Ads Analysis

Mitul Shah

8/8/2017

Loading the data

```
## Loading the data
ad_table <- read.csv("ad_table.csv")
```

Checking Data Quality

```
## Load the library dplyr
library(dplyr)
```

```
##
## 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
```

```
## Looking into possible data quality issues
unclean_data <- filter(ad_table, shown == 0 | clicked == 0 | converted == 0 | avg_cost_per_click == 0 |

## Remove ad group 25 (since converted variable is 0 for all rows in group 25; it's a bug)
ad_table <- filter(ad_table, ad != "ad_group_25")

## Replace zeroes with mean of value 1 row above and 1 row below for all columns
for(i in 2:nrow(ad_table)) {
  for(j in 2:6) {
    if(ad_table[[i, j]] == 0) {
      ad_table[[i, j]] = (ad_table[[i - 1, j]] + ad_table[[i + 1, j]])/2
    } else {
      ad_table[[i, j]] = ad_table[[i, j]]
    }
  }
}

## Convert the class of date variable to date
ad_table$date <- as.Date(ad_table$date)
```

```
## Warning in strptime(xx, f <- "%Y-%m-%d", tz = "GMT"): unknown timezone
## 'zone/tz/2018c.1.0/zoneinfo/America/Los_Angeles'
```

If you had to identify the 5 best ad groups, which ones would they be? Which metric did you choose to identify the best ad groups? Why? Explain the pros of your metric as well as the possible cons.

The goal here is to get more customers, i.e. increase engagement. We can use click through rate (CTR) to identify the best ad groups here.

Pros:

1. Getting more traffic
2. Increase Quality Scores

Cons:

1. Higher CTR only helps in getting more traffic; doesn't necessarily mean high quality traffic (which lead to high conversion rate)
2. CTR doesn't account for ad frauds this means that the click could come from a willing human, a bot, or could just be down to human error. Take mobile devices, for example, on which 'fat fingers' account for 50% of mobile clicks.

CPI (Conversion per Impression) is a better metric to use than CTR when your goal is to get high quality traffic.

```
## Load library
library(dplyr)

## CTR
data_to_find_best_ad_groups <- ad_table %>% group_by(ad) %>% summarise(shown = sum(shown), clicked = su

## Top 5 ad groups
head(select(data_to_find_best_ad_groups, ad, ctr), n = 5)
```

```
## # A tibble: 5 x 2
##            ad        ctr
##        <fctr>      <dbl>
## 1 ad_group_18 0.09895876
## 2  ad_group_3 0.09496853
## 3 ad_group_19 0.09411519
## 4 ad_group_26 0.09098659
## 5 ad_group_28 0.08904567
```

For each group, predict how many ads will be shown on Dec, 15 (assume each ad group keeps following its trend).

```
## Load dplyr library
library(dplyr)

## Subset only Ads Shown
ads_shown <- select(ad_table, date, shown, ad)
```

Cluster ads into 3 groups: the ones whose avg_cost_per_click is going up, the ones whose avg_cost_per_click is flat and the ones whose avg_cost_per_click is going down.

```
## Load dplyr library
library(dplyr)

## Subset data having only date, average cost per click and ad
data_to_cluster <- select(ad_table, date, avg_cost_per_click, ad)

## Loading tidyr
library(tidyr)

## Converting video count data to wide format
wide_data_to_cluster <- reshape(data_to_cluster, idvar = "ad", timevar = "date", direction = "wide")

## Replace missing values
for(i in 1:nrow(wide_data_to_cluster)) {
  for(j in 2:ncol(wide_data_to_cluster)) {
    if(is.na(wide_data_to_cluster[[i, j]] == TRUE)) {
      wide_data_to_cluster[[i, j]] = (wide_data_to_cluster[[i, j - 1]] + wide_data_to_cluster[[i, j + 1
    } else {
      wide_data_to_cluster[[i, j]] = wide_data_to_cluster[[i, j]]
    }
  }
}

## Calculate proportion of days with increase

## Initialize proportion of days with increase
wide_data_to_cluster$proportion_of_days_with_increase <- 0

## Calculate number of days with increase in avg cost per click
for(i in 1:nrow(wide_data_to_cluster)) {
  for(j in 2:(ncol(wide_data_to_cluster)-2)) {
    if(wide_data_to_cluster[[i, j]] < wide_data_to_cluster[[i, j+1]]) {
      wide_data_to_cluster[[i, 55]] = wide_data_to_cluster[[i, 55]] + 1
    } else {
      wide_data_to_cluster[[i, 55]] = wide_data_to_cluster[[i, 55]]
    }
  }
}

## Divide the number of days by total days
wide_data_to_cluster$proportion_of_days_with_increase <- wide_data_to_cluster$proportion_of_days_with_i

## k-means to cluster ads
kmeans(wide_data_to_cluster[,55], 3, alg="Lloyd")
```

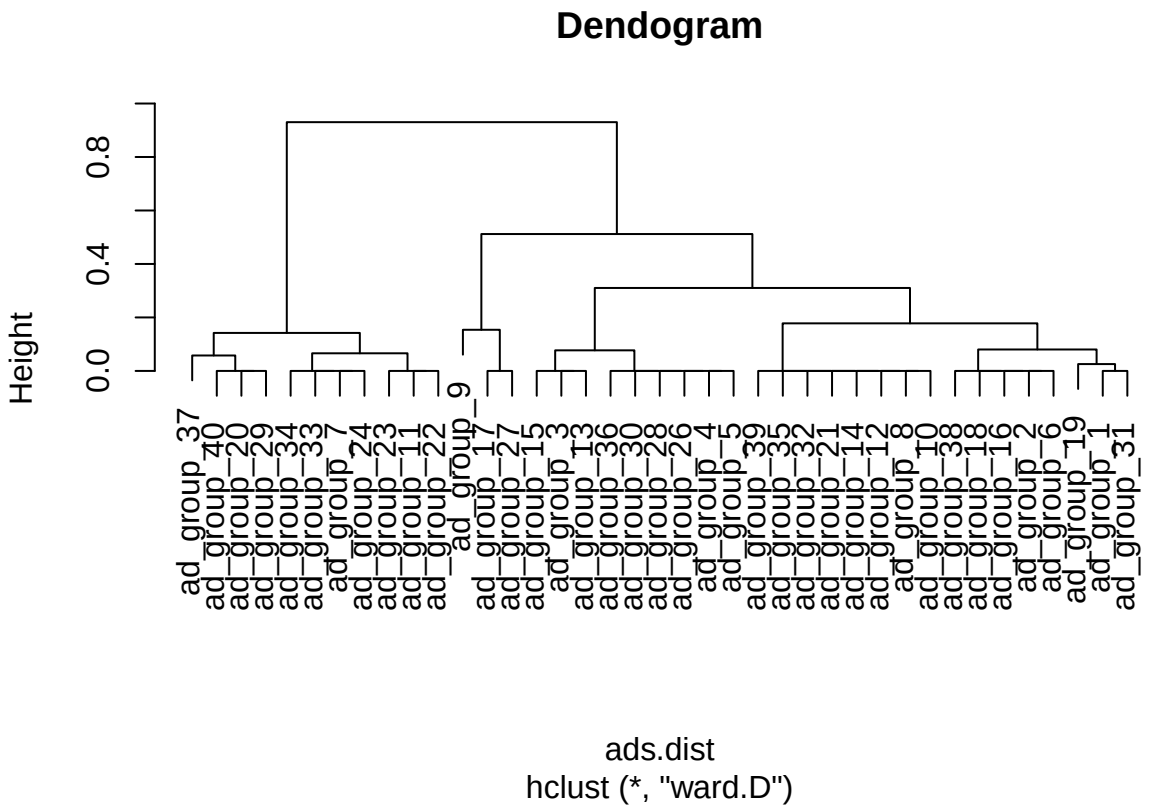
```
## K-means clustering with 3 clusters of sizes 14, 22, 3
##
## Cluster means:
```

```
##        [,1]
## 1 0.5329670
## 2 0.4562937
## 3 0.3269231
##
## Clustering vector:
##  [1] 2 2 1 2 2 2 1 2 3 2 1 2 1 2 1 2 3 2 2 1 2 1 1 1 2 3 2 1 2 2 2 1 1 2 2
## [36] 1 2 2 1
##
## Within cluster sum of squares by cluster:
## [1] 0.009932375 0.009749866 0.008875740
##  (between_SS / total_SS =  80.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
## Distance matrix
ads.dist <- dist(wide_data_to_cluster[,55], method = "euclidean", diag = FALSE, upper = FALSE, p = 2)

## Hierarchical Clustering using Ward's method
ads.hclust <- hclust(ads.dist, method = "ward.D")

## Visualize the dendogram
plot(ads.hclust, labels = wide_data_to_cluster$ad, main='Dendogram')
```



```r
## Forming 3 clusters
groups.3 <- cutree(ads.hclust, 3)

## Looking at the items in all 12 clusters
sapply(unique(groups.3), function(g)wide_data_to_cluster$ad[groups.3 == g])
```

```
## [[1]]
##  [1] ad_group_1  ad_group_2  ad_group_3  ad_group_4  ad_group_5
##  [6] ad_group_6  ad_group_8  ad_group_10 ad_group_12 ad_group_13
## [11] ad_group_14 ad_group_15 ad_group_16 ad_group_18 ad_group_19
## [16] ad_group_21 ad_group_26 ad_group_28 ad_group_30 ad_group_31
## [21] ad_group_32 ad_group_35 ad_group_36 ad_group_38 ad_group_39
## 40 Levels: ad_group_1 ad_group_10 ad_group_11 ad_group_12 ... ad_group_9
##
## [[2]]
##  [1] ad_group_7  ad_group_11 ad_group_20 ad_group_22 ad_group_23
##  [6] ad_group_24 ad_group_29 ad_group_33 ad_group_34 ad_group_37
## [11] ad_group_40
## 40 Levels: ad_group_1 ad_group_10 ad_group_11 ad_group_12 ... ad_group_9
##
## [[3]]
## [1] ad_group_9  ad_group_17 ad_group_27
## 40 Levels: ad_group_1 ad_group_10 ad_group_11 ad_group_12 ... ad_group_9
```