Waze User Churn Project: Results of random forest & XGBoost models

Deliverables

The development of a machine learning model that predicts user churn.
(Since this is an exercise, all models are predetermined. All Python code can be located at: https://github.com/izsolnay/WAZE_Python)

Results of predictive model

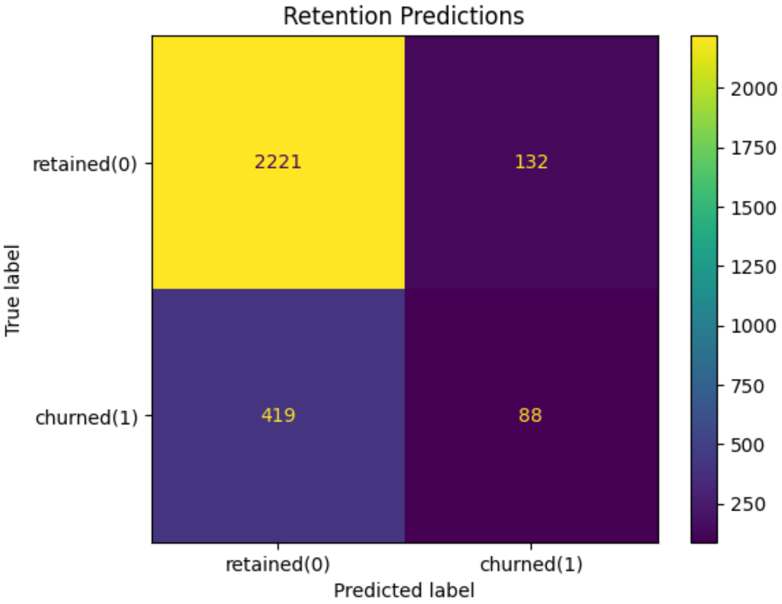
Process

Two tree-based models were built: a random forest model and an XGBoost model. Both models were refitted to the Recall metric to increase the success rate of the logistic regression model in predicting users who would churn. After being fitted to the training data, both models were validated on the set aside validation data set. The XGBoost model achieved both the best Recall and best F1 scores on the validation data set:

Model	Precision	Recall	Accuracy	F1
XGBoost Val	0.412037	**0.175542**	0.809441	**0.246196**
XGBoost cv	0.404893	0.171484	0.808135	0.240830
Random Forest cv	0.454665	0.128763	0.818160	0.200545
Random Forest Val	0.423077	0.108481	0.815734	0.172684
Logistic Regression	0.535088	0.096215	0.824895	0.163102

Results of the winning XGBoost model on the test data set

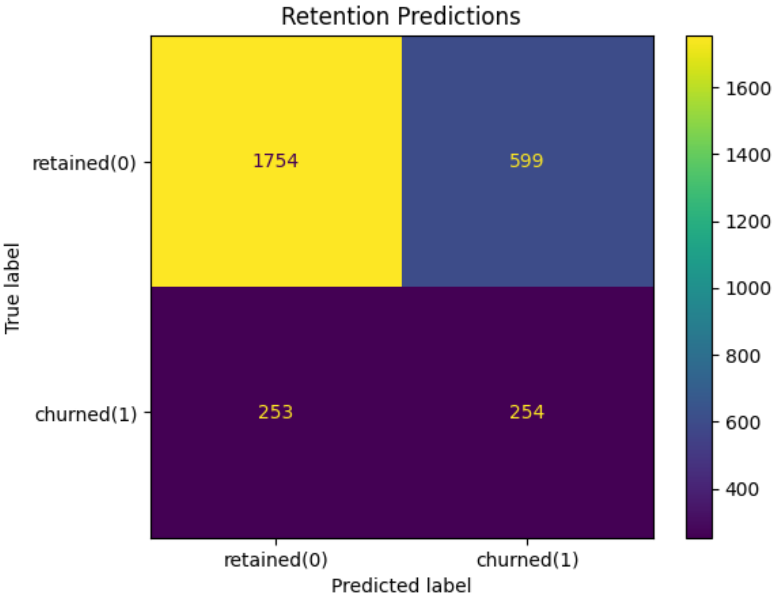
Out of the 2,860 users in the test data set, 507 churned. Of these, the XGBoost model captured 88, thereby achieving an ~18% Recall score. The test data set also included 2353 remaining users. Of these, the model captured 2221 – missing only 132. By refitting the model to Recall, this also drove the F1 score up from the logistic regression model to ~25%. This metric balances the scores the model received in predicting either churned or retained users.



Since the goal of this project is to build a model which can predict whether a user will churn using the available variables, this model's performance is less than stellar. However, these results are somewhat expected in a data set as unbalanced as this one is.

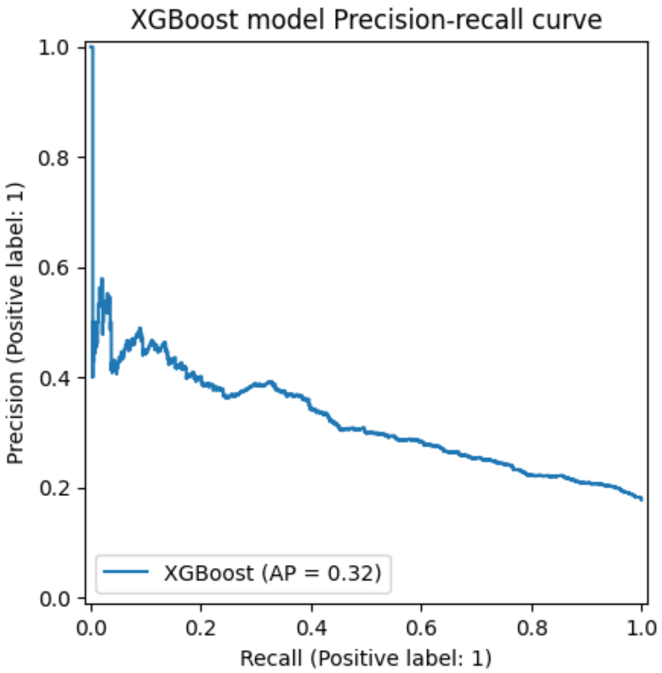
Boosting results further

In order to account for the imbalance in the dataset, the threshold at which the model classified a user as likely to churn was adjusted. Targeting a 50% Recall score required the threshold be set to 0.134 and this provided far more satisfactory results:



With this new threshold setting, out of the 507 churned users in the test data the model now captured 254 (166 more users) and the number of Type II errors decreased from 419 to 253, meaning that with the lower threshold the model predicts significantly better. This altered threshold did mean that the number of users incorrectly predicted to churn rose significantly (Type I error), lowering the Precision score from ~41% to ~30%.

As can be seen in the Precision-Recall plot, it is normal that as recall increases, precision will decrease:

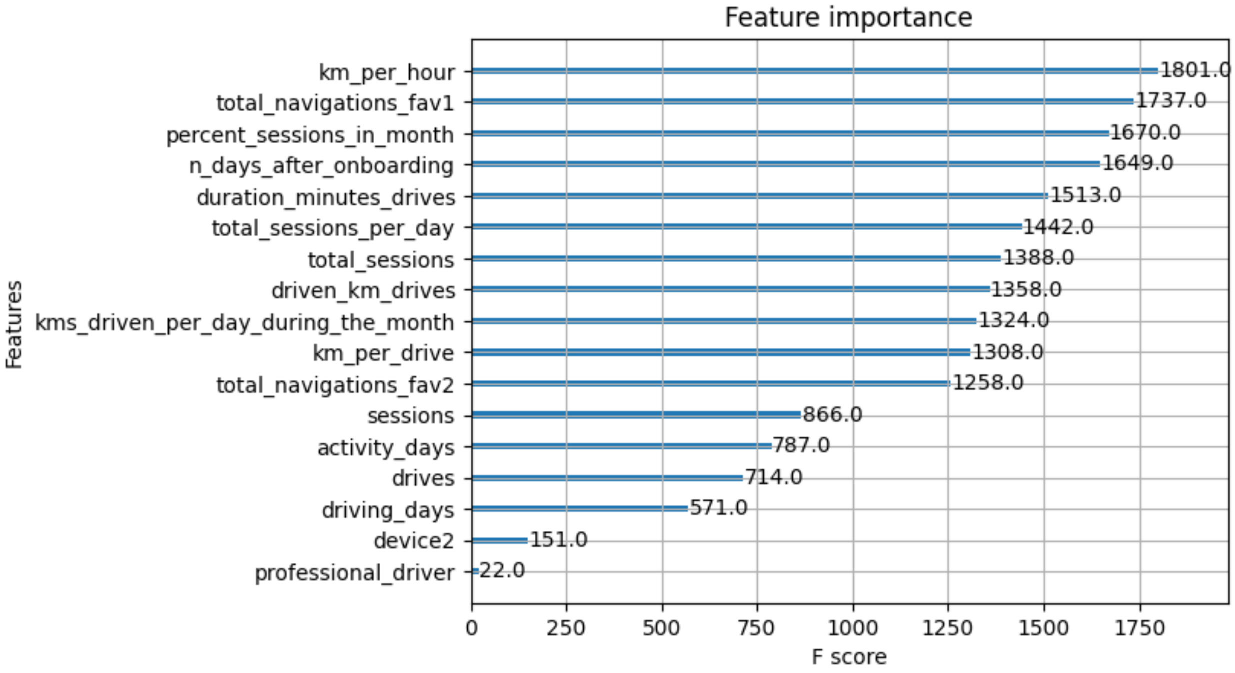


In an effort to identify more customers, the model will become more sensitive to finding churned users and will become prone to incorrectly categorizing people as churned (false-positives/Type-I error) when they remained. This then lowers the Precision score.

Finally, although the Accuracy rating also decreased to 70% (from ~80%), the combined F1 score increased substantially from ~25% to ~37%. This means that on the whole the model was far better at achieving the desired result: locating users who would churn.

Most predictive features

According to the feature importance plot, the rate at which a driver drove (km_per_hour) and whether or not they took advantage of the favorite navigations feature (total_navigations_fav1) were the most predictive features in the data set. Second were percent_sessions_in_month and n_days_after_onboarding. The created feature professional_driver had little to no impact on predictability. Also, the device a user used (device2) was not relevant for the model's calculations.



However, relying on the results of a feature importance rank in an unbalanced data set such as this one is not wise, since it can be biased towards the majority class. This is especially true when it is the minority class that is the target. Feature importance may prioritize features that are more prevalent in the majority class, thereby missing those features crucial for predicting the minority class.

Going Forward

As is, this XGBoost model with the revised threshold could be used to predict users who churn, if the company's retention strategy is cost productive per user. This is because the model will make more Type-I errors. It will classify users as likely to churn when they would not, regardless of intervention.

Alternatively, the data could be scaled so that the minority class (churned users) is more represented or majority class deemphasized. Or, new models could be built using more or different features. For example, the data set does not contain true records of total drives. And, it might be helpful to learn why so many sessions seem to have taken place in the last month.