CAL Project - Knowledge Transfer Documentation - Data Engineering 4.0

4/28/2020

This file includes the data engineering steps taken on the Events and Email files in order to build predictive models. Its output is utilized in both "4.1 - Predictive Model Prep Existing Member" and "4.2 - Predictive Model Prep New-Member".

Please contact musch.sam@gmail.com if you have any questions.

Introduction

Load packages and read files.

```
library(tidyverse)
library(data.table)
library(dtplyr)
library(lubridate)

indy <-
  read_csv('../individual_info.csv') %>%
  select(ID_DEMO, MEMBERSHIP_STATUS_CODE)

emails <- fread('../emails_cleaned.csv')

events <- read_csv('../events_cleaned.csv') %>%
  filter(YEAR_FISCAL > 2014) %>%
  inner_join(indy, by = 'ID_DEMO')
```

Emails

We are only looking at emails since 2015 and only the important categories. We are also creating columns that will allow us to create click-through-rate and click_open rate.

```
emails <- emails %>%
  filter(YEAR_FISCAL > 2014) %>%
  merge.data.table(indy, by = 'ID_DEMO') %>%
  filter(broad_cat %in%
         c('Learning', 'Legislature',
           'Social', 'Sports')) %>%

  # Creating 2 new columns
  mutate(ctr =
         ifelse(CODE_OUTCOME == 'CL', 1, 0)) %>%
  mutate(click_open =
         ifelse(CODE_OUTCOME == 'CL' |
                CODE_OUTCOME == 'OE', 1, 0))
```

We are calculating a person's click_open rate for **every category**.

```
clickthru_rates <-
  emails %>%
    group_by(ID_DEMO, broad_cat) %>%
    dplyr::summarize(total_possible = n(),
                   total_clicked = sum(click_open)) %>%
    mutate(click_or_open = total_clicked / total_possible)
```

We are taking each email category and making each one a separate column.

This chunk of code looks very confusing so I will walk thru each piece. Note that this does the same thing as the spread function but can handle more than 1 column.

- reshape2::dcast
    - we are using the package "reshape2" and "dcast" lets us take each email category and make it its own column
- ID_DEMO ~ broad_cat
    - We are keeping ID_DEMO as the primary key, but taking the categories from broad_cat and using them to create new columns
- value.var
    - This is the value we calculated in the previous chunk
- We are adding "emails" at the end of each of our email categories

```
clickthru_rates_spread <-
  reshape2::dcast(data = clickthru_rates,
                ID_DEMO ~ broad_cat,
                value.var = 'click_or_open',
                fun=sum) %>%
  rename_at(vars(-ID_DEMO), ~ paste0(., '_emails')) %>%
replace(is.na(.), 0)
```

This is a person's **general** click-through-rate.

```
clickthru_rates_general <-
  emails %>% group_by(ID_DEMO) %>%
  dplyr::summarize(total_possible = n(),
                   total_clicked = sum(ctr)) %>%
  mutate(general_ctr = total_clicked / total_possible) %>%
  select(ID_DEMO, general_ctr)
```

This joins the person's general click-thru-rate with each of their category specific click_open rates.

```
clickthru_rates_spread <-
  clickthru_rates_spread %>%
  inner_join(clickthru_rates_general, by = 'ID_DEMO')
```

Events

This just counts how many events a person has gone to (per-category).

```
events_adj <-
  events %>%
  count(ID_DEMO, broad_cat) %>%
  rename(total_type_person = n)
```

This is the same operation we used for emails. Each person will get 1 row, and each event category will become a column.

```
events_spread <-
  reshape2::dcast(data = events_adj,
                  ID_DEMO ~ broad_cat,
                  value.var = 'total_type_person',
                  fun=sum) %>%
  rename_at(vars(-ID_DEMO), ~ paste0(., '_events')) %>%
  replace(is.na(.), 0)
```

This calculates a person's total events attending by summing up the category-specific events.

```
events_spread <-
  events_spread %>%
  mutate(all_events = rowSums(.[2:7]))
```

Per-person

This joins together the person's information from emails & events so we can include them in the predictive model.

```r
# Connecting emails & events
per_person_with_id <-
  clickthru_rates_spread %>%
  full_join(events_spread, by = c('ID_DEMO')) %>%
  ungroup() %>%
  inner_join(indy, by = 'ID_DEMO') %>%
  select(-MEMBERSHIP_STATUS_CODE) %>%
  replace(is.na(.), 0)


setwd("D:/Group Folder/Data")
fwrite(per_person_with_id, file='pred_emails_events.csv')
```