Multichannel Attribution in Marketing

Karthik

In this document, I have tried to apply Heuristic models and Markov model to solve the problem of Multichannel Attribution that the Marketing departments face.

The data set used, and code inspiration is from “Analyticsvidhya.com”. **It is a survey data obtained by an e-commerce company from its customers about various touchpoints visited by the customer before finally purchasing**.

In total, there are 19 channels where customers can encounter the product or the product advertisement. There are three more cases as well:

#20 – customer has decided which product to buy; #21 – customer has made the final purchase, and; #22 – customer hasn’t decided yet.

This analysis is only beginning into Channel attribution. Some real world problems such as “Customer Journey length” and “Cross device tracking” are not part of the analysis.

Loading the data

```
channel = read.csv("Channel_attribution.csv", header = T)

head(channel)
```

```
##   R05A.01 R05A.02 R05A.03 R05A.04 R05A.05 R05A.06 R05A.07 R05A.08 R05A.09
## 1      16       4       3       5      10       8       6       8      13
## 2       2       1       9      10       1       4       3      21      NA
## 3       9      13      20      16      15      21      NA      NA      NA
## 4       8      15      20      21      NA      NA      NA      NA      NA
## 5      16       9      13      20      21      NA      NA      NA      NA
## 6       1      11       8       4       9      21      NA      NA      NA
##   R05A.10 R05A.11 R05A.12 R05A.13 R05A.14 R05A.15 R05A.16 R05A.17 R05A.18
## 1      20      21      NA      NA      NA      NA      NA      NA      NA
## 2      NA      NA      NA      NA      NA      NA      NA      NA      NA
## 3      NA      NA      NA      NA      NA      NA      NA      NA      NA
## 4      NA      NA      NA      NA      NA      NA      NA      NA      NA
## 5      NA      NA      NA      NA      NA      NA      NA      NA      NA
## 6      NA      NA      NA      NA      NA      NA      NA      NA      NA
##   R05A.19 R05A.20 Output
## 1      NA      NA     NA
## 2      NA      NA     NA
## 3      NA      NA     NA
## 4      NA      NA     NA
## 5      NA      NA     NA
## 6      NA      NA     NA
```

Each column represents a touchpoint or channel and the each row represents the number of customers that had the corresponding touchpoint in their customer journey.

Taking a quick look at the data we understand that it needs to be wrangled to make the data ready into a standard format before using Markov Models:

Bring the data into the required standard format

```r
library(tidyr)
channel$path = unite(channel, " > ", remove = T)

new_data = channel$path
colnames(new_data) = c("path")
new_data = cbind(channel[, 1:21], new_data)

library(stringr)
new_data$path = str_replace_all(new_data$path, "_", " > ")

head(new_data$path)
```

```
## [1] "16 > 4 > 3 > 5 > 10 > 8 > 6 > 8 > 13 > 20 > 21 > NA > NA > NA > NA > NA > NA > NA > NA > NA > N
## [2] "2 > 1 > 9 > 10 > 1 > 4 > 3 > 21 > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > N
## [3] "9 > 13 > 20 > 16 > 15 > 21 > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > N
## [4] "8 > 15 > 20 > 21 > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > N
## [5] "16 > 9 > 13 > 20 > 21 > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > N
## [6] "1 > 11 > 8 > 4 > 9 > 21 > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA > NA >
```

```r
new_data$conversion = ifelse(str_detect(new_data$path, "21"), 1, 0)

df = as.data.frame(str_split_fixed(new_data$path,  "> 21", 2))
colnames(df) = c("new_path", "V2")

new_data = cbind(new_data, df$new_path)
names(new_data)[names(new_data) == "df$new_path"] = "new_path"

df = new_data[, c("new_path", "conversion")]

head(df)
```

```
##                                     new_path conversion
## 1 16 > 4 > 3 > 5 > 10 > 8 > 6 > 8 > 13 > 20          1
## 2                2 > 1 > 9 > 10 > 1 > 4 > 3          1
## 3                     9 > 13 > 20 > 16 > 15          1
## 4                               8 > 15 > 20          1
## 5                          16 > 9 > 13 > 20          1
## 6                        1 > 11 > 8 > 4 > 9          1
```

Aggregating the data for every path to summarise the conversions

```r
library(dplyr)
df_grouped = df %>%
  select(names(df)) %>%
  group_by(new_path) %>%
  summarise(conversion = sum(conversion))

head(df_grouped)
```

```
## # A tibble: 6 x 2
##   new_path                        conversion
##   <fct>                                <dbl>
## 1 "1 > 1 > 1 > 20 "                        1
## 2 "1 > 1 > 12 > 12 "                       1
## 3 "1 > 1 > 14 > 13 > 12 > 20 "             1
## 4 "1 > 1 > 3 > 13 > 3 > 20 "               1
## 5 "1 > 1 > 3 > 17 > 17 "                   1
## 6 "1 > 1 > 6 > 1 > 12 > 20 > 12 "          1
```

New-Path variable indicates the customer journey path before a transaction.

This is the final format of the data that now goes into the modeling.

creating Heuristic Models

```r
library("ChannelAttribution")
Heuristic = heuristic_models(df_grouped, 'new_path', 'conversion', var_value='conversion')
```

creating Markov Model

```r
library("ChannelAttribution")
Markov = markov_model(df_grouped, 'new_path', 'conversion', var_value='conversion', order = 1, seed = 12
```

Let us merge the both models to get a more clearer picture

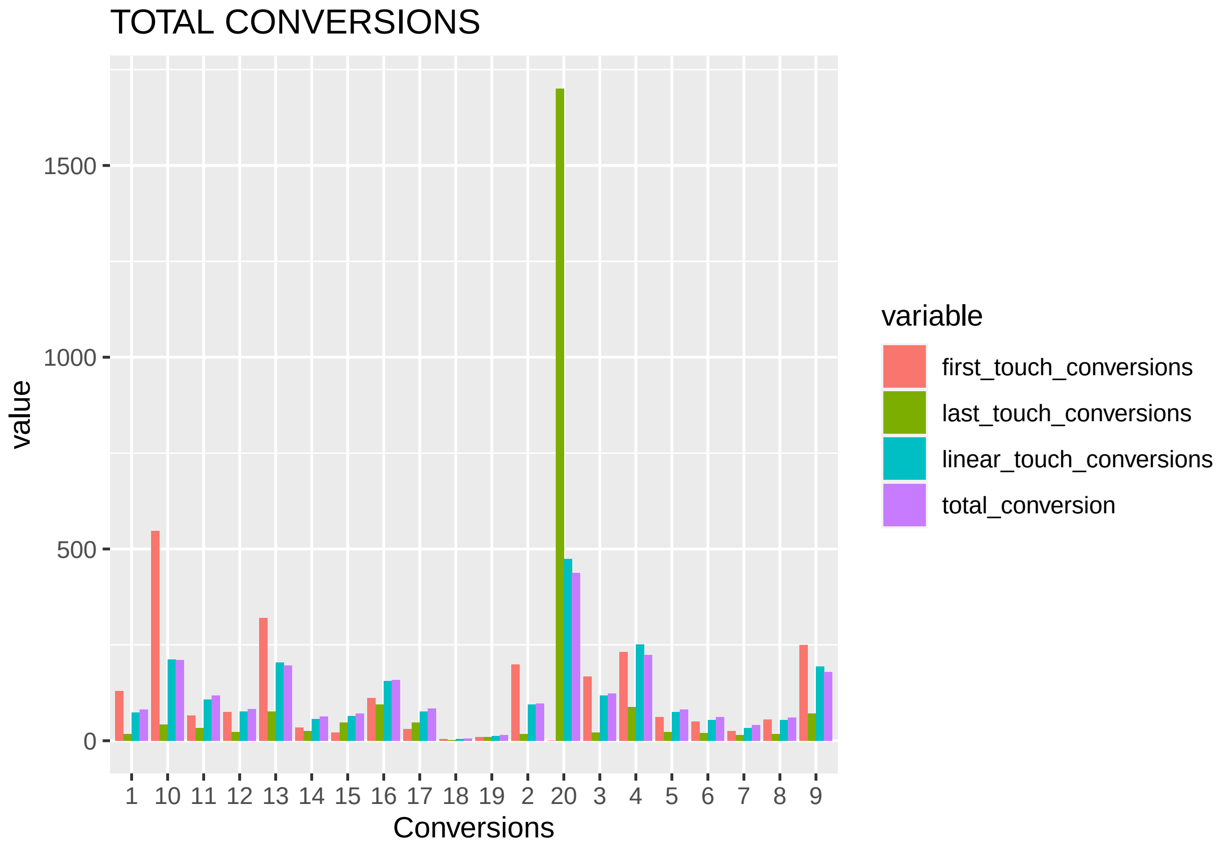
Data Visulization

```r
R = merge(Heuristic, Markov, by='channel_name')

R1 = R[, c("channel_name", "first_touch_conversions", "last_touch_conversions",
                              "linear_touch_conversions", "total_conversion")]

library("reshape2")
R2 = melt(R1, id = "channel_name")

# Plot the total conversions
library("ggplot2")
ggplot(R2, aes(channel_name, value, fill = variable)) +
  geom_bar(stat='identity', position='dodge') +
  ggtitle('TOTAL CONVERSIONS') +
  xlab("Conversions")
```



The plot above helps us understand that from a first touch conversion perspective, channels 10, 13 and 9 are the three most important. From a last touch conversion perspective, channela 20, 4 and 9 are the three most important.

From a linear touch conversion perspective, channels 20, 4 and 9 come out important. From the total conversions perspective, channels 10, 13, 20, 4 and 9 are quite important.

The above case gives us a good insight into the application of Channel Attribution problem using R.