Alec MacMillen
CAPP30254 – Machine Learning
Assignment #3 – Improving the Pipeline
Due 5/30/2019

**Policy Problem**

Public education suffers from system under-investment in the United States. Many public school teachers do not receive sufficient funding for classroom materials and resources from their school districts, so they are forced to either foot the bill themselves or turn to public fundraising methods like DonorsChoose. DonorsChoose is an online platform on which teachers can post funding requests for "projects," which can range in type from classroom materials to less tangible learning opportunities like trips. Donors can then view information about these projects (including location, teacher, type of resource and use, number of students reached, etc.) and donate towards the stated funding goal.

The objective of this project is to build and validate machine learning models that will predict which projects posted on DonorsChoose will *not* reach their full funding goals within 60 days of posting. The ultimate aim of the project is to identify 5% of posted projects that are predicted not to reach their funding goals. Possible interventions for these projects include more aggressive advertising on the DonorsChoose website, matched donation incentives, or requests for more teacher input (pictures, descriptions, etc.) to make their requests more attention-grabbing.

**Data Exploration and Pre-processing**

After loading and summarizing the projects_2012_2013.csv data, certain variables present themselves as good candidates for features. Namely, the categorical variables *school_state, school_metro, school_charter, school_magnet,* primary and secondary *focus_subject* and *focus_area, resource_type, poverty_level, grade_level,* and *eligible_double_your_match* contain important demographic information about the school and project that could aid in classification. Similarly, the *students_reached* and *total_price_including_optional_support* continuous variables can be standardized.

A potential future extension of this project is to merge Census demographic data with geographic information present in the projects dataset and use socioeconomic information about schools' location as feature variables.

Another important step is creating the binary target variable, *not_funded_within_60_days*, which takes a value of 1 if the project does not reach full funding in 60 days (indicating a need to intervene) and a value of 0 otherwise.

The dataset covers projects posted from 1/1/2012 to 12/31/2013, so there are three six-month periods used to *test* the trained models: 7/1/2012-12/31/2012, 1/1/2013-

6/30/2013, and 7/1/2013-12/31/2013. The data from the beginning of the dataset to 60 days before each of these respective test periods was used to train the models.

**Model Training and Evaluation**

I trained a variety of different models (K-nearest neighbors, logistic regression, decision trees, support vector machines, random forests, bagging, boosting, and naïve Bayes) with a variety of different model specifications to determine which were best suited to this task. The main metric of interest in my opinion is precision. Given that the confines of the problem dictate that we only have the resources to intervene with 5% of projects, we're interested in making sure that none of the projects we choose to intervene with are "false positives" that don't actually need any intervention. Maximizing precision will minimize the false positives we erroneously intervene with.

Different models appear to perform best at the thresholds 0.3 and 0.5. The threshold 0.3 maximizes the AUC measure, but the threshold 0.5 maximizes precision. If we're concerned primarily about avoiding intervening with false-positive projects, a decision-tree classifier using the Gini (impurity) criteria, a random splitter, and a maximum depth of 5 maximizes our precision measure. This is the model I would recommend using.

**BEST PERFORMING MODELS AT 0.05 THRESHOLD**

| **Model & specifications** | **Train/test period** | **Precision** |
|---|---|---|
| Random forest<br>*(n_estimators=50, max_depth=3, criterion=entropy)* | *Period 1*<br>Train: January 1, 2012 – April 30, 2012<br>Test: July 1, 2012 – December 31, 2012 | 0.42 |
| Logistic regression<br>*(penalty=L1, C=0.5)* | *Period 2*<br>Train: January 1, 2012 – October 31, 2012<br>Test: January 1, 2013 – June 30, 2013 | 0.59 |
| Random forest<br>*(n_estimators=100, max_depth=2, criterion=gini)* | *Period 3*<br>Train: January 1, 2012 – April 30, 2013<br>Test: July 1, 2013 – December 31, 2013 | 0.48 |

Overall, the best-performing model was the logistic regression model trained in training period 2 (1/1/12-10/31/12) and tested in testing period 2 (1/1/13-6/30/13). Models trained and tested in period 2 overall performed much better in terms of precision than models trained in either period 1 or period 3. Generally speaking, random forest and logistic regression models tended to perform better (result in higher precision) than most other model types, although bagging and gradient boosting also performed fairly well, too.

Ultimately, the random forest and logistic regression models are preferred for identifying a top-5% highest priority projects with which to intervene.