**Assignment 5 Writeup**
**by Lilian Huang**

DonorsChoose.org is an online charity that allows K-12 schoolteachers to propose projects to benefit their students' education, and to request logistical support for these projects through the donation of materials; donors can contribute any amount to the project of their choice. The purpose of this analysis is to predict whether a project posted on DonorsChoose will not be fully funded within 60 days of posting. If a project is identified as being at risk of not being fully funded in a timely fashion, intervention can be undertaken to support it; this has a direct bearing on whether students can get the educational resources they need.

**The Classifiers**

We examined the performance of 8 different types of classifiers: simple classifiers - logistic regressions, k-nearest neighbors, decision trees, and Support Vector Machines - and ensemble methods - random forests, boosted decision trees (adaptive and gradient boosting), and bagged logistic regressions.

All of these are supervised learning classifiers, which learn from training data with the desired output labels, to then make predictions on unseen test data. The difference is that ensemble methods construct multiple models, rather than running only a single model, and then combine the predictions from these models. Bagging (bootstrap aggregating) involves randomly sampling the original dataset with replacement, running a model on each random sample, and averaging the predicted probabilities. A random forest is produced by bagging decision trees. Boosting involves training consecutive models by increasing the weight of misclassified observations, to help ensure that subsequent models focus more on correctly classifying those observations. Adaptive boosting and gradient boosting are two variations of the boosting algorithm.

We also tried out different parameters for these classifiers. For example, in ensemble methods, we tried different values for the n_estimators parameter, which controls the number of models included in an ensemble; for the k-nearest neighbor classifier, we tried different values for n_neighbors, namely the number of neighbors that are taken into consideration when classifying an observation.

**The Metrics**

Besides testing different types of classifiers, we also looked at their performance across a variety of metrics. We looked at precision, which measures the proportion of actual positives among the predicted positives; recall, which measures the proportion of actual positives that the model manages to correctly classify as positive; the F1 score, which is the (harmonic) average of both precision and recall; and the AUC-ROC (the area under the receiver operating characteristic curve). The AUC-ROC indicates whether, on average, a classifier will assign a higher probability to a randomly selected positive observation than to a randomly selected negative observation. An AUC-ROC score of 0.5 indicates that the model's performance is roughly equivalent to randomly labeling observations, while a score of 1.0 indicates that the model's performance is perfect. It should be noted that precision and recall have to be calculated for each specific probability threshold, whereas the AUC-ROC score summarizes how well a model performs across different probability thresholds; as such, we generally rely on the AUC-ROC score here as our primary metric.

Looking at the different metrics, we find that the various classifiers all perform extremely similarly. In terms of precision and recall at different thresholds, the different types of classifiers yield very similar

results, with only simple logistic regression having notably lower values for precision and recall than the other classifier types.

It is also noteworthy that all the classifiers do substantially better in terms of precision than recall. Up to the "20% of the population" threshold, average precision is close to 1.0, but average recall at that threshold is around 0.74. This indicates that our classifiers are perhaps classifying too many observations as having negative outcomes - thus avoiding false positives, and maintaining precision, but sacrificing recall, as the classifier fails to pick up on many actual positives. Our baseline models show that, in each test dataset, there are 25%-30% of observations with positive outcomes, and our recall score indicates that many of these are being classified as negative outcomes instead.

We therefore look at the AUC-ROC score as being more informative, as aforementioned. Here, we see more distinctions between the different classifiers' performances - the simple and boosted decision trees perform very similarly, but otherwise, there is greater differentiation between the classifiers. However, the AUC-ROC score hovers around or slightly above 0.5 for all models (and is just above 0.5 on average), indicating that all our models are quite weak and do not provide substantially more information than randomly assigning observations to classes. It is also hard to identify a single type of classifier that performs better than the others; the random forest classifier performs slightly better than the others, with an average AUC-ROC score of 0.499, when trained with the smallest and largest sets of training data. However, this advantage is not consistent; when the classifiers are trained with the second-largest training set, the support vector machine classifier performs best, with an average AUC-ROC score of 0.509.

**Change Over Time**

The random forest classifier only emerges as having the highest average AUC-ROC score as the timeframe of the training data increases; when trained with the second training set, which is 10 months long, the random forest classifier actually has a lower average AUC-ROC score than the others.

It should be noted that the AUC-ROC score does not increase consistently over time (i.e. as the starting date of the testing data moves further back and the size of the training data increases). For some classifiers, such as simple logistic regression and support vector machines, the AUC-ROC score increased from the first to second timeframe, and then decreased from the second to third timeframe. In other words, performance does not necessarily improve as the amount of data in the training set increases. This may have to do with the differing baseline in each timeframe; the baseline accuracy (as calculated by the number of positive outcomes in the dataset - this is the accuracy we would get if the classifier simply predicted every observation as positive) increases from the first to second timeframe, and then decreases from the second to third timeframe.

**Best Performance**

The single model with the highest AUC-ROC score (0.509) is a logistic regression classifier trained on data from 1/1/2012 to 10/31/2012. The logistic regression classifier fits a linear model to training data, to calculate the probability of a data point's membership in a certain outcome class, based on a linear combination of features.

The exact parameters of this highest-performing model are: C=0.001, penalty='l1'. An l1 penalty means that the model uses lasso regularization - in order to avoid the model overfitting on the training data, a

penalty is imposed for complexity (having more coefficients); an l1 penalty means that the square of the model's less important coefficients is driven down to exactly zero, thus removing some features completely. The value C is the inverse of regularization strength; a smaller value of C means that regularization is stronger, and the model is more constrained, which leads to a sparser solution (with fewer features) when an l1 penalty is used. 0.001 is the second-smallest value of C we tested as a parameter, meaning this is the second most strongly regularized version of the model.

**Final Recommendation**

In practice, precision as a metric translates to how efficiently resources are being allocated (i.e. to what extent do the people/projects that are receiving resources actually need those resources), while recall as a metric translates to the effective of coverage (i.e. to what extent are we overlooking people/projects that actually need to receive resources). There is a general notion of a tradeoff between these two concepts. However, since the person working on this model knows that they can intervene with 5% of posted projects, that means they have a threshold of 5%, and once the threshold is fixed, there is no tradeoff between optimizing for precision and for recall at that threshold - a model that has higher precision will, by definition, have higher recall at the same threshold.

As such, the question is what the ranking of models is at the 5% threshold (where we can serve 5% of the population). We therefore look at the F1 score at that 5% threshold (which conveniently takes into account both precision and recall, as previously mentioned), and find that the best-performing classifier is a random forest classifier trained on data from 1/1/2012 to 4/30/2012. The random forest model is, as previously mentioned, produced by bagging decision trees, i.e. randomly sampling the training dataset, fitting a decision tree classifier on each sample, and averaging the predictions.

The exact parameters of this highest-performing model are: max_depth=5, max_features='sqrt', min_samples_split=2, n_estimators=10, n_jobs=-1. This means that there are 10 decision trees in the random forest; that the tree can have a maximum depth of 5 levels (the deeper a tree is, the more splits it has); that the minimum number of samples needed to split an internal node is 2; that the maximum number of features to be considered when looking for the best split is the square root of the total number of features; and that all processors are used in fitting and predicting the model.

However, despite this result, the final recommendation would be to work on these models for longer before picking a single model to deploy. Deployment at this stage would be premature, given the aforementioned caveats - regarding the overall weak performance of all the classifiers, their similar performance on multiple metrics, the better performance on precision than recall, and the variation in which classifier performs best over different timeframes of training data.

Therefore, rather than focusing on the specific choice of classifier or fine-tuning classifier parameters, more judicious selection of features might be more helpful here - these classifiers currently examine all available features, and narrowing or refining the range of features included in the models might help to improve the models' performance. Alternatively, refining the temporal split of the datasets might help; currently, the testing datasets consist of 4 months of observations, and making the testing window narrower (e.g. 2 months long) would give us more training data prior to that point, which might also help to improve model performance or at least identify consistent temporal patterns in model performance, thus allowing for a more reasoned choice of model to deploy.