Troy Kayne
Nov. 24, 2018
Github Repo: [kayne10/gaia-take-home-exercise]

Problem 3: Data Modeling Approach

Imagine that the business has decided that it would be very valuable to understand the likelihood that each of our users will churn within the next 30 days in order to (a) target retention efforts/communications and (b) accurately forecast how many members we are expected to lose (assuming we do not acquire any new members). In this scenario are tasked with generating the predicted likelihood that each member will churn within the next 30 days (however your questions for **this exercise will ask at a high level how you would approach the problem** rather than having your actually solve it using real data)

Questions

1. You do not actually have the data to solve this problem, but based on the sneak peak of data you saw in the “Data Analysis” problem and the data you expect us to have, what predictive data would you look to use to solve this problem?

Based from the sneak peak of data I’ve seen, there are some significant features for predicting if a user will churn or not. This data stores a lot of relevant information for generating predictions such as total number of videos watched/completed, total number of seconds watched, number of days as a member and since last login. In order to view how significant each feature is, I would train and fit a classifier data model, where I could then observe the significant features according to the model.

2. Assuming you had the data from question (1) and were to work on this problem, describe how you would generate predictions.

I would find the most optimal classifier model out of Logistic Regression, a single Decision Tree, Random Forest, and Gradient Boosting. First, I would clean the data as much as possible by filling in any missing values with data imputing. Before fitting the model, I would define my target value (y) as the churn data column and my features (X) as the data columns I described in question 1 along with a few more I did not mention. Then, I would perform a train test split on my X and y columns (80% train, 20% test). Now my data model is ready to be trained. After fitting my model, I can generate prediction probabilities of the likelihood a user will churn or not within the next 30 days. Also, I would use KFold cross validation to further optimize my model.

3. If you were to work this problem, how would you evaluate/measure the quality of your predictions? For example, how would you determine if adding new data into your model improved its predictions or not?

In order to evaluate the quality of my predictions, I would consider metrics such as accuracy, precision, and recall when using my model with unseen data. Since I am using a classifier model, I can measure these metrics with a confusion matrix and ROC curve. If the area under the ROC curve is between .7 and .9, the model would be considered pretty good. If I were to add more features, I would want to see how significant each feature would be. Conveniently enough, these classifier models have an attribute to check the importance of each feature. Then I could evaluate the model with the same metrics and observe if the model improved its predictions or not.