Data Science Challenge for Relax, Inc.

In this challenge, I calculate adopted users, evaluate predictive features, and summarize my findings--all in one page. Plus, I list more details and additional work in the appendices.

Calculate adopted users

I calculate adopted users as those who logged into our product three times within seven days. By my calculation, 1,656 users of our 12,000 users with user_id (i.e., about 14%) are adopted users.

Identify predictive features

I identify predictive features by preparing features, predicting adoption with a random forest, extracting feature importance, predicting adoption with logistic regression, and examining the regression coefficients.

To evaluate features, I use the random forest's importance metric as a proxy for predictive power. This measures how much the feature decreases the average Gini impurity across the forest of decision trees. Simply put, a feature with a higher importance has more predictive power, contains more information and is more useful, all with respect to splitting users into buckets of adopted or not across the random forest. Also, I use logistic regression's coefficients to understand the direction in which features impact adoption, and I use the magnitude of the coefficients from these features, after being scaled, as a (very rough, mind you) cross validation of importance.

Summarize my findings

Time-based user-access features have the most predictive power. The feature of 'months_since_last_session' has the most predictive power via importance, has a negative direction (i.e., more months means less likely to be adopted), and is validated with a logit coefficient of large absolute value. The feature of account age (i.e., 'months_since_creation') has high importance, positive direction, and a logit coefficient of large absolute value as well.

Marketing-email and user-email features have some predictive power but only as much as a random variable. Two marketing features--'mailing_list' and 'marketing_drip'--have medium predictive power (although marketing drip, surprisingly, has a negative direction). Two features from user email--whether user's top-level domain was .com or .de and whether the user's secondary domain was a personal email provided (e.g., @gmail)--have similar predictive power. Nevertheless, these features' importance was approximately that of a random variable.

Features of invited, invitation source and organizational membership have even less predictive power. These seven features have even lower predictive power as indicated by random-forest importance. However, the logit model hints that this conclusion requires more investigation as these features (while implemented a bit differently, mind you) have coefficients of a 'medium' magnitude.

Appendix 1: Results of feature analysis

Here are the results of my models after the run upon which I evaluated the features. For more details, see my notebook titled 17-2-2_challenge_relax_notebook_v2.ipynb.

Feature	Random forest importance	Random forest predictive power	Random forest rank	Logit coefficient	Logit direction	Logit magnitude	Logit rank
months_since_last_session	0.7104	high	1	-27.016	negative	high	1
months_since_creation	0.2531	high	2	5.907	positive	high	2
random	0.0069	med	3	0.011	positive	low	11
opted_in_to_mailing_list	0.0049	med	4	0.039	positive	low	9
enabled_for_marketing_drip	0.0047	med	5	-0.008	negative	low	12
second_level_domain_category	0.0036	med	6	0.034	positive	low	10
top_level_domain_category	0.0033	med	7	-0.004	negative	low	13
source_personal_projects	0.0029	low	8	-1.88	negative	med	6
member_of_an_org	0.0021	low	9	0.251	positive	low	8
source_guest_invite	0.0019	low	10	0.254	positive	low	7
source_org_invite	0.0017	low	11				
source_signup	0.0016	low	12	-2.26	negative	med	3
source_signup_google_auth	0.0015	low	13	-2.232	negative	med	4
invited	0.0014	low	14	-2.174	negative	med	5

Appendix 2: Additional work

There is more work to do, time permitting.

- **Improve the calculation of adopted users.** I would work with my stakeholder to better understand adopted users. Specifically, what behaviors are we looking to measure in our customers? What outcome are we looking to drive? Based on this understanding, I might analyze: Do we care about three logins within 7 days (which I implemented)? Or do we care about three logins each with 24 hours between logins, but all logins within the same 7 day period (another interpretation of the adopted definition). More importantly, I would want to know: Are there better events than login? What's the right number of such events (e.g., more or less than three)? What's the right window (e.g., more or less than seven days)? What happens after "first" adoption (e.g., do users stay adopted)?

- **Determine if this is an exploratory or a predictive problem**. I would work with my stakeholder to understand what we really want to first accomplish. The exploratory problem (i.e., "help me understand the drivers of user adoption") may have different approaches than the predictive problem (i.e., "accurately predict user adoption").

- **Improve the models**. I would work to fully tune, score, and evaluate the predictive models. In general, I want a robust model before I examine what the model says about the predictive powers of any features. In this case, I used mainly-default, little-tuned, simply-scored, and not fully-evaluated models. For example, while the random forest model has sufficient accuracy and a good out-of-bounds score (i.e., both above 95%), it's got a recall problem--with only 80% of the actual adopted users being properly labeled.

- **Communicate the limits of random forest's importance and logit's coefficients.** I would help my stakeholder intuitively understand the limits of how I evaluated features. For example, random forests often overweight the importance of continuous features (e.g., our time-based features) and fail to cleanly allocate importance across groups of correlated features (e.g., our marketing and source features). For example, to compare magnitude across logit's coefficients, we would have to talk through how to best scale the various features.

- **Explore additional models**. There are many classification algorithms which may better fit this problem. With more time, I would explore them.

- **Explore feature importance**. Similarly, there are many approaches to examining feature importance. In this work, I used two of the basic embedded methods. But there are many more filter-based and wrapper-based methods. These include permutation importance, recursive feature elimination, partial dependencies, SHAP, LIME and others.