Pebble Post Problem Answer

Xiaosheng Luo

August 17, 2018

This RMarkdown document is for report purpose only. All codes can be checked in github.

Read Data

1. Manually download data from Google Drive into local disk. (An R package "googledrive" can automate this step)
2. Read csv files, then combine them.

```
# obtain file names
filenames_cookie <- dir('data/raw/cookie_match_sample', full.names = T)
filenames_event <- dir('data/raw/event_sample', full.names = T)

# read csv files into one dataframe
cookie_match <- do.call(rbind, lapply(filenames_cookie, read.csv,
                                      stringsAsFactors = F,
                                      colClasses = c('character', 'character', 'Date')))

event <- do.call(rbind, lapply(filenames_event, read.csv,
                               stringsAsFactors = F,
                               colClasses = c(rep('character', 4), 'Date')))

# save as R binary data files to fast read data again if something happens
saveRDS(cookie_match, 'data/processed/cookie_match.Rds')
saveRDS(event, 'data/processed/event.Rds')
# cookie_match <- readRDS('data/processed/cookie_match.Rds')
# event <- readRDS('data/processed/event.Rds')
```

3. Merge data to find matched events

```
# find all matched events
matched_events <- event %>%
  inner_join(cookie_match, by = c('ppid', 'date'))
```

Question 1

What are the number of events for each brand for each day?

```
# number of events/brand/day
num_events <- event %>%
  count(brand_id, date)

kable(num_events)
```

brand_id	date	n
1034	2018-03-15	1679
1034	2018-03-16	1488

brand_id	date	n
1034	2018-03-17	1165
1034	2018-03-18	1257
1034	2018-03-19	1838
1034	2018-03-20	1678
1034	2018-03-21	1484
1034	2018-03-22	1471
1034	2018-03-23	1471
1034	2018-03-24	1220
1034	2018-03-25	1403
1034	2018-03-26	1939
1034	2018-03-27	1609
1034	2018-03-28	1554
1101	2018-03-15	12638
1101	2018-03-16	12625
1101	2018-03-17	12830
1101	2018-03-18	14856
1101	2018-03-19	14155
1101	2018-03-20	13916
1101	2018-03-21	13917
1101	2018-03-22	13482
1101	2018-03-23	13219
1101	2018-03-24	12763
1101	2018-03-25	14432
1101	2018-03-26	13503
1101	2018-03-27	12952
1101	2018-03-28	12346
1472	2018-03-15	82546
1472	2018-03-16	82397
1472	2018-03-17	82523
1472	2018-03-18	95790
1472	2018-03-19	97745
1472	2018-03-20	92943
1472	2018-03-21	93668
1472	2018-03-22	85006
1472	2018-03-23	78230
1472	2018-03-24	72791
1472	2018-03-25	86516
1472	2018-03-26	83225
1472	2018-03-27	74411
1472	2018-03-28	73309

What are the number of matched events for each brand for each day?

```r
# number of Matched events/brand/day
num_match_events <- matched_events %>%
  inner_join(cookie_match, by = c('ppid', 'date')) %>%
  count(brand_id, date)

kable(num_match_events)
```

brand_id	date	n

brand_id	date	n
1034	2018-03-15	25
1034	2018-03-16	19
1034	2018-03-17	6
1034	2018-03-18	8
1034	2018-03-19	14
1034	2018-03-20	12
1034	2018-03-21	85
1034	2018-03-22	11
1034	2018-03-23	77
1034	2018-03-24	9
1034	2018-03-25	6
1034	2018-03-26	18
1034	2018-03-27	8
1034	2018-03-28	11
1101	2018-03-15	469
1101	2018-03-16	508
1101	2018-03-17	493
1101	2018-03-18	634
1101	2018-03-19	452
1101	2018-03-20	464
1101	2018-03-21	478
1101	2018-03-22	383
1101	2018-03-23	412
1101	2018-03-24	430
1101	2018-03-25	557
1101	2018-03-26	415
1101	2018-03-27	457
1101	2018-03-28	1638
1472	2018-03-15	76917
1472	2018-03-16	93598
1472	2018-03-17	82869
1472	2018-03-18	85015
1472	2018-03-19	80829
1472	2018-03-20	88260
1472	2018-03-21	81874
1472	2018-03-22	72562
1472	2018-03-23	56429
1472	2018-03-24	78506
1472	2018-03-25	101238
1472	2018-03-26	124461
1472	2018-03-27	63626
1472	2018-03-28	51861

Question 2

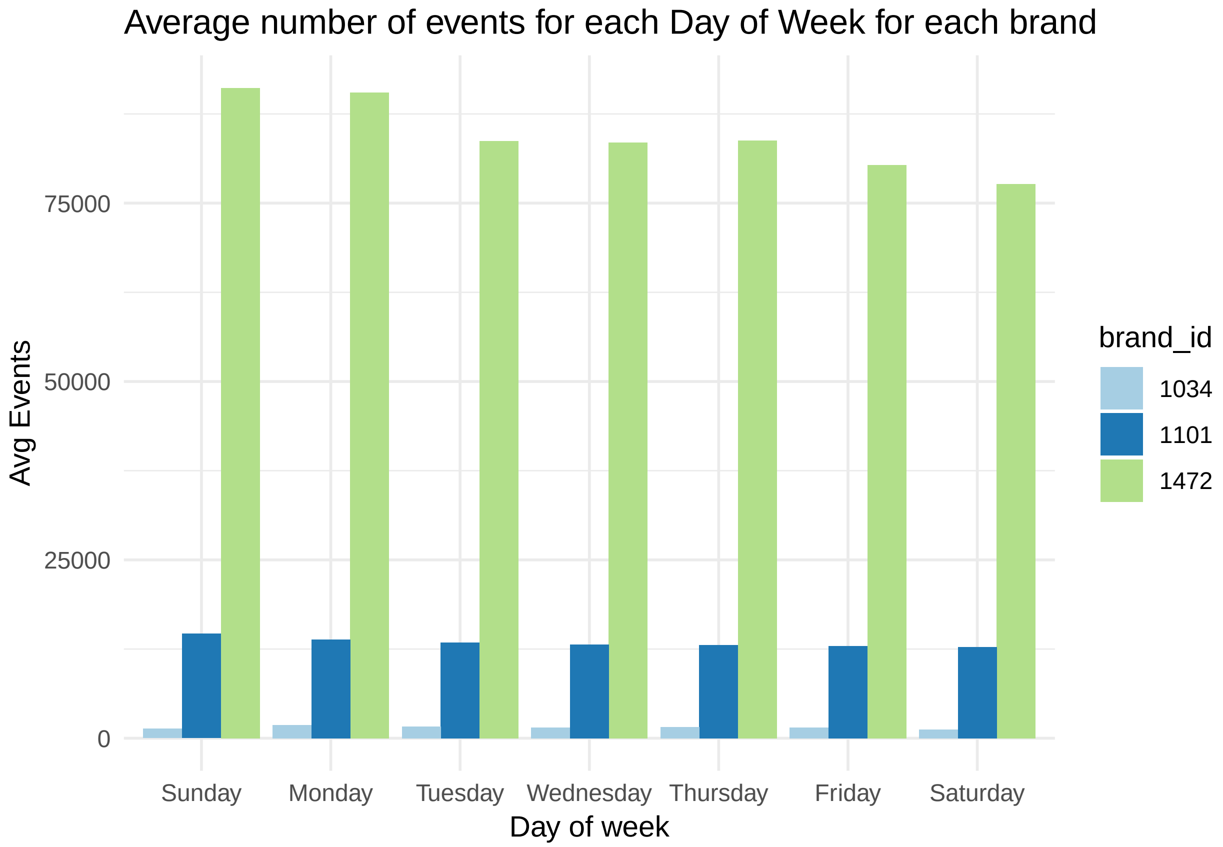
What is the average number of events for each Day of Week for each brand? Can you create a graph or plot to visualize this information?

```r
# create new column for day of week
num_events <- num_events %>%
  mutate(day = weekdays(date))

# calculate average events per day of week
avg_events_dayofweek <- num_events %>%
  group_by(brand_id, day) %>%
  summarise(avg_evnts = mean(n))

# plot
week <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
avg_events_dayofweek$day <- factor(avg_events_dayofweek$day, levels = week)

ggplot(avg_events_dayofweek, aes(x=day, y=avg_evnts, fill=brand_id)) +
  geom_bar(stat='identity', position=position_dodge()) +
  scale_fill_brewer(palette="Paired") +
  theme_minimal() +
  xlab('Day of week') +
  ylab('Avg Events') +
  ggtitle('Average number of events for each Day of Week for each brand')
```



Question 3

If you have any interesting observations about the sample data, please describe them here.

```r
str(event)
```

```
## 'data.frame':    1389990 obs. of  5 variables:
##  $ brand_id     : chr  "1472" "1472" "1472" "1472" ...
##  $ ppid         : chr  "35cd25e7-0edd-4094-8933-fe69adfc8dd1" "c6572797-d795-4d87-8964-8eb0bd0b01cf"
##  $ event_type   : chr  "seg" "seg" "seg" "seg" ...
##  $ device_family: chr  "Desktop" "Desktop" "Desktop" "Phone" ...
##  $ date         : Date, format: "2018-03-20" "2018-03-20" ...
```

```r
# unique values in event_type & device_family
unique(event$event_type)
```

```
## [1] "seg"  "conv"
```

```r
unique(event$device_family)
```

```
## [1] "Desktop" "Phone"   "Tablet"  "Other"   ""
```

```r
# Total events of each brand
table(event$brand_id)
```

```
## 
##    1034    1101    1472 
##   21256  187634 1181100
```

```r
# Total events of each type
table(event$event_type)
```

```
## 
##    conv     seg 
##    3214 1386776
```

```r
# Total events of each device family
table(event$device_family)
```

```
## 
##         Desktop   Other   Phone  Tablet 
##       2 1200507     806  147638   41037
```

Note that there are duplicates both in event and cookie_match df, I will use ppid "78C1840A54EE7F57EE1622290236DC03" as an example. Possible reason is the customer visit the web mutiple times per day.

```r
# duplicates in event
event %>% filter(ppid == '78C1840A54EE7F57EE1622290236DC03') %>% kable()
```

brand_id	ppid	event_type	device_family	date
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-20
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-20
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-20
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-20
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-19
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-19
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-19
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-20
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-20
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23

brand_id	ppid	event_type	device_family	date
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-22
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-23
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-21
1472	78C1840A54EE7F57EE1622290236DC03	seg	Desktop	2018-03-15

```r
# duplicateds in cookie_match
cookie_match %>% filter(ppid == '78C1840A54EE7F57EE1622290236DC03') %>% kable()
```

ppid	matched_id	date
78C1840A54EE7F57EE1622290236DC03	AB7AE8BE80B83DC361D6AD726140E9726BC970BC	2018-03-20
78C1840A54EE7F57EE1622290236DC03	AB7AE8BE80B83DC361D6AD726140E9726BC970BC	2018-03-22
78C1840A54EE7F57EE1622290236DC03	AB7AE8BE80B83DC361D6AD726140E9726BC970BC	2018-03-23
78C1840A54EE7F57EE1622290236DC03	AB7AE8BE80B83DC361D6AD726140E9726BC970BC	2018-03-20

Question 4

How do you test the difference between the conversion rates for test group and control group is statistically significant or not?

```r
# create contingency table
mat <- matrix(c(500, 200, 10000, 5000), nrow = 2,
             dimnames = list(c('test', 'control'),
                             c('converted', 'not_converted')))
chisq.test(mat)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  mat
## X-squared = 6.633, df = 1, p-value = 0.01001
```

p-value = 0.01001, reject null hypothesis. It gives the strong evidence to suggest that test and control group have statistical significance difference.

What if the test group has 10000 users and 2 converters, and the control group has 4000 users and 1 converter?

Sample size is too small to be statistically significant.

Question 5

1. How will you formulate the problem?

Here, the outcome(converted, not_converted) that we want to predict is binary categorical outcome, and the label variable is given. So, this is a supervise learning classification problem. I feel this analysis is sensitive to date, I may consider to obtain the day of week, seasonality, holiday as well. In other words, this may consider as a time series problem as well.

2. What users will you use as training and testing examples?

For this two week data, I will use first 10 days data as the training dataset, the rest as testing.

3. What user features/data do you plan to collect?
    - Demographic data like age, gender, income, education, employment and etc.
    - Social media data like facebook, twitter, and etc.
    - Conversion history.
    - Day and time when the user access the brand web.
    - Device information.
    - Number of times that the user visited the brand web.
4. How will you preprocess the collected data to generate input for your system?

Basiclly, cleaning, transforming. Not going to go deep in cleaning, because it may vary depends on different data. For transforming,

- Create variables. For example, I may create zipcode group if user zipcode is available. This step is based on the marketing expertise suggestion as well as each sub-group contains at least enough events(I prefer 10 cases) to build the model.
- Create dummy variables.
- Imputation variables.
- Scaleing and centering.
- Run PCA/MCA analysis to try to get insights of the features, reduce the dimensions of the features. This step include remove correlated features, remove zero/near-zero variance features.

5. What algorithm(s) to use and why?

All the algorithms that I pick will friendly for binary category outcome, mixed feature types supervise learning classification problem. I will try use elastic regression first, since the training time was the fastest one. Then I can get an general view of the model. Then will try use logistic regression, random forest, svm, and etc.

6. How will you evaluate the performance?

Cross-validation while tring the model and use AUC, confusion matrix(accuracy, sensitivity, specificity) to evaluate the model as well as evaluate the performance while using the test dataset.

Question 6

Continuous from built model from Question 5, the uplift modeling's general Steps:

1. Predict the outcome on the promotional item applied users.
2. Predict the outcome on the no promotional item applied users.
3. Find the uplift as the difference in the rates (step 1 - step 2).
4. Find upper and lower confidence limits on the uplift.

Results:

- If confidence limits of the uplift includes zero. The promotion effect is unknow and not significant.
- If confidence limits of the uplift significantly greater than zero, those are swing user.
- If confidience limits of the uplift significantly less than zero, those are the no purchase user.

The uplift package in R can handle this type of modeling.