Predicting Adopted User Factors

# Introduction

The goal of this assignment was to analyze data on users from Asana to identify which factors predict future user adoption. An “adopted user” is defined as a user who has logged into the produt on three separate days in at least one seven-day period.

# Analysis

To predict what factors lead to user adoption I created a RandomForest model. Figure 1 displays the table of coefficients of each predictor when regressed against adopted\_user, which was a 1 if the user met the criteria for being considered adopted and 0 elsewise. Results are as follows.

#### **Analyzing the Data**

Before we make any inferences, we listen to our data by examining all variables in the data.

The main goal of data understanding is to gain general insights about the data, which covers the number of rows and columns, values in the data, datatypes, and Missing values in the dataset.

**shape** – **shape** will display the number of observations(rows) and features(columns) in the dataset

**head()** will display the top 5 observations of the dataset

**data.info()** shows the variables

### **Feature Engineering**

Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modeling. The main goal of Feature engineering is to create meaningful data from raw data.

### **S**tatistics Summary

The information gives a quick and simple description of the data.

Can include Count, Mean, Standard Deviation, median, mode, minimum value, maximum value, range, standard deviation, etc.

Statistics summary gives a high-level idea to identify whether the data has any outliers, data entry error, distribution of data such as the data is normally distributed or left/right skewed

In python, this can be achieved using describe()

describe() function gives all statistics summary of data

**describe()**– Provide a statistics summary of data belonging to numerical datatype such as int, float

user\_eng dataframe seems to be sorted according to the *user\_id* and *time\_stamp* column and there is data for 8823 users out of the total 12000 users registered under the *users* dataframe.

Converting the *last\_session\_creation\_time* column in the users column to datetime.

#### **For determining which users can be labeled as adopted users, we can use groupby and rolling methods to find the users who were active for more than 3 days in any of the 7 day rolling period.**

Graphs

for col in ['opted\_in\_to\_mailing\_list', 'enabled\_for\_marketing\_drip', 'creation\_source\_GUEST\_INVITE',

           'creation\_source\_ORG\_INVITE', 'creation\_source\_PERSONAL\_PROJECTS', 'creation\_source\_SIGNUP',

           'creation\_source\_SIGNUP\_GOOGLE\_AUTH']:

    g = sns.FacetGrid(df\_users, hue = "adopted\_user", height=3, aspect=1.5,)

    g.map(plt.hist, col, alpha=.5, bins = 20)

    g.add\_legend()

nitialize the matplotlib figure and FacetGrid object.

This class maps a dataset onto multiple axes arrayed in a grid of rows and columns that correspond to levels of variables in the dataset. The plots it produces are often called “lattice”, “trellis”, or “small-multiple” graphics.

It can also represent levels of a third variable with the hue parameter, which plots different subsets of data in different colors. This uses color to resolve elements on a third dimension, but only draws subsets on top of each other and will not tailor the hue parameter for the specific visualization the way that axes-level functions that accept hue will.

The basic workflow is to initialize the **[FacetGrid](https://seaborn.pydata.org/generated/seaborn.FacetGrid.html" \l "seaborn.FacetGrid" \o "seaborn.FacetGrid)** object with the dataset and the variables that are used to structure the grid. Then one or more plotting functions can be applied to each subset by calling **[FacetGrid.map()](https://seaborn.pydata.org/generated/seaborn.FacetGrid.map.html" \l "seaborn.FacetGrid.map" \o "seaborn.FacetGrid.map)** or **[FacetGrid.map\_dataframe()](https://seaborn.pydata.org/generated/seaborn.FacetGrid.map_dataframe.html" \l "seaborn.FacetGrid.map_dataframe" \o "seaborn.FacetGrid.map_dataframe)**. Finally, the plot can be tweaked with other methods to do things like change the axis labels, use different ticks, or add a legend. See the detailed code examples below for more information.

PreProcessing Steps

There are too many email domains and most of them seem fake domains so it's good to drop the column entirely. We can also drop the name and object\_id columns.

And for the invited\_by\_user\_id let's convert the NULL values to 0 because the column has a Non Null value only if the creation\_source was a GUEST\_INVITE or a ORG\_INVITE anyways.

For the creation\_time column let's add a column which calculates how old the account is, i.e. the number of days since the account was created.

And since last\_session\_creation\_time can be removed as well because it was in a sense used to create the adopted\_user column.

Let's OneHotEncode the creation\_source column.

We can see that the adopted\_user class is pretty unbalanced because only about 13% of the total 12000 users are adopted.

keyboard\_arrow\_down

### **Let's try to fit Random Forest Regression model and find the feature importance. Since we will be using random forest using trees we don't need to scale any features.**

The top 5 important features seem to be:

1. days\_since\_creation
2. org\_id
3. invited\_by\_user\_id
4. creation\_source\_PERSONAL\_PROJECTS
5. opted\_in\_to\_mailing\_list

Furture possible work: We can also add a feature which calculates the difference between the creation date of the account and the first login of the user.

# Creation source

Since the creation source variable was a categorical variable, I dummied it out and matched each source against GUEST\_INVITE since that creation source type had the highest correlation with adopted\_user. We see that the other creation soucres have negative coefficient estimates with our response variable, indicating that GUEST\_INVITE is more indicative of adoption than these other sources. When analyzing this further, I created a separate regression of adopted\_user against these creation sources (Table 2). The coefficient estimate was most negative for personal projects at -0.1005334 and was minimally positively correlated with Google signups, but exhibited a p-value of close to 1, so we would not want to use this as a positive indicator.

# Emails

User signup for mailing lists exhibited a positive correlation with user adoption. In analyzing further (Table 3), enabled\_for\_marketing\_drip exhibited an estimate of 0.0068191, only slightly higher than opted\_in\_to\_mailing\_list. But again, we see high p-values with these estimates. Thus from this model we cannot make any clear assumptions about the relationship between email marketing and user adoption.

# Time Deltas

Time deltas are defined as follows: creation\_delta = time from account creation to last login and

last\_login\_delta: time from last login to when this report was created (12/9/16).

From the model, we see the coefficient estimates with the time delta variables. The creation\_delta exhibited the highest value, with a p-value approximately 0 indicating almost perfect estimation. Exploring this relationship further, I regressed only the time deltas (Figure 4) againstadopted\_user and found that creation\_delta has an estimate of 0.8385773 and that last\_login\_delta exhibits an estimate of -0.1223327, indicating it has a slightly negative relationship with becoming an adopted user.

# Conclusions & Recommendations

In conclusion, from this data the best predictors of future user adoption is our creation\_delta variable and the creation source to be GUEST\_INVITE. Again, the creation delta variable was constructed as *the time from account creation to the user’s last login*. Intuitively, this makes sense as users who have been using Asana for a longer span of time are more likely to use the product frequently. As users login more, they are likely to

import more information and rely on the platform as a larger part of their day to day tasks, and thus are more likely to be “adopted.” The other predictor variable was creation source as guest invite. Surprisingly personal projects as a creation source did not perform as well as I would have initially thought, but there may be something about the exclusivity and excitement of a friend or colleague inviting you to join the platform that is inticing.

Thus, conclusions would be as follows: get more guest invites sent and keep users logging in frequently (whether this be via push notification, email, or marketing advertisement).

# Further Exploration

Given these results, my next step in analyzing this data would be to explore the relationship of invites/creation source further as well as organization usage. Some questions I have from the data are, *1) If a user was invited by another Asana user, what is the likelihood that a) that initial Asana user is an adopted user and b) that the new user becomes an adopted user?* Another question I have is *2) Does the number of people in your organization make a difference in user adoption?* (i.e. is there a tipping point of X amount of people in your organization are using Asana that a user will be default become an adopted user?)

# Figures

**Least Squares Coefficient Estimates**

Table 1: Coefficient estimates: ALL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr(>|t|) |
| (Intercept) | 0 | 0*.*006 | 0 | 1 |
| creation\_sourceORG\_INVITE | -0*.*013 | 0*.*008 | -1*.*688 | 0*.*091 |
| creation\_sourcePERSONAL\_PROJECTS | -0*.*016 | 0*.*007 | -2*.*227 | 0*.*026 |
| creation\_sourceSIGNUP | -0*.*017 | 0*.*007 | -2*.*313 | 0*.*021 |
| creation\_sourceSIGNUP\_GOOGLE\_AUTH | -0*.*013 | 0*.*007 | -1*.*863 | 0*.*063 |
| opted\_in\_to\_mailing\_list | 0*.*001 | 0*.*006 | 0*.*165 | 0*.*869 |
| enabled\_for\_marketing\_drip | 0*.*001 | 0*.*006 | 0*.*151 | 0*.*880 |
| creation\_delta | 0*.*789 | 0*.*006 | 141*.*316 | 0 |
| last\_login\_delta | 0*.*039 | 0*.*006 | 6*.*401 | 0 |

Table 2: Coefficient estimates: Source

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr(>|t|) |
| (Intercept) | 0 | 0*.*009 | 0 | 1 |
| creation\_sourceORG\_INVITE | -0*.*053 | 0*.*013 | -4*.*185 | 0*.*00003 |
| creation\_sourcePERSONAL\_PROJECTS | -0*.*101 | 0*.*012 | -8*.*662 | 0 |
| creation\_sourceSIGNUP | -0*.*033 | 0*.*012 | -2*.*810 | 0*.*005 |
| creation\_sourceSIGNUP\_GOOGLE\_AUTH | 0*.*001 | 0*.*011 | 0*.*079 | 0*.*937 |

# Creation

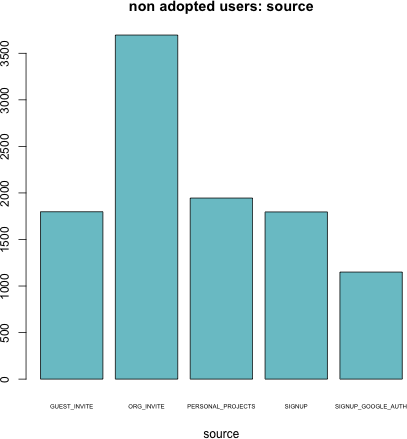
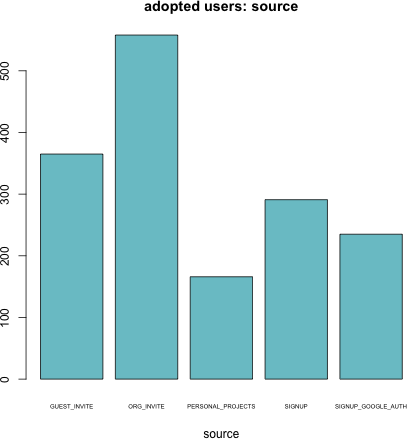
These first figures represent barplots of the relative frequencies of creation\_source, broken up by adopted users versus non-adopted users.

Table 3: Coefficient estimates: Email

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr(>|t|) |
| (Intercept) | 0 | 0*.*009 | 0 | 1 |
| opted\_in\_to\_mailing\_list | 0*.*004 | 0*.*010 | 0*.*390 | 0*.*696 |
| enabled\_for\_marketing\_drip | 0*.*007 | 0*.*010 | 0*.*654 | 0*.*513 |

Table 4: Coefficient estimates: Time Deltas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr(>|t|) |
| (Intercept) | 0*.*142 | 0*.*007 | 19*.*123 | 0 |
| creation\_delta | 0*.*839 | 0*.*008 | 104*.*122 | 0 |
| last\_login\_delta | -0*.*122 | 0*.*008 | -15*.*189 | 0 |



# Time Deltas

The figures below are boxplots of time deltas. Time deltas are defined as follows: creation\_delta = time from account creation to last login, lifespan\_delta = time from signup to when this report was created (12/9/16), and last\_login\_delta: time from last login to when this report was created (12/9/16).

