Building a Machine Learning Pipeline to Predict Educational Crowdfunding Projects at High-Risk for being Unfunded

Irene Li mengzeli | **Helen Ren** xutongr | **Zehao Wang** zehaow | **Brian Rhindress** brhindre

1 Executive Summary

This memo summarizes the methods and findings of the Donors Choose team throughout the Machine Learning for Public Policy Lab (MLPP) course, instructed by Rayid Ghani and Kit Rodolfa through CMU's Heinz College and SCS Machine Learning Department.

In this project, we developed an extensible, modular Machine Learning pipeline to predict educational crowdfunding project postings that are likely to be unfunded on the site Donors Choose. Donors Choose is a crowdfunding platform that allows teachers to solicit donations for classroom projects. The core datasets originated from a Kaggle competition with a similarly framed goal to help the site efficiently allocate and automate its project-screening resources. Our formulation of a binary classification problem, selecting the top 5% predictions of `unfunded_projects`, was decided upon using the Data Science for Social Good scoping guide[1] to meet the operating needs of Donors Choose.

The five models selected for having the best precision on the 'top 5%' of predictions are all Random Forests. The best model correctly predicts 59% of all predicted `unfunded_projects` over 8 train and test sets spanning 2012-2013. This is an average 26% point improvement over random selection and a 16% point improvement over an intelligent baseline. Top features of importance include the total price of the project posting (mutable by the teacher) and the ratio of past projects of a particular type/subject that were not funded (immutable historical trend). We find that compared to the baseline, the best performing model has a lower False Negative Rate (FNR) ratio of protected to reference groups, defined as high and moderate poverty students, respectively.

As a result of our findings, we recommend that Donors Choose use our estimator to identify projects for special resource allocation depending on the nature of its feature attributes. For projects selected that include feature attributes of importance that are immutable by teachers (e.g. rural schools), Donors Choose should provide advertising and match funding resources to aid the project. For feature attributes of importance that are mutable by teachers (e.g. projects with total price $\gg$ \$700), the organization should automatically screen the project and possibly devote staff hours to helping the teacher improve the project posting.

2 Background and Problem Formulation

2.1 Donors Choose Background

The US K-12 public education system exhibits funding rates that vary by regional and population demographics. Controlled studies have shown schools with the following features spend less per-pupil than their counterparts: high-poverty school districts (\$1,000),[2] non-white school districts (\$2,000),[3] and rural school districts (\$5,000).[4] To address this problem, Donors Choose was created as an educational crowdfunding platform to provide direct classroom funding to teachers. Donors Choose

crowdfunding bypasses the bureaucratic system of state and local education financing. According to Donors Choose data, 84% of public schools in America have posted a project. Over 4M citizens and corporations or foundations have donated, yielding over $900M in classroom funding for 39M students.[5]

Donors Choose reports that 64% of projects posted to the site are funded, while 36% of projects are unfunded and must be taken off the site.[5] Unfunded projects must be removed from the site after four months. Resources Donors Choose may employ to help fund projects include *staff time* for screening projects, *advertising* including targeted emails and online product placement, and *match funding* from corporate partners to provide incentives to individual donors for contributing to particular projects. Note that donations are typically small and few, with 98% of donations being less than $200 and 97% of projects receiving fewer than 20 donations.

2.2 Proposed Intervention

To guide employment of resources, Donors Choose wishes to develop a prediction tool to identify projects that are unlikely to be funded each month. We formulate this problem as a binary classification task, defining `unfunded_projects` as those that are not fully funded by the 120^{th} day after project posting. These projects are labeled '1', while funded projects are labeled '0.' The classifier generates predictions at the beginning of each month as probabilities that each project will go unfunded. From this ranked list of probabilities, the organization can then select the 'top k%' of project predictions, or the k% projects of highest-risk, on which to intervene.

2.3 Intervention Impacts

Prior to modeling the classifier, it is worth considering its potential impacts and trade-offs. This is best done through an example. Suppose in one month there are 20,000 projects posted to the Donors Choose platform, with 6,000 (30%) destined to be `unfunded_projects`. The classifier predicts a probability for all 20,000 projects. Higher-risk projects, according to the classifier, receive higher probabilities. Suppose we select the top 5% of projects on which to intervene, or 1,000 of the 20,000 projects posted. Now suppose that of these 1,000 highest-ranked projects selected, 700 truly will not be funded (true positives), and 300 truly will be funded (false positives). The other 2,000 projects are not classified as `unfunded_projects`, but truly will not be funded (false negatives). The remaining 16,700 projects not selected by the classifier will be funded (true negatives).* Let us now consider this example through the lenses of Effectiveness, Efficiency, and Equity.

Effectiveness can be defined as the "achievement of stated objectives [*but for*] the intervention."[6] For Donors Choose, effectiveness can be captured by the recall metric,

$$Recall = \frac{\Sigma \texttt{Correctly Classified unfunded\_projects}}{\Sigma \texttt{All unfunded\_projects}} \tag{1}$$

In our example, the recall is $\frac{700}{3{,}000}$ or 0.23. That is, 23% of all `unfunded_projects` would be predicted by this classifier.

Efficiency can be defined as "outcomes achieved at least cost."[6] Donors Choose wishes to minimally employ its resources to maximally fund projects. Without knowing the organization's funding spent per project funded and unfunded, we must use a proxy metric for measuring efficiency. Resources are wasted when they are used on projects that are falsely predicted to be unfunded. This is captured by the precision (a.k.a. positive predictive value) of our classifier,

$$Precision = \frac{\Sigma \texttt{Correctly Classified unfunded\_projects}}{\Sigma \texttt{All unfunded\_project predictions}} \tag{2}$$

In our example, the precision is $\frac{700}{1{,}000}$ or 0.7. Of all the classifier's predictions, 70% would be correct.

*Since 30% of all projects typically go unfunded, we assume that the probability of all such projects is above some arbitrary threshold, deeming them worthy of intervention (i.e. > 0.5 or another defined cutoff). These assumptions should be affirmed, in reality.

Equity is more elusive to define, but can be considered as a "distribution of material costs and benefits" with a dimension of equality of opportunity for protected and unprotected groups.[6] AI and ML definitions of fairness relevant to predicted and actual outcomes start by defining protected attributes for non-discrimination.[7] In the case of Donors Choose, given the educational disparities outlined in 2.1 we could consider protected groups based on poverty level, region (rural, urban, or metro), or proportion of racial minorities in each school. We limit our discussion to poverty for simplicity. The first standard of equity is the False Positive Error Rate Balance (a.k.a. predictive equality),[7] where False Discovery Rates (FDR) are equal for both protected and unprotected groups,

$$\begin{aligned} P(\texttt{unfunded\_project} = 1 \mid Y = 0, poverty = high) = \\ P(\texttt{unfunded\_project} = 1 \mid Y = 0, poverty = moderate) \end{aligned} \tag{3}$$

We must expand the example to demonstrate this standard. Let us further assume in the example above that 50% of `funded` (8,500) and `unfunded_projects` (1,500) are from high-poverty schools, while the other 50% from both categories are from moderate-poverty schools. Now suppose that of the 700 true positive `unfunded_projects`, 200 are high-poverty and 500 are moderate-poverty projects. Of the 300 false positive `unfunded_projects` 100 are high-poverty and 200 are moderate-poverty projects. Using these figures, the False Discovery Rate for high-poverty projects is $\frac{100}{1000}$ = 0.1. The False Discovery Rate for moderate-poverty projects is $\frac{200}{1000}$ = 0.2. The FDR ratio of high-poverty (protected group) to moderate-poverty (reference group) is 0.5. Thus, moderate-poverty projects are more likely to be falsely discovered.

Another equity standard is False Negative Error Rate Balance (a.k.a. equal opportunity),[7] which occurs when the False Negative Rate is equal for both protected and unprotected groups,

$$\begin{aligned} P(\texttt{unfunded\_project} = 0 \mid Y = 1, poverty = high) = \\ P(\texttt{unfunded\_project} = 0 \mid Y = 1, poverty = moderate) \end{aligned} \tag{4}$$

Referring to our example, False Negative Rate for high-poverty projects is $\frac{1{,}500-200}{1{,}500}$ = 0.87. Meanwhile, the False Negative Rate for moderate-poverty projects is $\frac{1{,}500-500}{1{,}500}$ = 0.67. The FNR ratio of high-poverty to moderate-poverty groups is 1.30. Thus, high-poverty projects are more likely to be false negatives than moderate-poverty projects.

Taking this example all together we have shown that this classifier has discovered 23% of all true `unfunded_projects` (recall), with 70% correct predictions (precision), an FDR Ratio 0.5, and an FNR ratio of 1.30. Changing the models and 'top k%' cut-off threshold will inevitably cause a trade-off amongst these metrics. These trade-offs will be considered as applicable in future sections.

3 Related Works

3.1 ML & Donors Choose

The datasets for this project originated from a Kaggle competition, the 2014 KDD cup.[8] There have been at least three Donors Choose challenges on the site. The original asked participants to identify 'exciting projects' that are likely to be funded, such that the organization can provide those projects resources early. Other iterations of the competition included the inverse task that we attempt of predicting projects that are unlikely to be funded.[9] Key models trained by winning teams include a Capsule Network Model,[10] XGBoost and LightGBM,[11], a Neural Network with Multichannel Input, and Enesmble methods with Logistic Regression.[12] Features engineered for these models included TF-IDF transformed text features (see appendix section on NLP), and various forms of encoding caegorical features from all given tables. Lastly, there is another competition with the goal of matching donors and projects for targeted funding solicitation emails.[13]

3.2 Literature Review

In the literature, several teams have predicted success in crowdfunding platforms, primarily Kickstarter. Yu et al. (2018) build on a body of Neural Network prediction with a Multi-Layer Perceptron

(MLP) that uses Binary Cross Entropy as a loss function, achieving 93% accuracy as an evaluation metric.[14] While we originally began testing with Neural Networks, both time constraints and a bug in our pipeline resulting in decreased performance led us to discard these models in our final test batch. T. Tranh et al. (2016) take a broader approach, identifying characteristics of funded projects and even predicting expected donations with Naïve Bayes, Random Forest, and AdaBoostM1 models.[15] T. Tranh et al. utilize given project features and engineer user features, temporal features, and twitter features, which are shown to modestly improve accuracy and AUC evaluation metrics. Interestingly, they find notable investment flows at project inception and completion, which can be used to model success.[15] While we similarly engineer historical trends, our framing of prediction at project posting prevents use of real-time donation data. Lastly, another line of research exists in which ML is being used to answer social science questions. Using GoFundMe data, Sudhir (2019), seeks to derive psychological and behavioral economic insights from donation data joined to image analysis and Natural Language Processing (NLP).[16] This sentiment analysis resembles some of the Topic modeling proposed in Appendix section 1 on NLP. While beyond the scope of this project, we acknowledge such techniques may be useful for both descriptive and predictive insights.

4 Data

4.1 Donors Choose Data

Data sources used in this project are described in Fig. 1. The Donors Choose Kaggle dataset includes 5 tables: `donations`, `essays`, `resources`, `outcomes`, `projects`. The `projects` table describes the basic information of each project, such as date posted, school, teacher, focus areas, etc. There are over 660,000 projects from 2000-2014. The `donations` table describes 3,097,989 donations dating from 2000-2014 and was used for label generation by joining with the project table. The other 3 tables each describe a different aspect of the projects. The `essays` table contains text information of each project (e.g. title, description, essay, etc.). At present, we only use the `word count`. The `resources` table stores records of items requested in each project. It is relatively unused in this version of the pipeline, though past Kaggle projects have shown it useful. The `outcomes` table contains analyzed results of each project, including if it is fully funded and donation characteristics. We do not use it, as we recalculate our own labels more suitable to the prediction problem of identifying unfunded projects at the time of posting.

Database Schema/Table	Years	General Description	Use
Donors Choose tables: • projects • essays • donations • resources • outcomes	2002-2014	All (anonymized) data collected on projects, their associated essays, resources requested,donations, and outcomes	Projects, Essays, used for features Donations used for labels Resources and Outcomes not currently used
Historical Features Table	2002-2009	Calculated features for historical funding by school, teacher, subject, resource	All historical features in use
Zip-School-Demographics Table	2010 & 2012	Racial demographics by zip code (ACS '12) Racial student enrollment by school (NCES '10)	Stored in database, not implemented in ML Pipeline

Figure 1: All data sources and uses

4.2 External Data Sources

A table was created of demographic information (race, poverty) from the American Community Survey 2012 joined on zipcode to the `projects` table.[17] This table was joined to National Center for Education Statistics Common Core Data (CCD 2010) on NCES School ID.[18] These tables were stored in the `school_zip_demographics` table of the course database. Ultimately, due to time constraints, neither of these data sources were incorporated into the final feature master list.

5 Solution

5.1 Label Generation

Labels for `unfunded_projects` are set to '1' if the project was unfunded within 120 days of project posting. Projects that are funded are set to '0.' This is achieved by joining the `projects` and `donations` tables by project and setting `unfunded_projects` to '1' if the following two conditions are satisfied.

$$\begin{aligned} \texttt{end\_date} - \texttt{start\_date} &<= 120 \ \texttt{days} \\ \texttt{donations} &>= \texttt{total\_project\_price} \end{aligned} \tag{5}$$

Note that label generation involves a 120-day waiting period between project posting and determining the status of `unfunded_projects`. This requires an intentional approach to creating train and test sets, explained below.

Train-Test Sets We utilize 8 train-test set pairs with data from 2009-2013, which constitute 83% of projects posted. Data is discarded prior to 2009 due to sparsity and after 2014 as it only includes 5 months. Train and test sets are designed with both an expanding training window and varying months to capture temporal funding patterns. Training sets always begin on 1 Jan 2009, with end dates spanning 2011-2013. Due to label-generation, training sets always end exactly 120 days before test sets begin. Test sets are always one month and begin on 1 Jan 2012, spanning 2012-2013. Like training sets, test labels are generated 120 days from the last project included in the set. Throughout train-test sets, the number of training examples increases by three months, while the test window shifts by that same amount. Lastly, historical features and pseudo-labels (`unfunded_project_trends`) include data up to 120 days prior to 1 Jan 2009. Train-test sets are shown conceptually in Fig. 2. See Table 1 for exact dates.

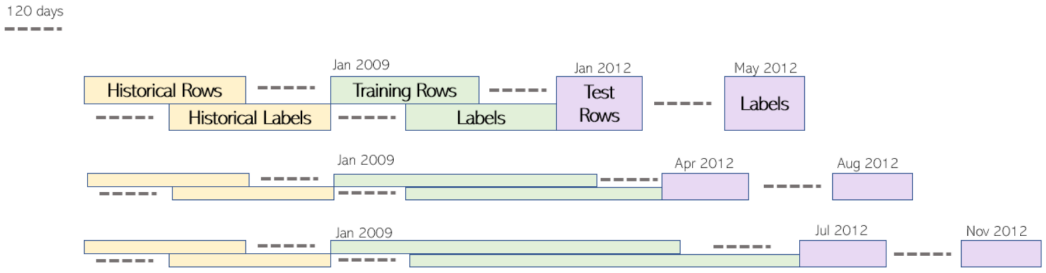

Figure 2: Train Test Splits along with Calculation of Historical Features and Labels

Table 1: Train Test Splits (Both start date and end date are inclusive)

Train Set		Test Set	
Start Date	End Date	Start Date	End Date
2009-01-01	2011-09-02	2012-01-01	2012-01-31
2009-01-01	2011-12-02	2012-04-01	2012-04-30
2009-01-01	2012-03-02	2012-07-01	2012-07-31
2009-01-01	2012-06-02	2012-10-01	2012-10-31
2009-01-01	2012-09-02	2013-01-01	2013-01-31
2009-01-01	2012-012-01	2013-04-01	2013-04-30
2009-01-01	2013-03-02	2013-07-01	2013-07-31
2009-01-01	2013-06-02	2013-10-01	2012-10-31

Features We use the following feature groupings: project choices, socioeconomic determinants, and historical trends. The full set of features is displayed in Table 2, below. One-hot encoding is used

Table 2: Features Used for Our Solution

Feature Group	Feature Name	Description
Socioeconomic Determinants	School Metro	urban, suburban, or rural
	Poverty Level	low, moderate, high, highest
	School State	all 50 states
	School Zip1	The first digit of school zip code
	School Latitude	degrees, normalized
	School Longitude	degrees, normalized
Project Choices	Resource Type	Technology, Supplies, Trips, Books, etc.
	Primary Focus Subject	Music, Performing arts, Sports, etc.
	Primary Focus Area	Math & Science, Music & the Arts, Health & Sports, etc.
	Total Price Excluding Optional Support	Total funding requested.
	Essay Length	Number of words in the project essay.
Historical Trends	Unfunded Ratio of the School	Unfunded percentage at the school within the past year of the project posting date.
	Unfunded Ratio of the Teacher	Same as above except aggregated over the teacher that posted the project.
	Unfunded Ratio of the Subject	Same as above except aggregated over the Primary Focus Subject.
	Unfunded Ratio of the Resource Type	Same as above except aggregated over the Resource Type.
Others	Teacher Prefix	Mr., Ms., Dr., etc.

to represent categorical features. For any non-categorical feature x, we standardize it by $(x - \mu)/\sigma$, where μ and are the mean and standard deviation of the corresponding feature column from the sample matrix. Historical features are generated by creating ratios of `unfunded` historical projects, teachers, subjects, as well as by counting the number of projects historically `unfunded` in each school and total donations within 1 and 2 years of project posting.

Baseline Model The logical baseline model predicts a project will go unfunded if `resource_type = technology` and `school_metro = rural`. This intelligent baseline was informed by exploratory data analysis, including a correlation matrix and by examining the joint distribution of unfunded projects over these attributes.

Experiment Models & Loss Functions We experimented with the following model types: Random Forest, Logistic Regression, Support Vector Machines, and Decision Trees. We trained 167 hyperparameter combinations, which can be found in Appendix Fig 5. All loss functions were chosen as those appropriate for a binary classification problem.

Random Forest models were trained by varying the number of estimators, max depth, min samples split, and max features. Decision trees were varied only by max depth. Both Random Forests and Decision Trees optimized loss functions with 'binomial deviance' (a.k.a. negative binomial log-likelihood).[19] Logistic Regression was trained using two solvers: lgfgs and liblinear. Logistic regression was optimized using 'logistic loss' (also a.k.a. negative log-likelihood), for three regularization scenarios: none, Ridge, and Lasso.[20] As a regularization using the ℓ_1-norm, Lasso both selects and shrinks features, whereas Ridge uses the ℓ_2-norm and only shrinks features.[21]

Regularization constant (λ) values are varied from 0.001 through 210 to estimate varying levels of shrinkage. Lastly, a Linear Support Vector Classifier was constructed. Unlike the other models which predict probability subject to a cutoff threshold, the LinearSVC attempts to classify linearly separable data and thus classifies samples as '0' or '1' (this is preferred to reduce computational complexity – probability outputs would require Platt-scaling to fit classifications with a logistic regression, which scales poorly with our tens of thousands of samples).[22] The LinearSVC uses a Squared-Hinge loss function, which rewards correct predictions with certainty, more harshly penalizing incorrect or uncertain classifications.[23] Note that in a more sophisticated setup, we might design a custom loss function to model the reward/cost of true/false negatives vs. true/false positives, but here we treat these issues with evaluation metrics instead.

6 Evaluation

6.1 Overview

We choose precision at the 'top 5%' of predictions as the ultimate model evaluation metric. As discussed in section 2.1, this refers to the 5% of predictions with the highest probability of `unfunded_project = 1`. We choose precision because 1) it can serve both as a measure of effectiveness and efficiency, and 2) Donors Choose has a limited number of projects it can provide resources to each month, so we need not identify every last project that will not be funded. In a future iteration of this project, we may wish to sweep the 'top k%' of predicted projects and compare performance. This is especially so, as the PR-K graph in Fig. 5 shows modest gains to recall with smaller marginal decreases in precision.

6.2 Selected models

Results shown in Fig 3 illustrate that most models perform better than the sensitive baseline (black dotted line). The random forest perform best on average followed by Logistic Regression and the LinearSVC. This is consistent with findings in the literature.

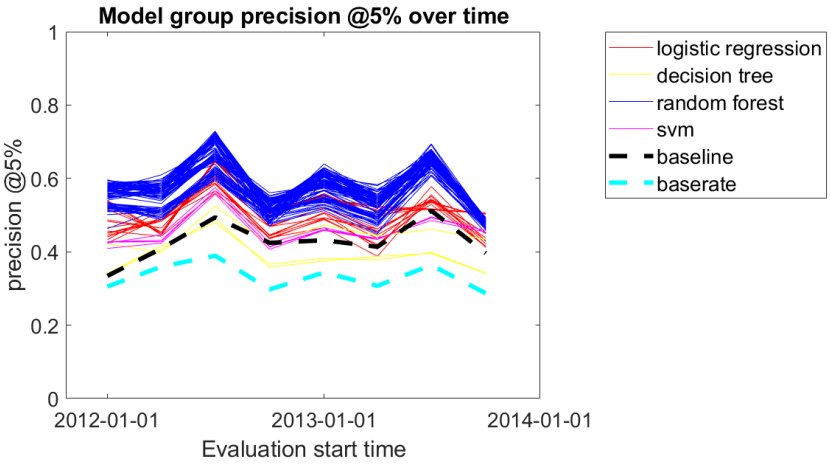

Figure 3: All models precision @5%

The top 5 selected models shown in Fig 4 are all Random Forests, which each share similar parameters. The top model correctly predicts 59% of all predicted `unfunded` projects over all 8 train and test sets. This is an average 26% point improvement over random selection and a 16% point improvement over the intelligent baseline. Our best model's parameters are listed in Table 3.

Table 3: Best model (random forest) parameters

n estimators	max depth	min samples split	max features
500	50	50	log2

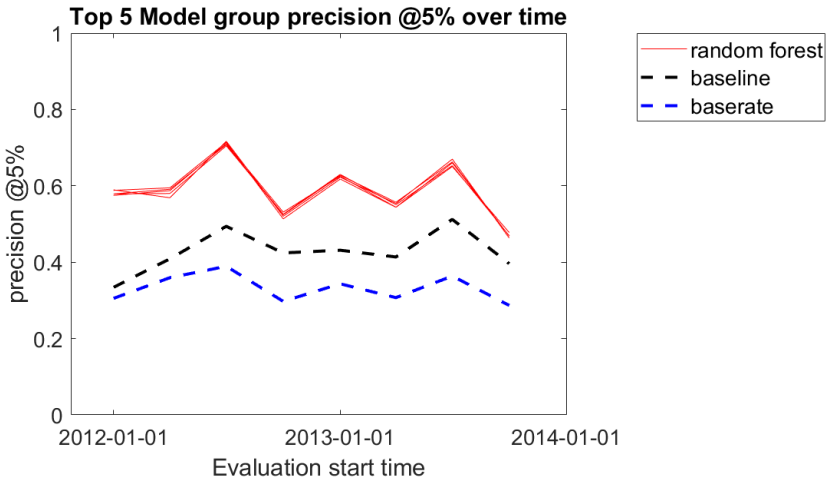

Figure 4: Top 5 models precision @5%

We also show the Precision-Recall-K (PR-K) graph used to balance trade-offs in Fig. 5. Note the high level of precision at the low 5% threshold, with very few of the total `unfunded` projects being discovered. This is acceptable to us due to the small number of projects being selected by the organization for intervention. But as stated, with more time we would have also compared precision across models at various cutoffs. To view PR-K graphs for all top models, see Appendix Fig.15.

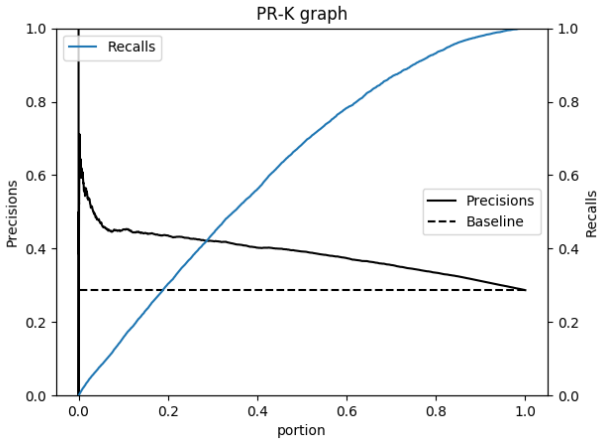

Figure 5: PR_k graph of top 1 model

Lastly, note that the top 5% precision varies cyclically over time, within a range of about 24% precision points for the trained models and about 18% precision points for the baseline. We hypothesize this may be due to surges in the sample sizes during "back-to-school" project postings in August and during the holiday season in December.

6.3 Important features

To interpret our models, we calculated feature importance values using a scikit-learn library, which calculates importance from "the expected fraction of the samples features [minus] the impurity from splitting them."[24] The top 10 features of importance in the best model are shown in Fig. 6. The most important feature is `Total price` which weighs 16.7% among all features. `Total price` is a feature mutable by the teacher. Our second most important feature is the `Historical ratio of projects unfunded by resource`, which weighs 8.4% among all features. While this historical ratio is immutable, the choices of which resources to select can be changed by the teacher. An

example of a true immutable feature of importance to the model is `School_metro = urban`. Note that the feature importance model shown does not show positive or negative importance. Implications of these findings are discussed below.

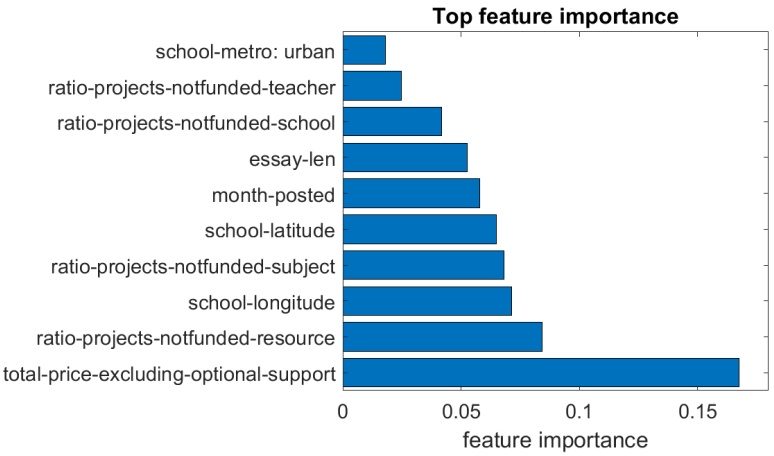

Figure 6: Feature Importance

6.4 Cross-Tabulated Results

Mean attribute values are calculated for the feature attributes with the greatest normalized mean difference between the top 5% and bottom 95% of predictions, shown in Table 4. We see that for the top 5% of projects, `Total price` for `unfunded` projects averages $7,391.62, while the average price of projects in the bottom 95% is $794.77. Foreign languages and Technology projects each independently constitute 43% of predicted `unfunded_projects`, while representing 2% and 3% of `funded_projects`, respectively. Other types of projects that are more represented by a factor of 10 in the predicted `unfunded_projects` include Special Needs, Applied Sciences, rural projects, and projects in the state of Hawaii and Mississippi.

Table 4: Cross-tab of 10 top features

Feature Name	bottom 95%	top 5%
Primary focus subject: Foreign Languages	0.02	0.43
Resource type: Technology	0.03	0.43
Essay length	1654.90	1861.70
School zip 1st digit: 6	0.03	0.29
Primary focus area: Special Needs	0.03	0.29
Total price excluding optional Support	794.77	7391.62
Primary focus subject: Applied Sciences	0.001	0.14
School metro: rural	0.01	0.14
School state: HI	0.02	0.14
School state: MS	0.02	0.14

6.5 Bias audit

To evaluate bias, we evaluated the False Discovery Rate (FDR) Ratio and False Negative Rate (FNR) ratio for our models based on their `poverty` attributes. The protected group was taken as highest poverty with the reference group as moderate poverty. Note that FNR is a more important metric in this problem, because false negatives mean failing to detect a project that may need help, while false discovery means giving more resources to a project that may not need it. We see in Fig. 7 that our selected model has greater precision than the intelligent baseline (as already discussed), but also a

lower FDR Ratio. This means that the highest poverty schools that are likely to be funded are less likely to be falsely predicted as `unfunded_projects` than moderate poverty projects, compared to the baseline. This is not necessarily a positive equitable outcome, but also does not seem to cause immediate harm. More importantly, the FNR Ratio (0.997) for the selected model, indicates that it is about equally likely to predict false negatives for the highest poverty and moderate poverty schools. This satisfies the definition of "Equal Opportunity" in section 2.1.

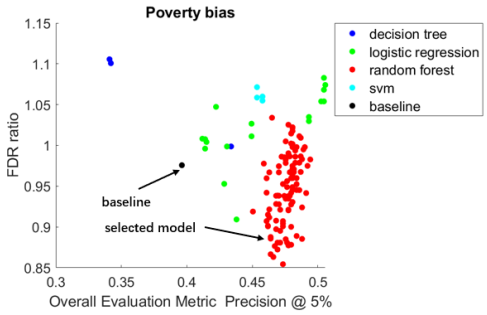

Figure 7: FDR of poverty

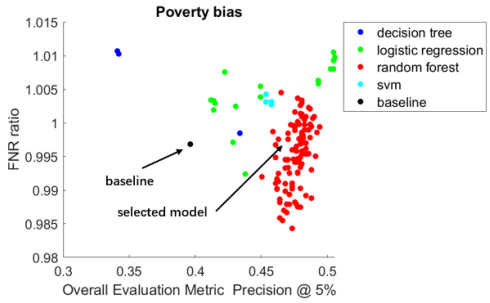

Figure 8: FNR of poverty

7 Discussion of Results & Recommendations

Based on the average precision at the top 5% of predictions over all test sets, our 5 best models are all Random Forests. A possible explanation is that Random Forest's nonlinear nature gives it an advantage over other models. However, whether Random Forest is best suitable for this problem is still debatable because the hyperparameter grid we used for Random Forests is larger than those of other models. We also did not test any Neural Networks due to time constraints, which may have performed better, as indicated by the literature.

Considering that the bias analysis showed favorable results for our top selected model, we feel confident in making the following recommendations based on the top 5% of predictions per month:

For projects with highly important immutable feature attributes (*e.g.*, `School metro = rural`, `School state:HI or MS`): Allocate advertising and funding resources to help secure project funding. Advertising and funding resources could include: placing high-risk projects next to successful projects on the main page, working with businesses to add matching offers to certain projects, advertising them to potential donors.

For projects with highly important mutable feature attributes (*e.g.*, total price of project $\gg$ \$700): Auto-screen the project and dedicate staff hours to help improve it. For attributes in features such as project resources and subject types that are unlikely to be funded (`Foreign Languages, Technology`), Donors Choose could incentivize teachers to choose more funding-likely attributes. These attributes could be identified through cross-tabulation at the bottom 5% of predictions, or perform the inverse classification task of that addressed in this project to identify such attributes.

Both of these recommendations are subject to field trials that 1) validate the classifier's predictions, and 2) estimate the effectiveness of the actual interventions. Our project provides an entire modular pipeline that can be reused and converted, if needed. for instance, if models begin performing poorly in new test periods, the models could be retrained. New intervention goals can be defined. And most importantly, decision makers should develop a process to thoughtfully integrate these machine learning recommendations into their operating procedures rather than simply supplanting all existing systems with this prediction tool.

8 Caveats and Future Work

There are three aspects of future work and limitations that should be considered upon evaluation of this course project.

Data As stated in the literature review, there is some benefit to incorporating real-time data that is generated within the initial moments of a project being approved. While this would change our problem formulation of predicting `unfunded_projects` at the time of posting, it is worth considering for improved performance. Additionally, future implementations should include more features such as the demographic features included in the unused `zip_school_demographics` table. Natural Language Processing features generated from the `Essays` table may improve performance (see Appendix section 1).

Models Future models should be more sophisticated and include neural networks (NN), as the literature has shown them to be most effective in predicting crowdfunding success. While we initially implemented NN with various hyperparameter combinations, we later discarded them to to lack of improvement for the significant computation time trade-off. Now that the pipeline is successful, these models should be re-considered.

Evaluation As discussed above, a more systematic method is suggested to choose the 'top k%' of predictions to classify as `unfunded_projects`. This is especially so, given the nature of the PR-K graph, which shows relatively steady precision with relaxing the threshold, thus increasing recall. Future iterations of this project should better examine the PR-K curve and precision across models at various cut-off thresholds to determine the exact point suitable to Donors Choose's prediction problem.

References

[1] Scoping data (for social good) projects. (n.d.). http://www.dssgfellowship.org/2016/10/27/scoping-data-science-for-social-good-projects/

[2] Azzam, A. M. (2005). Special report / the funding gap. http://www.ascd.org/publications/educational-leadership/feb05/vol62/num05/-The-Funding-Gap.aspx

[3] Meckler, L. (2019). Report finds $23 billion racial funding gap for schools. https://www.washingtonpost.com/local/education/report-finds-23-billion-racial-funding-gap-for-schools/2019/02/25/d562b704-3915-11e9-a06c-3ec8ed509d15_story.html

[4] Barry Newstead, P. W. (2009). Nonprofits in rural america: Overcoming the resource gap. https://www.bridgespan.org/insights/library/funding-strategy/nonprofits-in-rural-america-overcoming-the-resourc

[5] Donors choose. (n.d.). https://www.donorschoose.org/about

[6] Martin, A., Gross-Camp, N., Kebede, B., & McGuire, S. (2014). Measuring effectiveness, efficiency and equity in an experimental payments for ecosystem services trial. *Global Environmental Change*.

[7] Sahil Verma, J. R. (2018). Fairness definitions explained. https://dl.acm.org/doi/10.1145/3194770.3194776

[8] Kdd cup 2014 - predicting excitement at donorschoose.org. (2014). https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose

[9] Donorschoose.org application screening. (n.d.). https://www.kaggle.com/c/donorschoose-application-screening/overview

[10] Zafar. (2018). Beginner's guide to capsule networks. https://www.kaggle.com/fizzbuzz/beginner-s-guide-to-capsule-networks

[11] Lukyanenko, A. (2019). Eda, feature engineering and xgb + lgb. https://www.kaggle.com/artgor/eda-feature-engineering-and-xgb-lgb

[12] Anderson, M. (2018). Ensembling with logistic regression (lb 82.4%). https://www.kaggle.com/matthewa313/ensembling-with-logistic-regression-lb-82-4

[13] Data science for good: Donorschoose.org. (n.d.). https://www.kaggle.com/donorschoose/io

[14] Yu, P.-F., Huang, F.-M., Yang, C., Liu, Y.-H., Li, Z.-Y., & Tsai, C.-H. (2018). Prediction of crowdfunding project success with deep learning, In *2018 ieee 15th international conference on e-business engineering (icebe)*. IEEE.

[15] Tran, T., Dontham, M. R., Chung, J., & Lee, K. (2016). How to succeed in crowdfunding: A long-term study in kickstarter. *arXiv preprint arXiv:1607.06839*.

[16] Sudhir, K. (n.d.). Machine learning on crowdfunding platforms to understand drivers of altruism. https://economics.yale.edu/undergraduate/tobin/spring-2019/machine-learning-crowdfunding-platforms-understand-drivers-altruism

[17] American community survey (acs). (2012). https://www.census.gov/programs-surveys/acs
[18] National center for education statistics common core of data (ccd). (2010). https://nces.ed.gov/ccd/
[19] 1.11. ensemble methods. (n.d.). https://scikit-learn.org/stable/modules/ensemble.html
[20] Sklearn.metrics.log$_loss$. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html
[21] Nagpal, A. (2017). L1 and l2 regularization methods. https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c
[22] Sklearn.svm.linearsvc. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[23] Squared hinge. (n.d.). https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/squared-hinge
[24] 1.11. ensemble methods. (n.d.). https://scikit-learn.org/stable/modules/ensemble.html
[25] Urllib3. (n.d.). https://urllib3.readthedocs.io/en/latest/
[26] Requests: Http for humans, release v2.23.0. (n.d.). https://requests.readthedocs.io/en/master/
[27] Scrapy. (n.d.). https://scrapy.org/
[28] Beautiful soup documentation, 4.9.0. (n.d.). https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[29] Muthukadan, B. (n.d.). Selenium with python. https://selenium-python.readthedocs.io/
[30] Scrapy spiders. (n.d.). https://docs.scrapy.org/en/latest/topics/spiders.html#topics-spiders
[31] Geitgey, A. (2018). Natural language processing is fun! https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e
[32] Chen, G. H. (n.d.). Basic text processing with spacy. https://gist.github.com/georgehc/316773984309d0a0739e4e8cc16617ee
[33] Working with text data. (n.d.). https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
[34] Word_cloud. (n.d.). https://github.com/amueller/word_cloud
[35] Blei, D. M. (2012). Probabilistic topic model. *Communications of the ACM*. http://www.cs.columbia.edu/~blei/papers/Blei2012.pdf
[36] Kelechava, M. (2019). Using lda topic models as a classification model input. https://towardsdatascience.com/unsupervised-nlp-topic-models-as-a-supervised-learning-input-cf8ee9e5cf28

Appendix

1 Utilizing Natural Language Processing

Due to time constraints, we did not use potential features from the `Essays` table, other than the calculated feature: `total_word_count`. Future work could use Natural Language Processing to generate many more information-bearing features. Because project data originated on Kaggle, the `Essays` table already presents the textual content of each project loaded to the database.[8] Suppose this table was not given. The researchers could use a standard web-scraping approach. Libraries such as `urllib.request`[25] and `Requests`[26] may be useful for sending HTTP requests. `Scrapy`[27] and `BeautifulSoup4`[28] can be used to cleanly navigate and extract HTML data, and `Selenium`[29] allows browser automation and form filling. If the site has an unknown number of pages that require scraping, a method of "pagination" should be used either to scroll through numbered pages, using the 'next button,' or by using a Queue data structure to crawl through all linked pages. There are several good tutorials available such as Web-Scraping for Machine Learning with BeautifulSoup4 and How to Crawl Infinite Scrolling Pages Using Python. Pruning or cross-referencing already visited pages may be necessary to prevent infinite scrolling. Scrapy provides a `Spider` Class to define custom crawling and extraction patterns.[30]

The textual data loaded into the database constitutes a linguistic corpus on which Natural Language Processing (NLP) methods can be used for exploratory data analysis, unsupervised learning, and supervised learning goals. Pre-processing steps generally include sentence segmentation, word tokenization, lemmatization (conversion to root form), and vectorization and word-dependency parsing ("Bag of Words," n-grams, discussed below).[31] Several of the libraries mentioned above, such as Scrapy, are capable of this functionality[32] and Sci-Kit Learn also offers libraries for working with text data.[33]

Once pre-processing is completed on the corpus table of text data, options are available for the problem at hand. Exploratory data analysis can help describe overall trends in the text and may include metrics such as word occurrences, word frequencies, encoding of word rarity per document by the "Term Frequency-Inverse Document Frequency" (TF-IDF) transformation,[33] and visualizations such as word clouds.[34]

For practical problems of pattern recognition, the researcher could perform a matching search using regular expressions on each page to find desired words or pairs. Pre-processing using the "Bag of Words" vectorization method is a form of one-hot encoding the presence of all words in each document, irrespective of the context or order in which words appear.[33] If phrase or word-context is a concern, n-grams could be used. In terms of our problem of predicting Donors Choose projects unlikely to be funded, we must engineer features for our supervised binary classifier. Since we already used essay length, a more comprehensive approach would be to create categorical features for words and phrases of interest. One method for selecting features is to use the TF-IDF transformation to restrict selected language features to those in the middle of the frequency band, filtering overly rare and frequent words.[33] Sci-kit Learn provides off-the-shelf methods for this methodology. Overall, since we are currently limited to about 250 features, another method of categorizing essays may be more useful than including a long list of individual word features.

Topic models are an unsupervised learning approach that can be used discover unobserved themes in text documents.[35] Examples of topic models include Latent Semantic Analysis (LSA/LSI) and Latent Dirichlet Allocation (LDA), a more modern approach. LDA is a generative process which maximizes the posterior probability of a "topic distribution, given the observed documents."[35] The output of LDA includes word assignments and topic distributions. This information is valuable on its own but could also be converted to categorical features for our prediction task.[36] Sci-kit learn also provides an LDA library and a good tutorial can be found here: End-to-End Topic Modeling in Python with LDA. Beyond LDA, more sophisticated Topic models include the Author-topic model, which may include probabilistic modeling of authors' past work. One may also consider the Spherical topic model, which augments LDA's capabilities with negative topic associations.[35] These are only some of the ways we could generate useful features for the Donors Choose project classifier. An alternative problem statement may warrant selection of different methods.

2 Pipeline Architecture

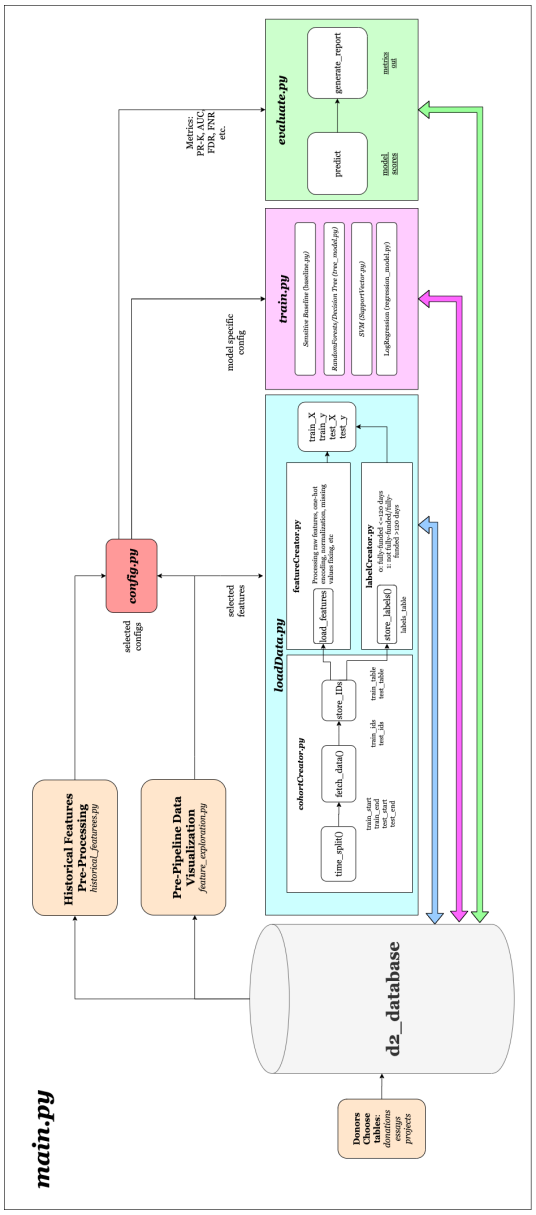

Figure 9: Pipeline Architecture

3 Model Grid

Table 5 gives the model types and hyperparameter choices.

Table 5: Model Type and Hyperparameters

Model Type	Hyperparameter Name	Hyperparameter Values
Random Forest	Number of Estimators	$100, 500, 1000$
	Max Depth	$2, 5, 10, 50,$ None
	Min Samples Split	$2, 10, 25, 50$
	Max Features	sqrt, log2
Decision Tree	Max Depth	$10, 100, 200$
Logistic Regression	Penalty	None, Ridge, Lasso
	C (Regularization Strength)	$0.001, 0.01, 0.1, 1, 210$
	Solver	lgfgs, liblinear
Support Vector Machine	Kernel	rbf
	Penalty	ℓ_1, ℓ_2
	Loss	Squared Hinge
	Dual	False
	Tolerance	0.0001
	C (Regularization Strength)	$0.01, 1, 100, 10000$
	Max Iterations	10000

3.1 Top 5 Models

All our top 5 models are random forests. Their configurations are summarized in Table 6 and their PR_k graphs are listed in Fig. 10-14. The importance of all features can be found here in our drive.

Table 6: Top 5 Models (Random Forests) Parameters

n estimators	max depth	min samples split	max features
500	50	50	log2
1000	None	50	sqrt
1000	50	50	log2
1000	None	50	log2
500	50	10	sqrt

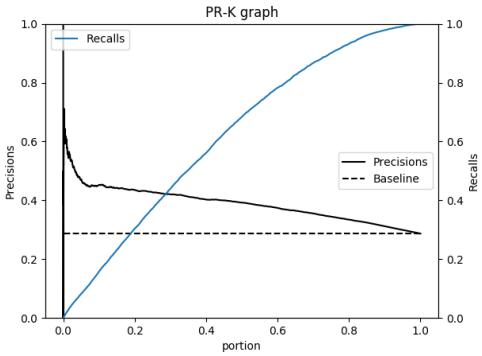

Figure 10: Top 1 model precision @5%

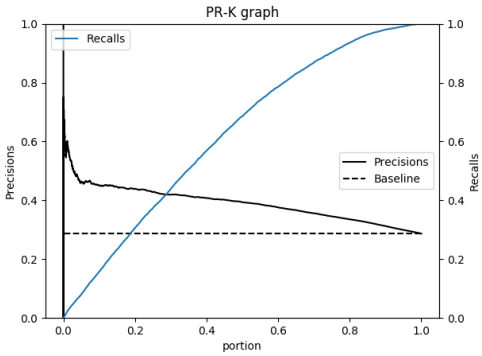

Figure 11: Top 2 model precision @5%

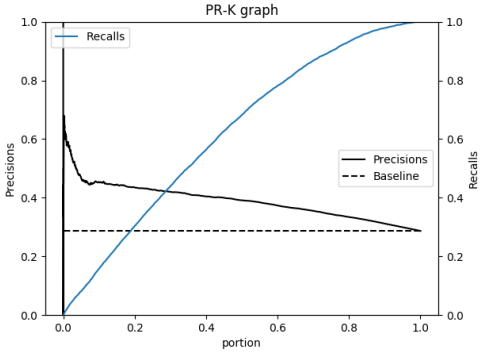

Figure 12: Top 3 model precision @5%

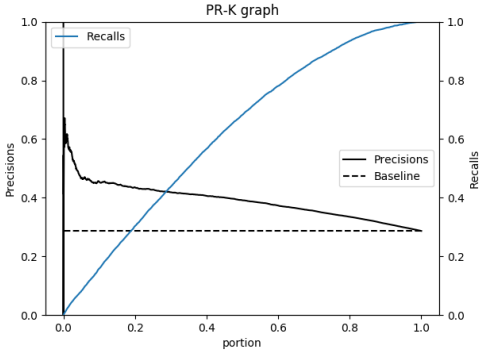

Figure 13: Top 4 model precision @5%

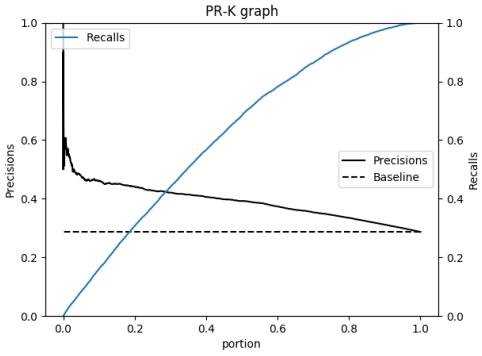

Figure 14: Top 5 model precision @5%

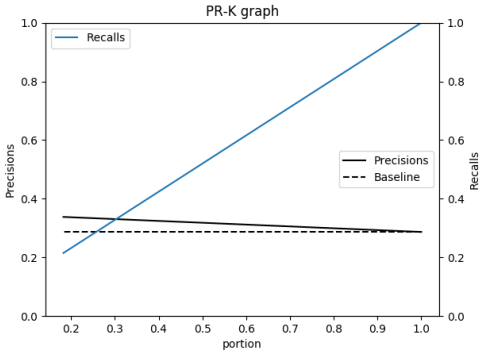

Figure 15: Baseline model precision @5%