Relax Data Science Challenge – Solution

Defining Adopted User

"Defining an *"adopted user"* as a user who *has logged into the product on three separate days in at least one sevenday period"*

Hence, I created a pivot-table using 'takehome_user_engagement.csv', and used the given definition to write a script and created a binary column that assigns '1' to an adopted user and '0' to a user that is not yet adopted. You can have a look at the pivot table and the script in the Jupyter-Notebook.

Classification

After defining the binary class to every user, I used data from takehome_users.csv to create a set of features to predict whether a user is adopted or not. Some features are used as it and some are created for the sake of better classification. Further feature selection and engineering could be done, but I have kept it simple it simple to save time.

Designed Features

1. A binary class '**last_session_binary**' based on last_session_creation_time whether a last_session_creation_time value is null or not
2. '**Account Age**' based on creation time and last session creation time
3. '**Invitation**' binary-class – a user of invited by another user or not
4. Turned 'name' and 'emailId' into integers – their lengths '**namelen**' and '**emaillen**'

Original Features

creation_source, opted_in_to_mailing_list, enabled_for_marketing_drip, org_id, invited_by_user_id

Prediction – Binary Classification

The dataset is not symmetrical where only **13.8%** of the users being 'Adopted Users'. So, the normal implementation of the Random Forest Classifier does not yield a good result (recall) on the minority class. And in most of the unbalanced classifications, it is most important to correctly classify the minority class.

So, there are additional methods that can be used such as oversampling, undersampling, etc. I have used random oversampling in this case to improve the result. My final result is as follows after the implementation of random oversampling -

```
Classification Report on the Training data -
              precision    recall  f1-score   support

          0       0.98      0.53      0.69      3150
          1       0.22      0.93      0.36       450

avg / total       0.89      0.58      0.65      3600

Accuracy Score = 0.56
Recall Score =  0.870646766169

Confusion Matrix -
 [[3654 3540]
 [ 156 1050]]

Classification Report -
              precision    recall  f1-score   support

          0       0.96      0.51      0.66      7194
          1       0.23      0.87      0.36      1206

avg / total       0.85      0.56      0.62      8400

Area under the ROC curve = 0.739116540824743
```