Relax Inc. Data Challenge

Data Exploration / Feature Engineering

To check for errors, I looked at each dataset for duplicates, inconsistent labeling, and numerical outliers. Most of the work came in feature engineering.

From the engagement data, I resampled the data into daily vists per user and using the rolling() function to count the number of days logged-in for each 7-day window. Grouping this data down to the max 7-day logins allowed me to flag each user as ‘engaged’ or not. I also used the data to count the number of days between the user’s creation date and second login – hoping this could be a good indicator of future adoption.

For the user data, I broke out a new table of just the referral user ids to flag each user as having referred another user or not. I also calculated an account age. And finally merging all these features into a final DataFrame. Exploring our numerical features, the second login differential had some outliers, so I dropped any records >= 50 days. My logic being users that wait 50 days for another login aren’t representative of our larger user base.

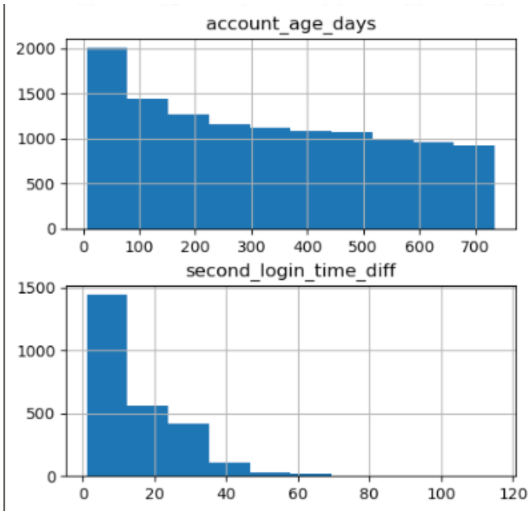
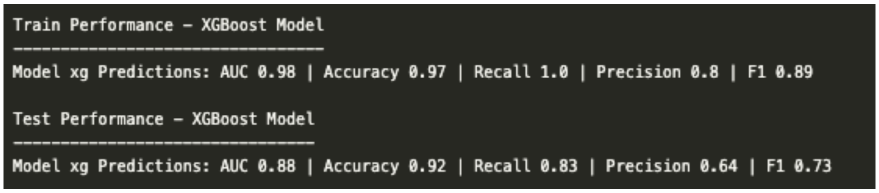


Figure 1: Numerical Features Histograms

Modeling

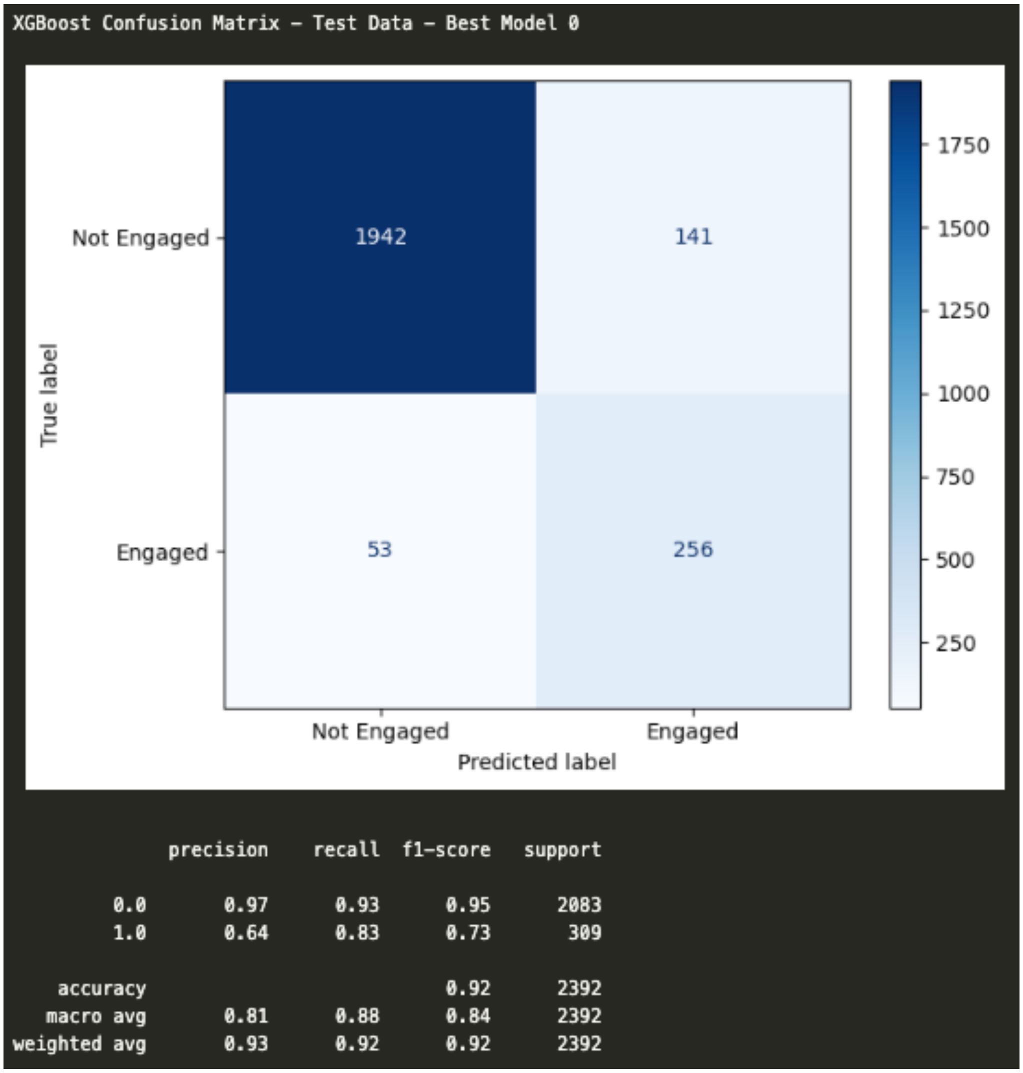
I tested two models: Logistic Regression and XGBoost, using a pre-processing pipeline and GridSearchCV for hyperparameter tuning. For scoring I prioritized accuracy because I want our engagement predictions to not have a high rate of false positives. My Logistic Regression model had high accuracy (93%) but only 69% precision due to the imbalance classes. My XGBoost returned 86% accuracy, adjusting better to the class imbalance.

I chose X for my final model. Here is a summary of the stats and impactful features:



```
Train Performance - XGBoost Model
--------------------------------
Model xg Predictions: AUC 0.98 | Accuracy 0.97 | Recall 1.0 | Precision 0.8 | F1 0.89

Test Performance - XGBoost Model
-------------------------------
Model xg Predictions: AUC 0.88 | Accuracy 0.92 | Recall 0.83 | Precision 0.64 | F1 0.73
```



XGBoost Confusion Matrix - Test Data - Best Model 0

```
              precision    recall  f1-score   support

         0.0       0.97      0.93      0.95      2083
         1.0       0.64      0.83      0.73       309

    accuracy                           0.92      2392
   macro avg       0.81      0.88      0.84      2392
weighted avg       0.93      0.92      0.92      2392
```

Figure 2 & 3: Train vs. Test Summary and Test Result Details

We did lose performance because of some overfitting to training data, but we are predicting 83% of all engaged users correctly. The 64% precision on the positive class means 46% of our predictions are false positives. This we would need to improve on before moving to production.

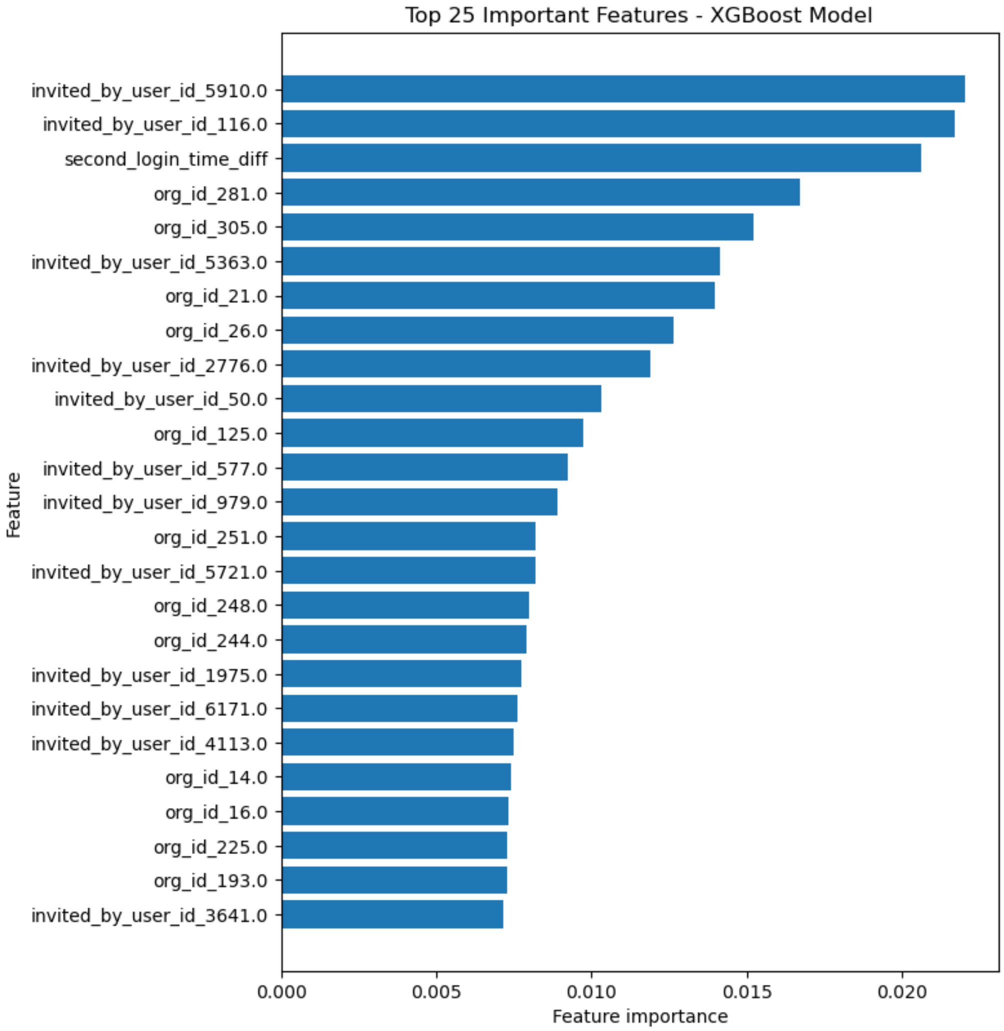


Figure 4: Feature Importance of XGBoost Model

Some of our most important features are the user's organization and referring user, this tells me collaboration within organizations and/or other users is likely driving adoption. The feature we added for second login time is also playing into the model.

Future Improvements

I prioritized a simpler model for this exercise, but I think the primary improvement that could be made is addressing the class imbalance through SMOTE or another technique. We could also try removing the individual referral detail to see if it runs faster with minimal performance loss and doing some further trimming to generalize better to new data.