.6.1

## Loading required package: RItools

## Warning: package 'RItools' was built under R version 3.6.1

## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: coin

## Warning: package 'coin' was built under R version 3.6.1

## Loading required package: survival

## Loading required package: tables

## Warning: package 'tables' was built under R version 3.6.1

## Loading required package: Hmisc

## Warning: package 'Hmisc' was built under R version 3.6.1

## Loading required package: lattice
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: penalized
```

```
## Warning: package 'penalized' was built under R version 3.6.1
```

```
## Welcome to penalized. For extended examples, see vignette("penalized").
```

```
fit1 <- upliftRF(purchased ~ Promotion + Device + trt(treatment), data = df, mtry = 1, ntree = 50, spli
```

```
## uplift: status messages enabled; set "verbose" to false to disable
```

```
## upliftRF: starting. Sat Dec 21 00:26:56 2019
```

```
## 10 out of 50 trees so far...
```

```
## 20 out of 50 trees so far...
```

```
## 30 out of 50 trees so far...
```

```
## 40 out of 50 trees so far...
```

```
### Fitted values on train data
pred <- predict(fit1, df)
### Compute uplift predictions
uplift_pred <- pred[, 1] - pred[, 2]
```

So we can now compute uplift predictions. Also, we are able to profile our data based on the fitted model. More details can be seen in the documentation of Uplift package.