Homework 2

Spencer Au

Introduction

We are predicting customer churn for a streaming service, utilizing a dataset with various customer-related variables. The aim is to develop predictive models that identify customers that are at a high risk of churning, or quitting the service. These models hold significant value for the streaming service, as they enable the business to successfully identify high risk customers in addition to offering recommendations to increase retention rate. In summary, these predictive models empower the streaming service to make informed decisions, improve customer retention, and deliver tailored experiences, resulting in a positive impact on both customer satisfaction and business outcomes.

Methods

In this analysis, we use two models to predict customer churn for the streaming service. To prepare the data for model training, we undertook a series of preprocessing steps, including handling missing values, normalizing continuous variables using StandardScaler(), and encoding categorical features using OneHotEncoder(), and then z scoring this via make_column_transformer(). We then utilized two models: Logistic Regression and Gradient Boosting Trees. Using a train/test split of 90/10, both models were evaluated using Accuracy, Recall, Precision, F1 score, and ROC AUC to assess their performance on both training and testing data. The goal of these methods is to enable the streaming service to proactively identify high-risk customers and implement optimal retention strategies, ultimately improving both customer satisfaction and business outcomes.

Results

Performance Metrics

Accuracy:

Definition: Accuracy measures the overall correctness of a model's predictions. It calculates the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset.
Use Case: Accuracy is often used as a general measure of model performance when the classes in the dataset are balanced (roughly equal in size). It tells you the proportion of correct predictions.
Performance: Both the Logistic Regression and Gradient Boosting Tree models achieved relatively high accuracy on both the training and test datasets.

Precision:

Definition: Precision measures the accuracy of positive predictions made by the model. It calculates the ratio of true positives to the sum of true positives and false positives.
Use Case: Precision is crucial when the cost of false positives is high. It tells you how well the model performs when it predicts a positive outcome.
Performance: The Gradient Boosting Tree model exhibits higher precision compared to the Logistic Regression model on both the training and test datasets. This suggests that the Gradient Boosting Tree model has a lower rate of false positive predictions.

Recall

Definition: Recall measures the ability of the model to correctly identify positive instances. It calculates the ratio of true positives to the sum of true positives and false negatives.
Use Case: Recall is important when missing positive cases can have severe consequences. It tells you how well the model captures all actual positive instances.
Performance: Both models have similar recall values on the test dataset, indicating that they perform similarly in capturing actual positive cases.

F1 Score:

Definition: The F1 score is the harmonic mean of precision and recall. It balances the trade-off between precision and recall. It is calculated as 2 * (Precision * Recall) / (Precision + Recall).
Use Case: The F1 score is useful when you want a single metric that considers both false

positives and false negatives. It provides a balanced measure of a model's performance.
Performance: Both models seem to perform similarly in terms of their F1 scores on the test dataset.

ROC/AUC

Definition: The Receiver Operating Characteristic (ROC) curve is a graphical representation of a model's performance at different classification thresholds. It plots the True Positive Rate (Recall) against the False Positive Rate (1 - Specificity or Precision) at various threshold values.
Use Case: ROC curves are useful when you want to visualize a model's performance across different decision thresholds. The Area Under the ROC Curve (AUC) summarizes the ROC curve into a single value. A higher AUC indicates better model performance.
Performance: The Logistic Regression model has a slightly higher ROC/AUC value on the test dataset

Logistic	Train	Test	Gradient Tree	Train	Test
Accuracy	0.7414	0.7381	Accuracy	0.7352	0.7376
Precision	0.6063	0.5891	Precision	0.5845	0.6025
Recall	0.281	0.2643	Recall	0.2618	0.2619
F1	0.384	0.3649	F1	0.3616	0.3651
ROC/AUC	0.7366	0.7284	ROC/AUC	0.7152	0.7181

Both models exhibit relatively consistent performance on the training and test datasets, suggesting that they are not heavily overfit. I would express a moderate level of trust in the models' predictions. The accuracy is reasonably high at 0.73-0.74, the precision is moderately high at around 0.6, recall isn't great at 0.26-0.28 as this is the ability to accurately identify true positives, meaning some customers who churn will "slip through the cracks", F1 score seems to be decent at 0.365, though it is weighed down by the recall, and ROC/AUC is solid at around 0.72 for the test cases.

In terms of calibration, it does seem like the Gradient Boosting Tree is better calibrated vs the Logistic Regression model, as there is less deviation from the diagonal line. However, both models do seem to be reasonably well calibrated. Calibration is crucial for identifying high-risk customers who may churn because it ensures that the predicted probabilities accurately reflect the likelihood of churn. This is important because it enables the streaming service to prioritize customers who are at a higher risk of churning, and implement retention strategies accordingly as well as ensuring resources as allocatd appropriately towards those customers.

In terms of model performance, Logistic Regression (LR) is good when the relationships between predictors and outcomes are relatively simple, while not being great at accounting for more complex relationships. On the other hand, Gradient Boosting Trees (GBT) is great

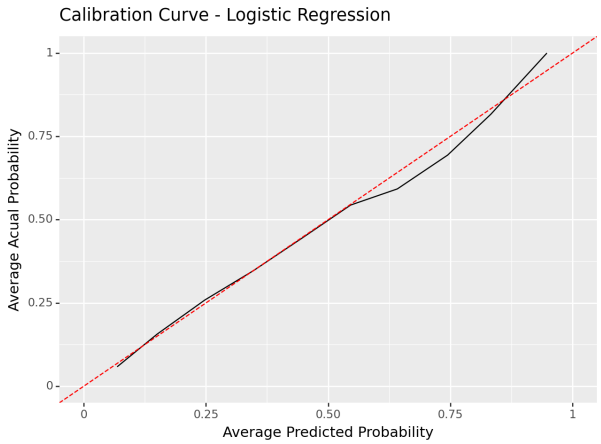

Figure 1: Calibration Graph of Logistic Regression

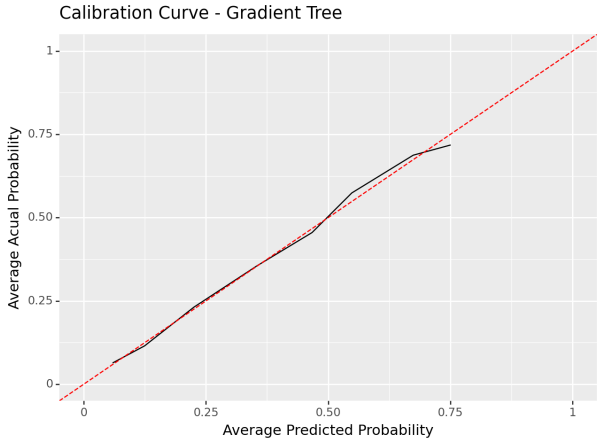

Figure 2: Calibration Graph of Gradient Boosting Tree

for capturing complex relationships between the predictors and outcomes but is more computationally expensive as well as being prone to overfitting. When talking about Time/Space Complexity, LR is efficient and requires less memory, while GBT is more computationally expensive and requires more memory. The tradeoff here is that GBT is able to generally offer better predictions. In terms of interpretability, LR is more interpretable as it is easier to understand the coefficients and how they relate to the outcome, while GBT is less interpretable as it is more difficult to understand how the predictors relate to the outcome.

I would probably choose the Gradient Boosting Tree model to "put into production" as it performs better on precision as well as calibration, and performs equally in terms of accuracy, recall, F1 score, and having a negligible difference in the ROC/AUC score. The CEO can regularly run the Gradient Boosting Trees model on the customer data to identify high-risk customers who are likely to churn. These are the customers with the highest predicted churn probabilities. For the identified high-risk customers, the CEO can implement tailored retention strategies. This could include offering personalized discounts, providing access to exclusive content, making aditional recommendations based on their favorite genres. In terms of leveraging the movie suggestions generated, the company could offer specific and personalized content packages, as well as exclusive access to new releases in order to make the customer feel "special" and valued. This would help to increase customer satisfaction and retention.

Discussion/Reflection

In performing these analyses, I learned a fair about about utilizing different models in order to predict high risk customers who may soon quit the service. In addition, I learned how to analyze the results and performance and choose an appropriate model as well as the respective pros and cons of each model. If I were to perform this analysis again in the future, I would probably try to implement a few more models and compare their performance. I would also try to implement a few more visualizations to better understand the data and the relationships between the predictors and the outcome.