Relax Challenge

Justin Huang

Loading in the data

Takehome_users and engagement had a strange format and to load them in I had to set encoding to 'iso-8859-1'.

Takehome_users.csv had user information, while takehome_enagement had the user login activity.

Data Cleaning/Wrangling

The goal was to take the engagement csv and transform this into the target variable. With the feature variables in takehome_users.

To create some features, I separated out the email to get email_domain and then also changed the creation_time to date time and created another feature called time_diff_mins which was the time difference between the last session created subtracting creation_time.

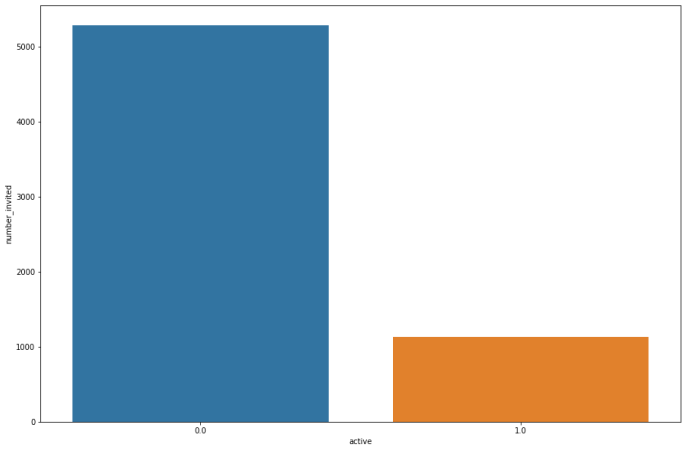
Another feature was to count the number of invites each user sent out by taking the invited_by_user_id column and summing it up and assigning the id to the object_id(later changed to user_id).

Now to get the information if the user was active or not, we had to create a datetime date so we could find if they were active 3 days out the week in the 2-year time span.

The steps were:

- Create dictionary to hold active user
- For loop to see if the user logged in 3 days out of a week in the 2-year time span.
- Create a feature that counts total visits for engagement dataframe
- Then match active users with the total visits count if it's more than 3 then it would count as an active user.
- Then merge it to takehome_users

In the end there were 10344 inactive users and 1656 active users.



Exploratory Data Analysis

- Created bokeh bar plot of number invited and name
- Bokeh bar plot active and email count
- Time series ploty plot of number invited and active over creation time

Stats

	active	number_invited	opted_in_to_mailing_list	enabled_for_marketing_drip	time_diff_mins
active	1	0.044	0.0088	0.0066	0.14
number_invited	0.044	1	0.0064	0.0017	0.0037
opted_in_to_mailing_list	0.0088	0.0064	1	0.48	0.01
enabled_for_marketing_drip	0.0066	0.0017	0.48	1	0.013
time_diff_mins	0.14	0.0037	0.01	0.013	1

Nothing was strongly correlated.

Machine Learning

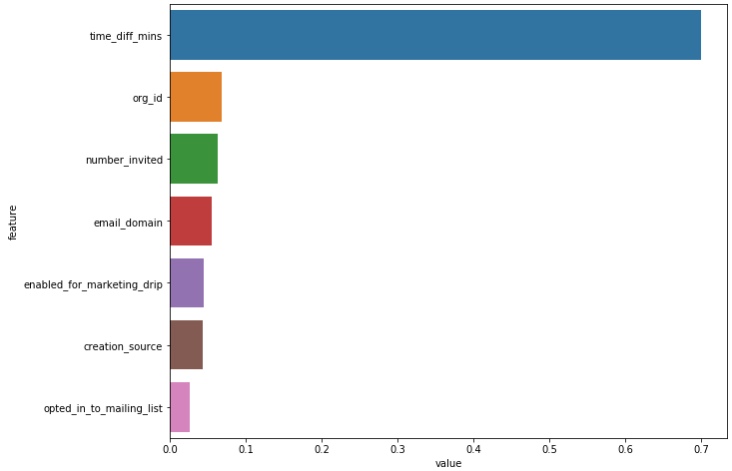
Used the sklearn library and ran 3 algorithms. The features were

- 'creation_source',
- 'opted_in_to_mailing_list',
- 'enabled_for_marketing_drip',
- 'org_id',
- 'email_domain',
- 'time_diff_mins',
- 'number_invited'

Xgboost had a cv score of 0.862, Logreg had cv score of 0.86175, random forest had a cv score of 0.858. Decided to further explore Xgboost for hyperparmeter tuning.

hyperparameters = {
  'n_estimators': [50, 100, 150, 200, 250],
  'max_depth': [5, 7, 11, 15],
  'learning_rate': [0.1,0.3,0.5,0.7,0.9,1],
  'alpha': [5,10,15,20]

The best parameters after running a randomized search cv for 10 iteration at a random_state of 77 is show below.



Time difference, the feature I created by taking the difference between creation_time and last_session was almost at 0.70. I decided to experiment by using only this feature using a quick cv score test.

```
In [ ]:
1  #Lets try to just have time_diff_mins

In [475]:
1  X3 = df.loc[:,['time_diff_mins']]

In [476]:
1  #XGBOOST
2  xg_clf = xgb.XGBClassifier()

In [477]:
1  cross_val_score(xg_clf, X3, y, cv = 5, scoring = 'accuracy').mean()

0.8619999999999999
```

And it looks like this was actually the most important feature and it explains 86 percent of the variability of whether the user is going to be active with Relax on non-active.