title: “Homework 2” author: “Decker Mecham” format: pdf

Introduction

We are using business churn data to help a streaming service predict wether or not the customer will churn (unsubscribe) We have 15 different variables to help us determine this.

Methods

I ran two different models on the business churn data, a logistic regression and a gradient boosting tree. The train-test set was split 90-10. Four variables were onehot encoded (gender,plan,topgenre,secondgenre). For the gradient boosting tree, we have 15 estimators, a depth of one was chosen, and a learning rate of 0.2 was found for the best outcome.

In the future, we would consider incorporating additional data sources, such as user reviews and social interactions, to further improve model accuracy and user satisfaction. Additionally, continuous monitoring and updating of the recommendation system would be essential to adapt to changing user preferences and content.

Results

Logistic regression has higher recall values in both the training and testing datasets, suggesting that it is better at identifying customers who are likely to churn.

Gradient boosting regression has higher precision values, but it comes at the cost of significantly lower recall. This means that the model is more conservative in labeling customers as churners, resulting in a lower proportion of true positives. Based on the priority of correctly identifying potential churners, Model 1, with higher recall values, is generally more suitable for a customer turnover prediction

The ratio of True positives to false postives is higher with Logistic regression. We see this through the higher AUC-ROC score. In the context of churn prediction, the primary concern is correctly identifying customers who are likely to churn (recall or sensitivity). This is because failing to identify potential churners can result in losing those customers.

The clear pros to logistic regression is the amount of time it takes. It is far faster than gradient boosting trees. The logistic regression is also better calibrated for the data. I myself would not trust much of the data as none of the test scores are that high, I would like to see atleast .75 to .80 for most scores. I also do not know much about the data and where it came from. What preprocessing was done, etc.

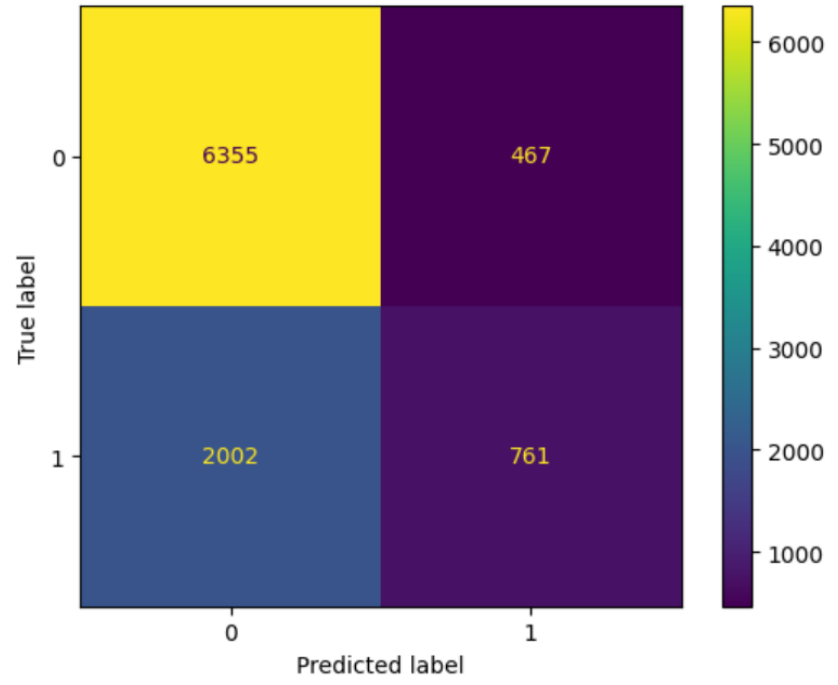

Figure 1: Logistic Regression Confusion Matrix

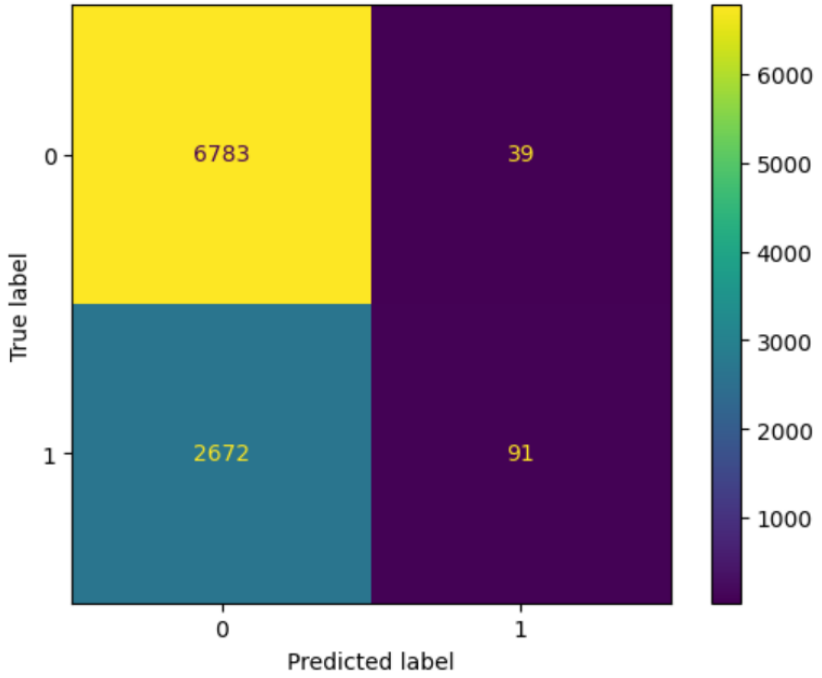

Figure 2: Gradient Boosting Tree Confusion Matrix

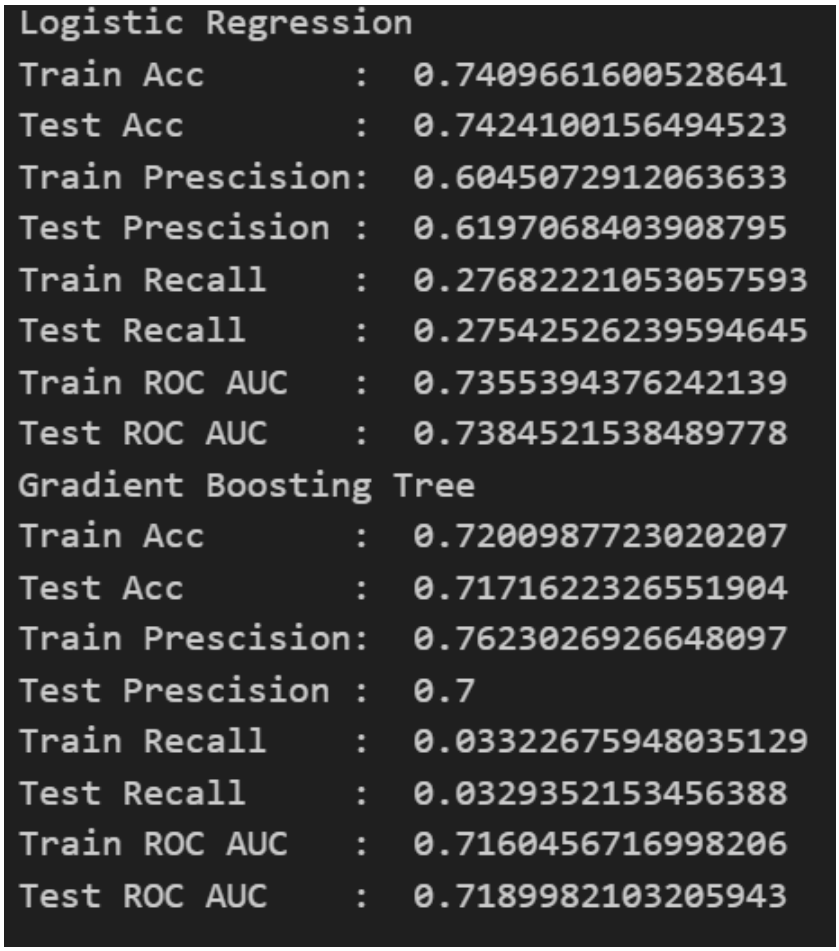

Figure 3: Data Values

Model calibration:

The Gradient Boosting Tree model exhibits overfitting tendencies, leading to a significant gap between training and testing precision and recall. This may indicate issues with the model's calibration and generalization to unseen data. While the Logistic Regression model shows a relatively balanced performance in terms of accuracy, precision, recall, and ROC AUC for both the training and testing sets. No major gaps between training and testing, so logistic is best calibrated for this set of data.

As CEO I would use the churn predictions to offer discounts on their subscriptions. Beyond that, I would give that person fewer ads in the meantime, until their rate of churn updated say the following year.

As a CEO, I would use the list of 200 high risk churn suspects and their similar users to show them targeted movies or shows. If I am able to show them what they like, they may stick around, as by finding the favorite movies of similar customers to them, I can hopefully make good predictions and show them the right movies that will make them stay and pay another months subscription.

Discussion/Reflection

I learned that gradient boosting trees take a lot longer than logistic regressions, especially when you add to the depth of the tree. Also learned that recall was very poor in the Gradient booting tree compared to the logistic regression. For logistic regression I would improve feature engineering by creating new features, transforming existing ones, or through PCA. For Gradient boosting tree I would Optimize the model using grid search or random search for learning rate and tree depth. Overall I found it fun to put new data into a csv, data that we captured from another set.