Relax, Inc. Data Science Challenge

Summary

Initially, the most important feature to predict adoption is the length of time between account creation and the most recent login. This makes sense; more time spent with an active account means more time to login three times in a week. With that feature and related features removed, the most important factors to predict adoption are:

1. opted_in_to_mailing_list – People on the mailing list are more likely to become adopted users.
2. enabled_for_marketing_drip – The effect of being on the marketing drip list are hard to quantify.
3. gmail.com – Gmail users are more likely to adopt.
4. Users with temporary email addresses such as cuvox.de, gustr.com, and jourrapide.com are less likely to become adopted users.
5. People who create accounts in the beginning or middle of the month are more likely to become adopted users, while people who create accounts at the end of the month are less likely to become adopted users.
6. hotmail.com – Hotmail users are more likely to adopt.
7. People who create accounts toward the end of the year, August or later, are more likely to become adopted users than those who create accounts before August.

Methodology

Data from the .csv files was loaded into Python. The engagement DataFrame was used to create the adopted target feature:

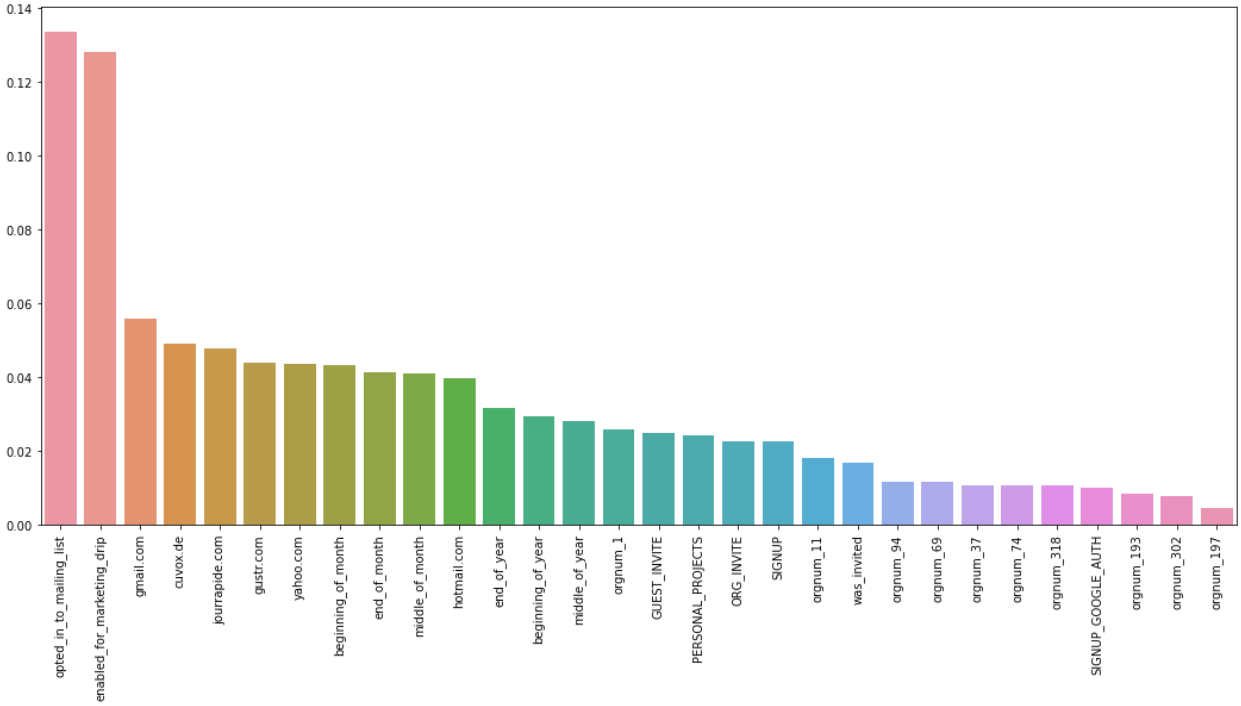
1. Grouped by user_id
2. Resampled to daily logins
3. Created a 7-day rolling window
4. Counted number of logins in that rolling window
5. If the user logged in more than 3 times in any 7-day rolling window, set adopted = 1
6. Attach the adopted column to the users DataFrame

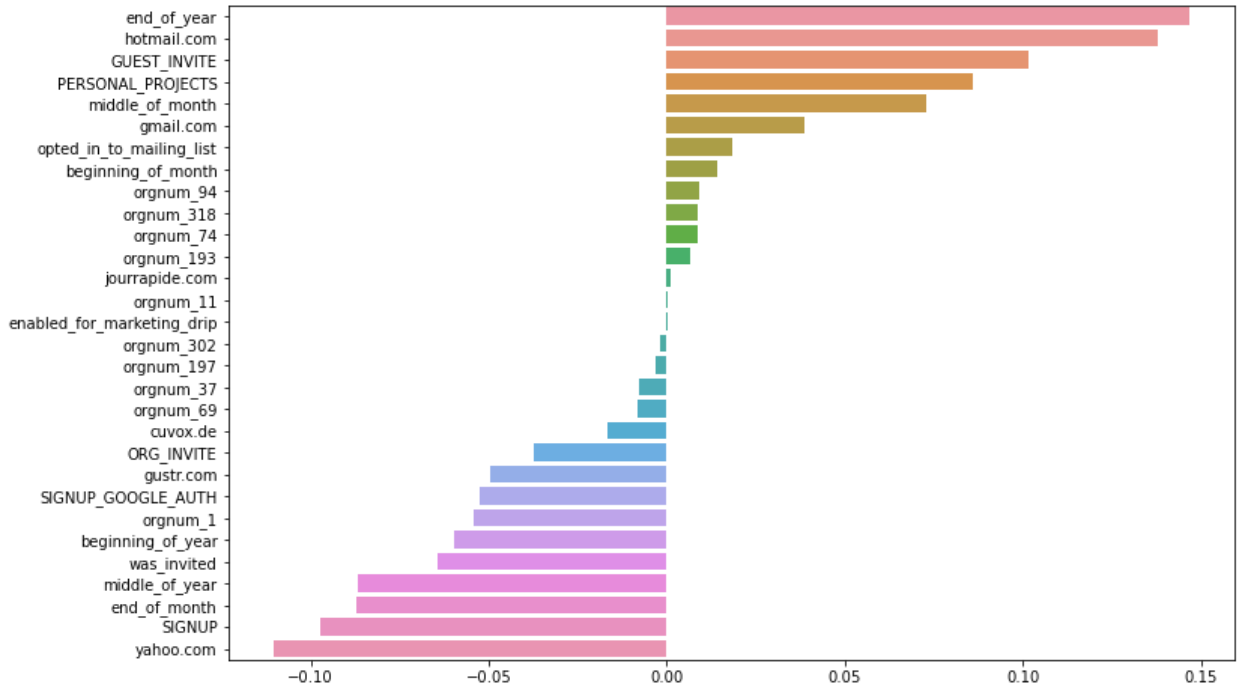
Feature Engineering in the users DataFrame:

1. Drop irrelevant features
2. Grab the suffix from email
3. Get number of days between creation_time and last_session_creation_time
4. Separate creation_time into creation_year, creation_month, creation_day
    a. Bin creation_day into beginning_of_month, middle_of_month, and end_of_month
    b. Bin creation_month into beginning_of_year, middle_of_year, and end_of_year
5. Get the large org_ids (with more than 100 members)
6. Create a was_invited column
7. One-Hot Encode all class variables (which was all of them)
8. Drop observations with missing values

Then I fit a random forest model, tuning hyperparameters to maximize Recall, and grab the feature importances. The time between account creation and last_session was so important that I decided to drop that feature and fit a new model. Then, creation_day and creation_month were so important that I binned them into beginning, middle, and end categories. After getting the final graph of feature importance, I fit a logistic regression model to see which features make adoption more likely, and which make adoption less likely. See the related Jupyter Notebook for a detailed walkthrough and code.

The plots on page 2 are of the final feature importances and the feature coefficients, respectively.


0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
opted_in_to_mailing_list
enabled_for_marketing_drip
gmail.com
cuvox.de
jourrapide.com
gustr.com
yahoo.com
beginning_of_month
end_of_month
middle_of_month
hotmail.com
end_of_year
beginning_of_year
middle_of_year
orgnum_1
GUEST_INVITE
PERSONAL_PROJECTS
ORG_INVITE
SIGNUP
orgnum_11
was_invited
orgnum_94
orgnum_69
orgnum_37
orgnum_74
orgnum_318
SIGNUP_GOOGLE_AUTH
orgnum_193
orgnum_302
orgnum_197


end_of_year
hotmail.com
GUEST_INVITE
PERSONAL_PROJECTS
middle_of_month
gmail.com
opted_in_to_mailing_list
beginning_of_month
orgnum_94
orgnum_318
orgnum_74
orgnum_193
jourrapide.com
orgnum_11
enabled_for_marketing_drip
orgnum_302
orgnum_197
orgnum_37
orgnum_69
cuvox.de
ORG_INVITE
gustr.com
SIGNUP_GOOGLE_AUTH
orgnum_1
beginning_of_year
was_invited
middle_of_year
end_of_month
SIGNUP
yahoo.com
−0.10
−0.05
0.00
0.05
0.10
0.15