Overview

The goal of this challenge is to predict if a client will stop running advertising campaigns (churn) based on the provided dataset. The dataset contains 9 predictor variables, including the length of time the client has been a client and the number of calls/clicks the client received as a result of the advertising, along with the target variable, churn. After exploring the data and performing data cleaning and feature engineering, I considered the following classification models using recall as an evaluation metric: logistic regression, decision tree, SVM, random forest, GBM.

Initial Findings

After some exploratory data analysis, there were three notable findings:

1. There is only one feature with any missing values, CPL_wrt_self, and it has about 10% missing values. The missingness does not appear to be systematic as among the clients missing this information, there's a variety of values for the other features.
2. There are two categorical variables in the dataset, client_state and BC, and both have high cardinality. This is a concern as creating dummy variables for these features would result in a sparse dataset with 80+ low variance features.
3. The dataset is imbalanced. 20% of the customers in the dataset churned while the other 80% stayed, and this is a concern because a model that always predicts that a client won't churn can get 80% accuracy.

Data Cleaning and Feature Engineering

First, I dealt with the missing values using nearest neighbors imputation, to replace a given observation's missing CPL_wrt_self with the average CPL_wrt_self for the 5 observations in the dataset closest (i.e. most similar) to that observation, using a nearest neighbors algorithm. I chose this over mean imputation because while mean imputation is unbiased, it adds no information and can introduce bias if the missingness is not random.

Next, for the two categorical variables, I used likelihood encoding, which essentially encodes each class of the categorical feature with the mean target value within that class, given some regularization. For example, with the categorical feature "client_state", if the client's state was "NY", I would replace the label "NY" with the churn rate (average "churn") among all clients in NY. Likelihood encoding allows me to create labels that are directly correlated with the target and avoid using dummy variables to represent categorical features.

Lastly, I considered resampling the data to correct the imbalance, using SMOTE to generate synthetic observations for clients who churned, but decided against it as it is very difficult to make any inference using models trained on resampled data. Since we may want to use the model to identify the factors that cause clients to churn, I chose not to resample here and instead decided to use class weighing (more details in the following section).

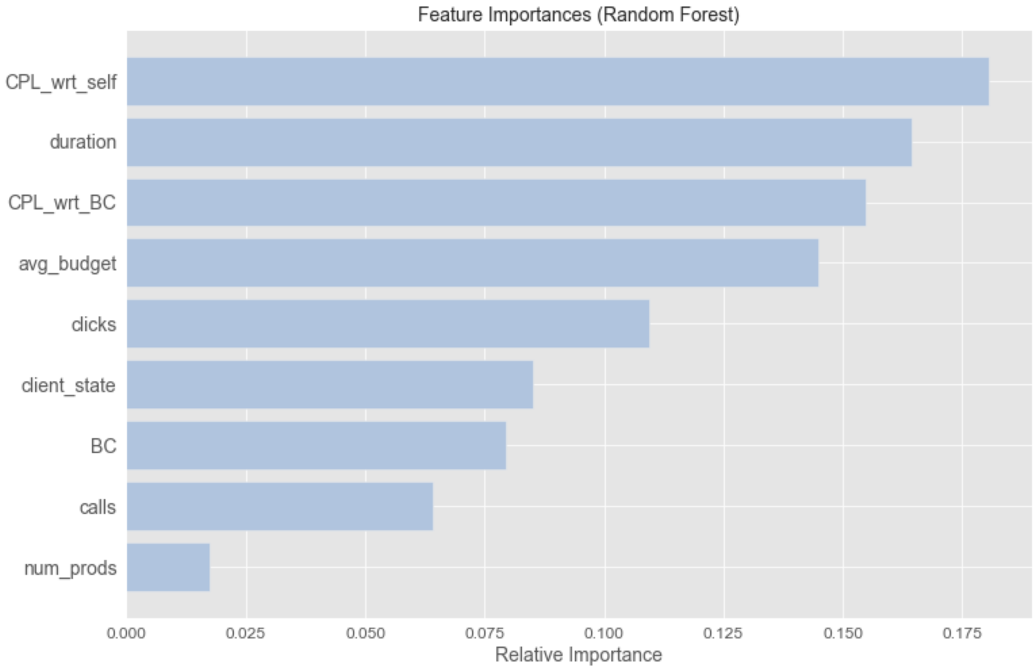
Model Fitting and Evaluation

For this classification problem, I created logistic regression, decision tree, SVM, random forest, and gradient boosting models. I used 80% of the dataset to train each model, using 5-fold cross validation to tune the hyper-parameters as needed, and used the remaining 20% to evaluate and compare model performance, using recall as the evaluation metric.

I chose to use recall because I want the model to be able to correctly identify all the clients who actually churned. I'm willing to misidentify a few clients who actually didn't churn if it means I can correctly classify more clients that actually churned. Optimizing for recall will allow the model to make better predictions for the customers that actually churned, which allows us to better target clients in order to reduce churn rate (example below).

To deal with the data imbalance mentioned earlier, I assigned class weights to the classification models, penalizing misclassifications of the minority class more heavily. In this case, there are 4 times as many retentions as churns, I will penalize the model 4 times as hard for misclassifying a client who actually churned.

After fitting the models, the random forest model performed the best, with a recall of 68.8% on the test set. The model test accuracy was 82.6%.



Lastly, I used the random forest model to get a sense of feature importance. According to the model, the three most important features are CPL_wrt_self, duration, and CPL_wrt_BC. So, if we want to maximize retention, one idea is to give a retention offer to a client that is predicted to churn, offering them a discount that temporarily lowers the client's CPL_wrt_self and increases the duration the business has been a client, in hopes that when the offer expires, the increase in duration will be enough for the customer to stay, even without a retention offer.