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.

A close up of a logo

Description automatically generated

**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**

A screenshot of a cell phone

Description automatically generated

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.

A screenshot of a cell phone

Description automatically generated

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.

A screenshot of a cell phone

Description automatically generated

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.