Relax Inc. makes productivity and project management software that's popular with both individuals and teams. An 'adopted user' of their software is considered to be someone who has logged into the product on three separate days in at least one seven day period, and I was tasked with identifying which features predict future use adoption.

I first cleaned and wrangled the data. Knowing that I would ultimately do an XGBoost Classifier model, my goal was to get all of the data into numeric formats. For the 'engagement' dataframe, I converted the 'time_stamp' column into a data object, and dropped the 'visited' column because all the values were the same. For the 'users' dataframe, I first imputed missing values; for the 'last_session_creation_time' I imputed the NANs with 'creation_time' as the only time they interacted with the software would have been when they created an account. For the missing values in 'invited_by_user_id' I imputed with 0 because there are no 0 user_ids. I then dropped the 'email' and 'name' columns as they are identifiers and converted my time columns to date objects. Next I joined the tables together and then created a column 'adopted user' as a user who has logged in on three separate days in at least one seven day period. I wrote the following function to determine if a user is adopted:

```
def is_adopted(x):
    if len(x) < 3:
        return 0
    x = x.sort_values()
    for i in range(len(x) - 2):
        if (x.iloc[i + 2] - x.iloc[i]).days <= 7:
            return 1
    return 0
```

Finally, I used the 'get_dummies' method to one hot encode creation_source, and converted the values to binary. My final dataframe had 9 integer features and 211094 rows.

I then developed an XGBoost Classifier model with 'is_adopted_user' as the target. The target data were unbalanced, with 198327 adopted users and 12767 not adopted users. For simplicity, I used the scale_pos_weight parameter to address the imbalance, but with more time I would have compared resampling methods. After hyperparameter tuning I had a model with excellent predictive ability for the positive class, which is what we are interested in. See to the right for the classification report.

	precision	recall	f1-score	support
0	0.34	0.82	0.48	2580
1	0.99	0.90	0.94	39639
accuracy			0.89	42219
macro avg	0.67	0.86	0.71	42219
weighted avg	0.95	0.89	0.91	42219

While this is definitely a useful predictive model, I next did SHAP analysis so that I could determine which features predict future use adoption. On the SHAP summary plot I did not see any features which clearly predict adoption, but a creation source of 'personal_projects' and 'signup' contributed negatively to user adoption. On the feature interaction plots I saw that higher values of 'invited_by_user_id' and 'org_invite' contributed to higher SHAP values in some interaction plots, and upon further investigation I found that certain 'invited_by_user_ids' and 'org_ids' had high SHAP values, indicating that certain organizations have higher adoption rates, and invites from certain users lead to higher user adoption rates. See below for the lists of the top 10 inviting users, the top 10 organizations and the SHAP summary plot.

Top Contributing invited_by_user_ids:		Top Contributing org_ids:	
11642	6.621700	151	4.252103
11700	6.475986	329	2.962231
11784	5.789236	400	2.791547
11744	5.330072	412	2.595500
9777	5.295250	396	2.581324
5728	5.256385	307	2.570740
11972	5.246023	346	2.542170
9802	5.202250	330	2.322977
11684	5.137185	394	2.080132
11716	5.129557	299	2.077443