Homework 2

Ethan E. Lopez

Introduction

Description of the problem (e.g. what are you predicting? What variables do you have available? How might this model be useful if you are successful). You should end with a sentence or two about what the impact of these models could be.

A streaming service company aims to classify the likelihood of their customers churning or not churning using past profile records, demographics, and other user activities. Of the information in the data, there are 15 independent variables geared towards predicting the churn value for future customers.

There are 9 categorical factors in the present dataset. *Gender* distinguishing the activities between males, females, nonbinaries, and others, *competitorsub, cancelled, downgraded, bundle*, and *kids* are the binary values identifying subscriptions to competitor streaming services, family plans, and other past profile changes. With *plan, topgenre,* and *secondgenre* categorizing current user preferences and content habits, *age, income, monthssubbed, meanhourswatched, numprofiles*, and *longestsession* are the remaining numerical factors determining traits in customer class, maturity, and levels of streaming involvement.

Given the model is proven successful after testing, it will be able to classify a user's susceptibility to churning (discontinuing the streaming service) or not churning given their latest profile characteristics. Allowing the service to identify customers who are on the brink of cancelling, the company may keep user retention through personalized recommendations to motivate this group's engagement. Encouraging features such as related film genres, profile customization, and longer viewing sessions, they'll be able to entertain these customers more to maximize consumer loyalty on the platform.

Methods

Describe your models in detail (as if explaining them to the store's CEO), as well as any pre-processing you had to do to the data.

Prior to building the models, data values had to be investigated for fair rationale and reasoning. Pulling up the head, shape, and info of the data frame, this was significant to confirm data size, types, and descriptive stats as practical and realistic. Counting null and missing values was further necessary to ensure that when dropping, a large portion of the data wouldn't be lost. Columns for *age, income,* and *cancelled* holding 3685 null values, this resulted in 3656 rows being dropped from the original dataset. Out of the 99500 rows total, however, this portion only amounted to 3.67% of lost data, which meant I was able to proceed with model building.

Two models in total were constructed using the 15 variables to predict customer churn. The first model an s-shaped logistic curve and the second model a gradient boosting decision tree, these were generated through one-hot encoding and using a method called train-test-split.

Through one-hot encoding, categorical columns for *gender, plan, topgenre,* and *secondgenre* were altered to include binary values (0s / 1s) for every unique input. For example, considering the *plan* variable, since there were three plans included (Premium, Ad-free, Basic), one-hot converted this column into three new columns, *plan_Premium, plan_Ad-free,* and *plan_Basic*, each with 1s and 0s to evaluate whether customers did or did not have that plan. This process was repeated for the remaining variables mentioned above to result in 38 features total.

Those 38 features plus the *churn* column were then implemented through a validation method called TTS (train-test-split) to split the data into two groups. The first group being 90% of the data, this portion was used for building the model during training with the remaining 10% applied for testing model accuracy by predicting *churn* values. The two models attaining differing interpretations and regression types, comparisons were then charted with various error measurements of binary classification and roc curves.

The first model being a logistic regression, this assumes a sigmoid binary outcome (churn or not) determined by the 38 features. With every one unit increase in a variable, this results in an increase or decrease in the predicted log odds, affecting individual probabilities to churn given that all the other variables don't change. This model works for relationship simplicity, though it can be moderately complex to interpret considering the slope is log odds. For future references, it may be necessary to convert these coefficients to odds using exponentials for explanation.

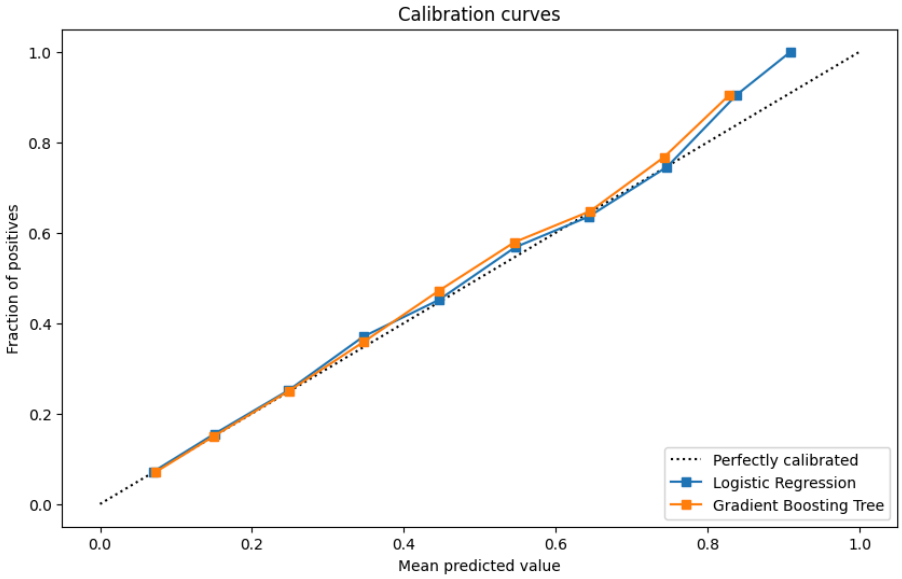
The second model is a gradient boosting decision tree. A normal decision tree being a model that works for binary and multi-class classification problems, it uses a flow chart like structure to establish relationships between variables and determine outcomes at its bottom leaves. Single decision trees often having high variance with predictions that are too optimistic, however, gradient boosting attempts to solve this problem by creating multiple trees and allowing them to build off each other, correcting previous errors to produce a final outcome.

As a whole, this model is more complex and prone to overfitting considering features may be repeated to memorize noise, creating a model that is potentially too large to observe key patterns while forming relationships that are too specific to training data. With each variable in a certain range affecting the outcomes of the others below it, there are no one-sentence remarks for singular factors and how they are to affect unseen data. Increasing a variable by one unit is, therefore, NOT likely to result in churning or not churning *by itself* in a certain magnitude because the tree is designed to operate cohesively with every other feature.

Results

How well did your models perform according to the various metrics, were the models overfit (how can you tell)? What do those performance metrics (Acc, Precision, Recall, ROC AUC, Calibration...) tell you about the model? How much do you trust the results of your models? Which one model would you choose to put "in production"? What are the pros and cons of each model (think about model performance, time/space complexity, interpretability...etc). Is the model well calibrated? Does that matter? How would you suggest the CEO use the classification model you chose? How would you suggest the CEO strategically use the movie suggestions you generated?

Figure 1:



A graph assessing calibration performance between the two models

Both the logistic regression and gradient boosting trees performed similarly across overall metrics without overfitting. Where ROC AUCs and accuracies for training and testing were

found to be around 74%, these measurements showcased an average likelihood to generate correct outcomes in customer churns combined with strong predicting power in well-calibrated curves. These models, on the other hand, were also identified to be weak in precision (avoiding false positives) and especially vulnerable in recall (correctly identifying positive churn cases). Depending on the context of the situation, this could mean several things for the company.

For starters, the company's models having weak precisions (approximately 62% in train / test sets) wouldn't necessarily be a problem. In fact, with the models identifying more customers churning than it's supposed to, this may work to the company's advantage allowing them to explore wider scopes of user activity. Leaving fewer customers unchecked and optimizing experiences for more people to be satisfied with their service, this practice in the long run would prove worthy in inclusive customer interactions and gaining additional subscribers.

It's with recall being at its lowest (27%) is where we see there to be some concerns. Lower measurements in recall amounting to the models avoiding many people who actually churn, this would be damaging to the streaming service as they would be attending to more customers who are already satisfied with their program rather than the ones who are thinking about cancelling. Defeating the overall purpose of the analysis itself to identify those who will churn, both models would thus be seen as insufficient to solve the actual problem.

However, if I were to ultimately choose the better model between the two, it would be the logistic regression. Achieving slightly better testing performance metrics of 0.2% in accuracy and ROC AUC, the regression worked to diminish precision by the same amount while heightening recall 2% more than the gradient boosting decision tree. Its calibration curve aligning closer to the main diagonal (y=x), this further indicates the model being slightly better at predicting actual outcomes overall. As a final note, while having a greater likelihood to predict more correct cases, logistic regression also works for a more interpretable model, understanding how individual variables affect the probabilities of a person churning or not.

While logistic regression is generally better, I advise the company to refer to it for referencing trends and not as a reliable model for classifying actual churn outcomes. Despite the model not showing signs of overfitting in standard predicting power on similar training / testing metrics, its inability to correctly classify a majority of positive cases raises concerns about targeting the wrong customers. The model assuming fixed changes in log odds probability for variables, this simplistic explanation also skews real life interpretations, so it's imperative the CEO discloses factors likely to result in churn rather than stating the coefficient's odds to other stakeholders.

Regardless, despite failing to classify positive churn values, the company may gain a strategic advantage implementing the model for movie suggestions directed to all users. Identifying the

patterns of 10 similar profiles and the 100 films they are attributed to watching, these lists would be useful for tuning user recommendation systems to engage customers with a variety of content they are apt to continue watching. Showing series and films related to customer expectations, the service would be preventing churns for all its consumers by encouraging longer viewing sessions with high desirability and personalized genre / film connections.

Figure 2:

	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision	Train_ROC_AUC	Test_ROC_AUC
Logistic Regression	0.741557	0.737298	0.274513	0.283545	0.604955	0.621517	0.735365	0.740098
Gradient Boosting Tree	0.744050	0.736046	0.265503	0.268715	0.621745	0.623770	0.740046	0.738945

A table displaying model performance metrics between training and testing sets

Discussion/Reflection

A few sentences about what you learned from performing these analyses, and at least one suggestion for what you'd add or do differently if you were to perform this analysis again in the future.

Performing this analysis, I observed the best model for binary churn classification to be a logistic regression for interpretation purposes, though conceptually both models were weak in determining positive churn outcomes. Several techniques observed included pre-processing the data through null values with a 90/10 train-test-split to evaluate model performance. Values of accuracy, precision, recall, ROC AUC, and calibration curves analyzing strengths in the models' predictions, though both were okay fits, none were completely trustworthy due to a large tendency to identify those who churned as those who didn't churn (low recall).

Operating on this dataset again in the future, I would address several weaknesses overlooked during this model building process. Considering an excess amount of features (38), I'll use LASSO and Ridge regularizations to limit the influence of weaker variables and determine the most important factors driving customer churns. Moreover, prior to model validation, I'll use other techniques like stratified sampling to even out proportions between training and testing sets. This way, the metrics would be less biased with similar churn class sizes on both sides.