# 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
   1. Bin creation\_day into beginning\_of\_month, middle\_of\_month, and end\_of\_month
   2. 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.



