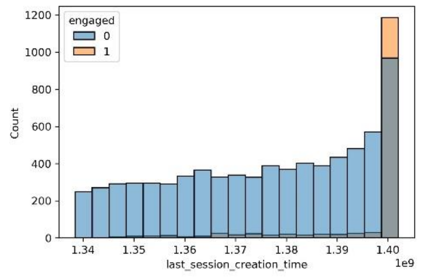
<u>Relax, inc. Challenge</u>

Of the factors provided, we want to figure out which are relevant in determining whether a user adopts the product. A user is deemed adopted if they have logged in three times or more in any given week over the entire period of observation.

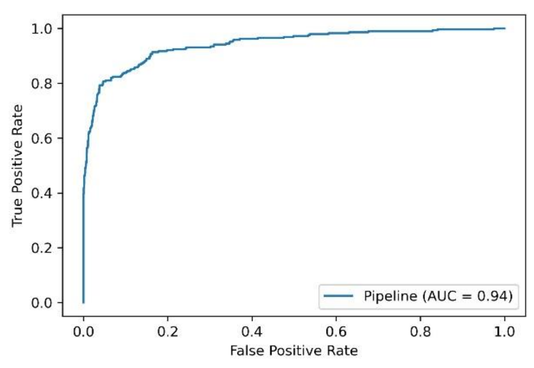
We first downsampled the "takehome_user_engagement.csv" timestamps for logins to a weekly frequency and sum aggregated these for each user. The result was a breakdown of the number of logins each week for each user. We then used the criterion for engagement to see if a user had a week where they used the product more than three times. This resulted in a Boolean series that was converted to integer labels for our two classes: 0 for not engaged. 1 for engaged. This series was then merged (inner join), by user id, into the data for "takehome_users.csv". The result was each user's id, the features, and corresponding labels. **We found that this dataset was highly imbalanced with the majority of the users in the "not engaged" class.** Appropriate methods for resampling need to be used.

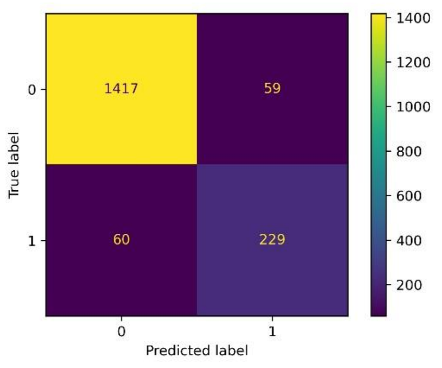
There were a lot of categoricals in the feature set. The cleaning, transforming, and encoding of the various features are contained within the notebook. After doing this, we visualized some of the features conditioned on the class. One of the features that stood out the most was ''last_session_creation_time". This is a stamp of the last time a user logged in. A histogram is seen below:



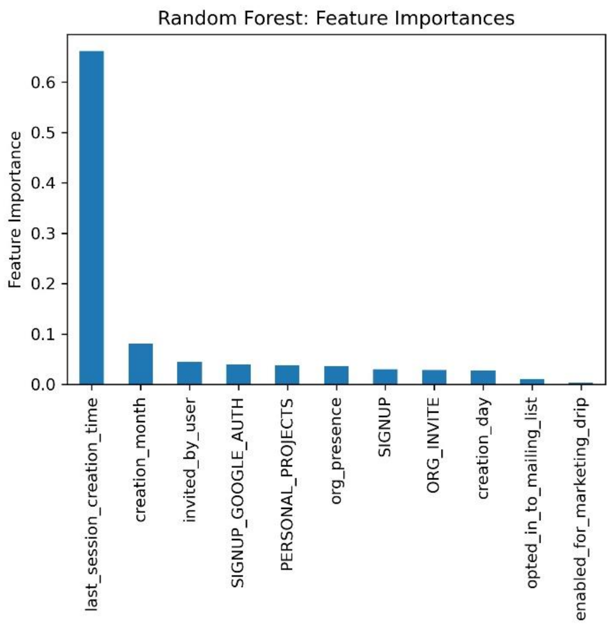
It's clear that for someone who is engaged it is likely that they have logged in very recently. There are, however, a lot of non-engaged people who have logged in recently as well but there is significant weight in this distribution at earlier times as well. We figured that this along with other features would be important for classification.

The only challenge was dealing with the class imbalance. After a 80-20 train-test split , we used SMOTE to "up-sample" the minority class in our train set and put this into a pipeline along with a Random Forest Classifier in a pipeline. After 5-fold CV optimizing for roc-auc score, we found our best model to be a random forest with 80 trees and a max depth of 9. The performance on the test set was actually quite good. Here are some of the reports below:





The ROC looks quite good as does the confusion matrix (even for the minority class). This is good. The precision and recall on the minority class were both ~ 0.8 – which is quite respectable. A look at the feature importance reveal what factors are driving the performance of the model:



It's dominated by the last_session_creation_time. The creation month seems to matter somewhat too. I suppose it's not that surprising…users that are engaged are most likely to have had their last login pretty recently. This is not much of a prediction. In the future, I might just look at the influence of the other factors. In any case, I refrain from any strong recommendations at this time.