Auto-Screening Project Proposals at DonorsChoose.org

PREPARED BY

Charles Guthrie
Justin Mao-Jones
Yasumasa Miyamoto
Lucy Wang
12.18.14

Introduction

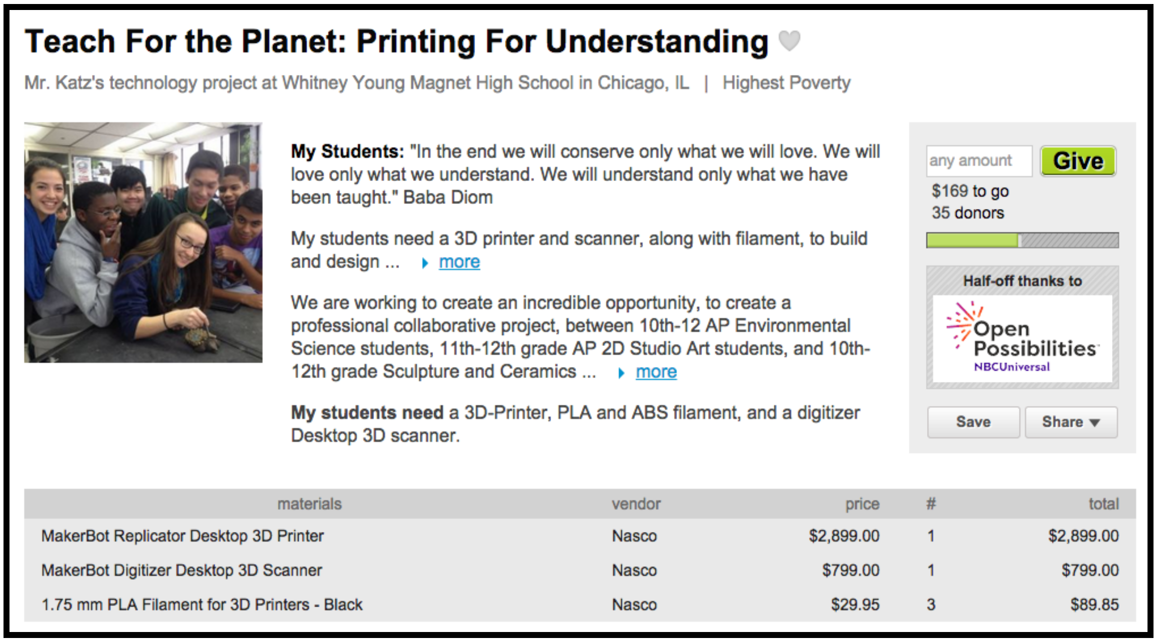
Donors Choose is an online donation marketplace that connects teachers to donors. From the Donors Choose about page[1], "Public school teachers post classroom project requests which range from pencils for poetry to microscopes for mitochondria," and individual donors give to projects that appeal to them. Since inception in 2000, Donors Choose has channeled over $283 million in funding and hosted over 700,000 projects on its website with 70% of projects fully funded.

In order to ensure the integrity of its website and hosted projects, Donors Choose employs a "small army" of 50 volunteers to screen projects that teachers submit. Proposals that meet screening requirements are posted and the rest are sent back to teachers for revisions. Unfortunately, this screening process is labor-intensive, and the number of proposals is growing.

[1] http://www.donorschoose.org/about

Originally motivated to work on a project related to Donors Choose, our team got in touch with the Donors Choose Head of Data Science, Vlad Dubovskiy, who suggested that we develop a method for reducing this workload with an automated screening process. This idea forms the basis of our project.

The auto-screening process would approve as many “good” proposals as it could find while flagging the remainder for manual review.



Teach For the Planet: Printing For Understanding

Mr. Katz's technology project at Whitney Young Magnet High School in Chicago, IL | Highest Poverty

My Students: "In the end we will conserve only what we will love. We will love only what we understand. We will understand only what we have been taught." Baba Diom

My students need a 3D printer and scanner, along with filament, to build and design ... ▸ more

We are working to create an incredible opportunity, to create a professional collaborative project, between 10th-12 AP Environmental Science students, 11th-12th grade AP 2D Studio Art students, and 10th-12th grade Sculpture and Ceramics ... ▸ more

My students need a 3D-Printer, PLA and ABS filament, and a digitizer Desktop 3D scanner.

any amount | Give

$169 to go
35 donors

Half-off thanks to
Open Possibilities
NBCUniversal

Save | Share ▾

materials	vendor	price	#	total
MakerBot Replicator Desktop 3D Printer	Nasco	$2,899.00	1	$2,899.00
MakerBot Digitizer Desktop 3D Scanner	Nasco	$799.00	1	$799.00
1.75 mm PLA Filament for 3D Printers - Black	Nasco	$29.95	3	$89.85

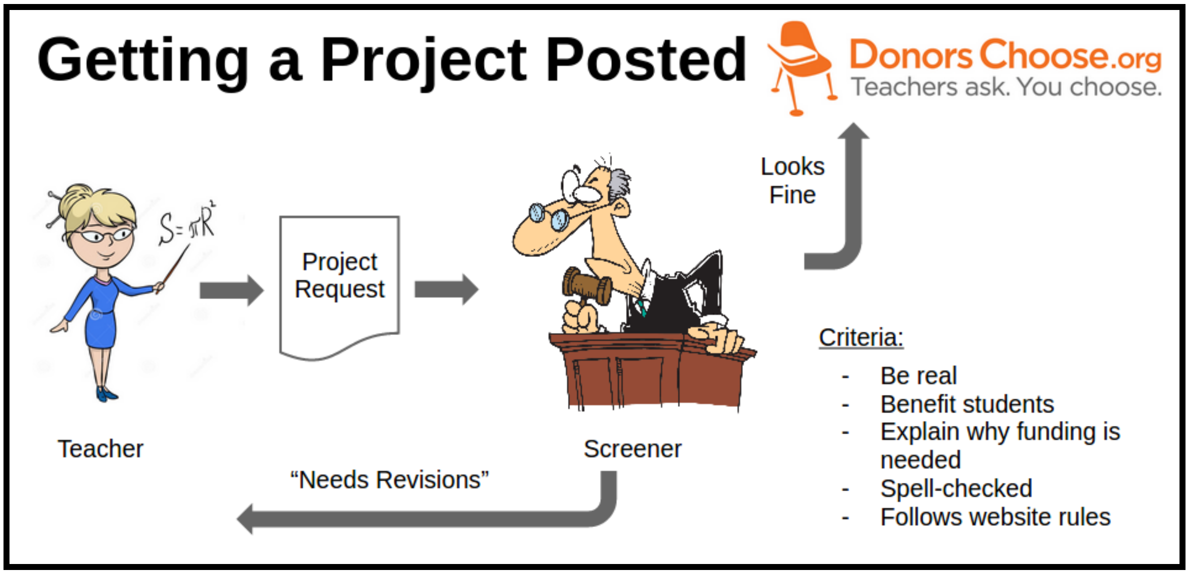
A posted project on DonorsChoose.org

The Donors Choose Screening Process

Each project proposal must go through a rigorous screening process to ensure the integrity of the charity and use of funds.

All ‘front-line educators’ that work directly with students for at least 75% of their time are eligible to create accounts on the site and post project requests. This includes teachers, librarians, guidance counselors, school nurses and full-time teachers who also act as coaches. To register for an account, teachers, from a list of pre-approved schools, go through a third party verification process. In the case the third party verification cannot be completed, Donor’s Choose will call the school principal to verify manually.

In order to get a project posted on the site, a teacher submits a project proposal that is reviewed and screened. Only approved projects are allowed onto the site for donors to see. A point system is put in place where the total amount of resources teachers are allowed to request increases as a teacher successfully funds more projects.

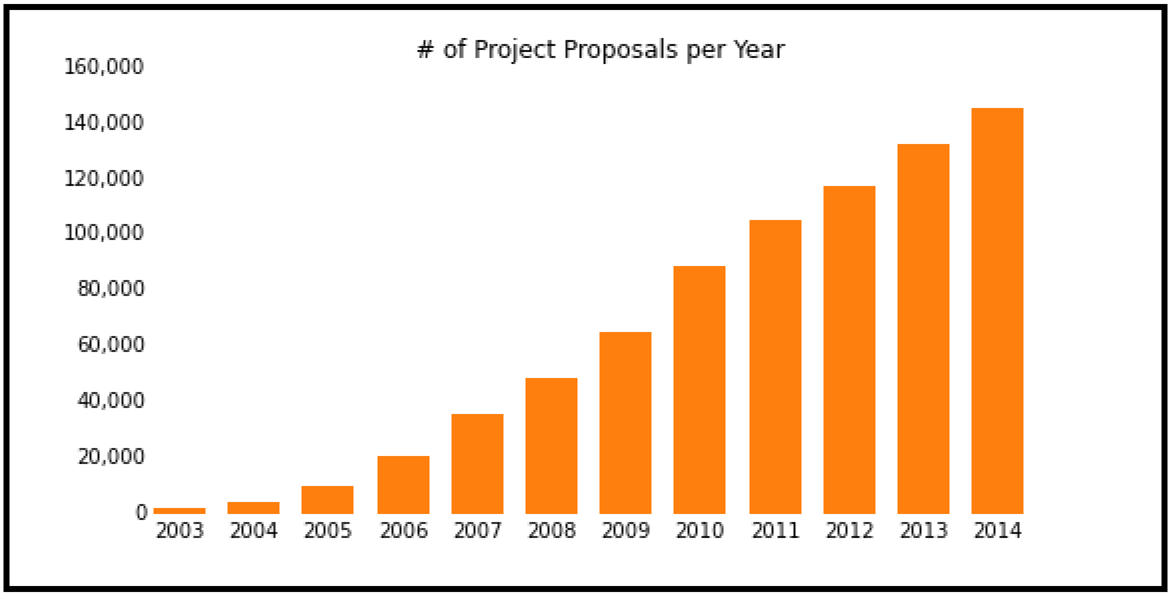


Each project proposal contains a list of resources for which teachers are requesting funds. After a proposal is submitted, an automated vendor system verifies the list of resources requested. For unverifiable resources such as museum tickets and field trip expenses, Donor's Choose manually verifies the requests. Projects with approved resource requests are then sent to a group of 50 volunteers that manually read through every project proposal. The review process takes between 2 – 4 weeks to complete.

Approximately 82% of proposals make it through the initial screening process. The remainder are sent back to teachers with suggested revisions. Teachers are notified of the sections that did not meet requirements. A proposal can go through multiple revisions. Ultimately, 97% of proposals are eventually approved and posted onto the website. Once posted, projects are eligible for donations.

Business Problem

Donors Choose has seen tremendous growth since its inception in 2000. The figure below illustrates the growth in the number of submitted proposals. In the first three quarters of 2014, for example, teachers have already submitted nearly 150,000 proposals.

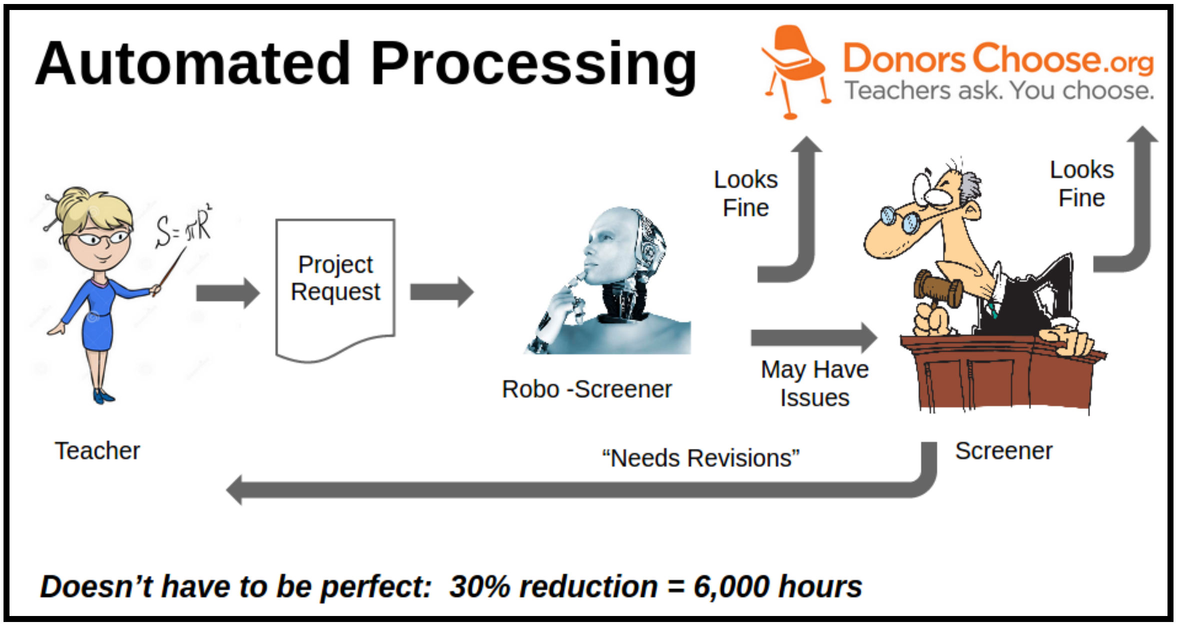


The screening process is incredibly labor intensive. According to Donors Choose, each proposal takes up to 8 minutes to process. At that rate, 150,000 proposals would require an estimated total 20,000 hours of volunteer time. Donors Choose employs a team of 50 volunteers to screen proposals. This does not include in-house paid staff needed to manage the effort.

As Donors Choose continues to experience high growth, it will become unsustainable to have all projects reviewed manually. A tool that can automate even part of the approval process will enable the organization to scale and support its growth. If a tool can automatically approve even just 30% of the "good" projects, at the current volume of projects, that translates to nearly 6,000 hours in savings.

Proposed Solution

We propose that Donors Choose would benefit from an automated process to reduce the overall screening workload. It would need to approve as many "good" proposals as it could find while flagging the remainder for manual review. Proposals would only ever be sent back for revision by a human reviewer and never by the auto-screener. The process would use a model trained on a set of features engineered from an archive of "approved" and "rejected" proposals. Approved proposals are those that passed the screening test (either immediately or after revision), while rejected proposals are those that did not make the cut.



Automated Processing

Doesn't have to be perfect: 30% reduction = 6,000 hours

No analytical model is perfect, including the one we propose here. While it may be more resource-efficient, the analytical model will not be better at judging approval worthiness of proposals than a team of volunteers. This means that an automated process will inevitably approve proposals that should have been sent back for revisions.

This raises an important business concern. Donors Choose website content integrity is of the highest priority. Posted projects with missing sections, unclear goals, spelling errors, etc. detract from donors' confidence in the website. Therefore it would be worse for the automated process to approve a "bad" project than to flag a "good" project. This means that decisions made by the auto-screener need to err on the side of caution. In other words, it needs to be very confident that a project it decides to approve is indeed good. Part of our analysis shows how one can adjust the "decision threshold" of the auto-screener to reach an appropriate tradeoff between approving good and bad projects. The decision of what is an appropriate threshold is beyond the scope of this project and left as an outstanding business question.

A strength of this process is that "good" essays that get flagged rather than approved will still get approved through the better judgment of the human screener. This process ensures that proposals are not unfairly rejected. This also prevents proposals from getting stuck in an infinite revision loop.

Project Scope and Longer-Term Planning

The success of the proposed solution, and indeed that of any analytical model, depends in large part on the quality of the data. Midway through the project, we discovered an important aspect about the nature of the proposal approval/rejection data. Donors Choose does not keep a record of the entire revision history of proposals. Whenever a teacher submits a new revision of a proposal, the previous revision is overwritten. This means that the data shows only the final revision state of the proposal. According to Mr. Dubovskiy, approximately 18% of new proposals require revision; but only 3% of proposals are available in the data labeled as “rejected” examples. That is a significant difference.

For any given approved proposal in the data, there is no telling what it originally looked like in the first submission. It could be a proposal that never required a revision, or a proposal that went through 5 revisions. Likewise, rejected proposals in the data could be the 1st, 2nd, 3rd, etc. revision of that proposal. We do not have a clear picture of what proposals look like in their original form.

This poses a significant hurdle in our ability to produce a useful model using currently available resources. Therefore, the scope of this project is a partial feasibility study. It demonstrates how such a model could be produced and implemented, given the proper data. The results could be used to decide whether more investigation into developing an auto-screening process would be worthwhile for Donors Choose.

A proper feasibility study should also include resource and schedule projections for development, prototyping, and implementation. This would require analysis beyond the scope of the class project instructions. As such, these kinds of projections are left out of this study.

Data Understanding

Data Sources

The data consist of 777,014 projects, identified by a unique **projectid**. The target variable is whether a proposal was approved or rejected, as indicated by a **got_posted** field. These datasets include all approved and rejected projects from 2000 to present. A comprehensive list of available data columns is in Appendix II.

The full dataset is split into 754,887 approvals and 22,047 rejections. As mentioned earlier, the apparent low rate of rejection is deceptive.

There are three primary data sets available for use to address the data-mining problem:

- ➢ *Project Essay Data (4GB);* full text of the teacher-written requests accompanying all classroom projects, indexed by **projectid**. This is the heart of each teacher's classroom project proposal. It includes:
  - o Title
  - o Short Description - Summary description, typically authored by the (volunteer) project screener by copy-pasting excerpts from the teacher's essay or paragraph answers.
  - o Needs Statement - Summary of the materials/resources needed, eg. "My students need…"
  - o Essay (Free-form or paragraph form)

    Paragraph 1 - Open with the challenge facing your students.
    Paragraph 2 - Tell us more about your students.
    Paragraph 3 - Inspire your potential donors with an overview of the resources you're requesting
    Paragraph 4 - Close by sharing why your project is so important
- ➢ *Project Resource Data (900MB);* all materials/resources requested for the classroom projects, including vendor name. Each record corresponds to a single requested material/resource from a specific vendor. Most classroom projects request multiple resources. Since the item name and other item info are provided by the vendors and used only for display purposes in the system, the formats are often inconsistent across vendors and over time.
- ➢ *Project data (500MB)*; all classroom projects including 45 columns of data about the school. This was explored but ultimately only the target variable was used in the model. Below is a list of highlighted metadata features.
  - o Approved/not approved ***← Target Variable***
  - o School characteristics (name, location, size, type, etc)
  - o Teacher attributes (name, teach for america boolean, teaching fellow boolean)
  - o Project categories (primary and secondary focus areas)
  - o Poverty level
  - o Grade level
  - o Project pricing (project costs, vendor, taxes, material list)
  - o Project impact (# students)
  - o Donation target amount

- Project donation attributes (funding status, dates, expiration)

Screening Criteria

Per Donors Choose requirements, a proposal write-up should include the following content:

- Description of the problem the teacher is facing.
- A solution to the problem with two components:
  - A description of the project and benefits for the students
  - A description of the resource materials needed to complete the project.
- An itemized list of the the requested resources.
  - Donors Choose maintains a system, linked to Amazon.com and other commercial websites, in which teachers add items to a “shopping cart”, which becomes the list of requested resources. Custom items not available in the system, such as field trips, are tagged as such.

To ensure fair treatment of proposals, screeners are only provided the proposal essays, need statements, and list of resource requested. Additional metadata, such as school location, number of students, etc., are withheld from the screeners unless this is specifically stated in the proposal write-up. We believe it is important for the auto-screener to follow these same guidelines, in order to ensure fair treatment of proposals.

Screeners are asked to apply the following criteria:

- Ensure that the write-up follows content guidelines.
- Identify inconsistencies between the lists of resources requested and the need statements, a summary of the resource requested.
- Remove/edit the following:
  - References to names or classroom numbers
  - Spelling errors
  - Strings of CAPITALIZED LETTERS
  - Email or web address links
  - Specific prices mentioned by the teachers (Donors Choose calculates the final price based on resource materials requested)

Proposals that fail any of these criteria to a subjectively large enough degree are sent back for revisions.

Data Preparation

During our data gathering process, we encountered a number of challenges. The process of gathering the data was complicated by our failure to set up a design session with the data science team at Donors Choose during the planning phase. Many of the issues we encountered with the data could have been avoided. Without a proper planning session, it took several weeks before we were able to obtain a complete data set with all of the necessary information.

The backend data system at Donors Choose was not set up for data analysis. For this reason, there were a number of issues associated with the datasets that needed to be dealt with before it can be used reliably for data mining. The following is a list of selected data issues we had:

- Donors Choose migrated from a legacy system in 2007. Pre-2007 projects data are unreliable and only post 2008 data should be used for analysis to ensure consistency;
- Projects pending approval are automatically assigned False for approval. Projects can be pending for up to 4 weeks. Any data created within the past 4 weeks should be removed;
- Essays include html tags and other irrelevant symbols. We were not provided with an exhaustive list of the symbols and extensive data exploration was necessary to ensure all unnecessary characters are removed;
- Each record in the dataset only represents the final version of the project. The loss of intermediary revision data presents a key challenge for us.

Munging

In order to prepare the data for modeling, significant data munging was required. The operations are summarized as follows:

1) Data Merge
    a. Merge data sets on **projectid**
    b. Combine datasets and eliminate inaccurate and unusable data
2) Clean data and add null indicators
    a. Eliminate inaccurate and unusable data (i.e. pre-2008 and anything younger than 4 weeks old)
    b. Text fields:

        i. Convert nulls to empty strings
        ii. Add indicator variable for null strings
        iii. Convert text dividers to string with one space
        iv. Convert strings used to represent absent data to empty strings

    c. Binary fields:

        i. Convert null values to 0
        ii. Add indicator variable for nulls

    d. Numerical fields:

        i. Convert nulls to 0 (very infrequent)
        ii. Add indicator variable for nulls

Given the size of the essay data sets, we typically waited to merge and clean data only after downsampling and selecting training/testing sets. Merges typically required iterating through large chunks of the dataset at a time. One team member learned a hard lesson when he attempted to run a merge script on the entire dataset, encountered a stack overflow, and performed a hard shutdown on his Ubuntu OS. Lesson learned: never do that.

Feature Engineering

The majority of the dataset is text-based. Therefore, a significant amount of effort was devoted to learning about and using natural language processing (NLP) methods for converting text into a numerical form that can be analyzed by a classification model. The underlying intuition behind these methods is that text in approved proposals and text in rejected proposals will have unique characteristics. These characteristics are found by coercing the text data into a numerical format and then putting it into a classification model.

We used the Python Natural Language Toolkit (NLTK) package http://www.nltk.org/ to perform the majority of our NLP operations. A summary of the NLTK operations we performed is as follows:

- Removed stop words and punctuations: Words such as “the”, “and”, “a”, etc., typically known as “stop words” and punctuation, would be so common in both approved and rejected proposals that they should be removed.

- Tokenized each essay into unigrams: Tokenizing means converting strings into a list of words, which is much easier to work with for the operations that follow.

- Stemming: Each token was stemmed using an implementation of Porter's stemming algorithm[2]. Common words with different suffixes and prefixes were converted to a common format. For example, "produced" and "production" would be converted to "produc". These words are similar enough to naively assume that they are the same.

The list of terms resulting from these processes can be analyzed to find terms that are distinctive of either approved or rejected proposals. To do this, we used the term frequency-inverse document frequency framework (TFIDF) in Scikit-Learn.

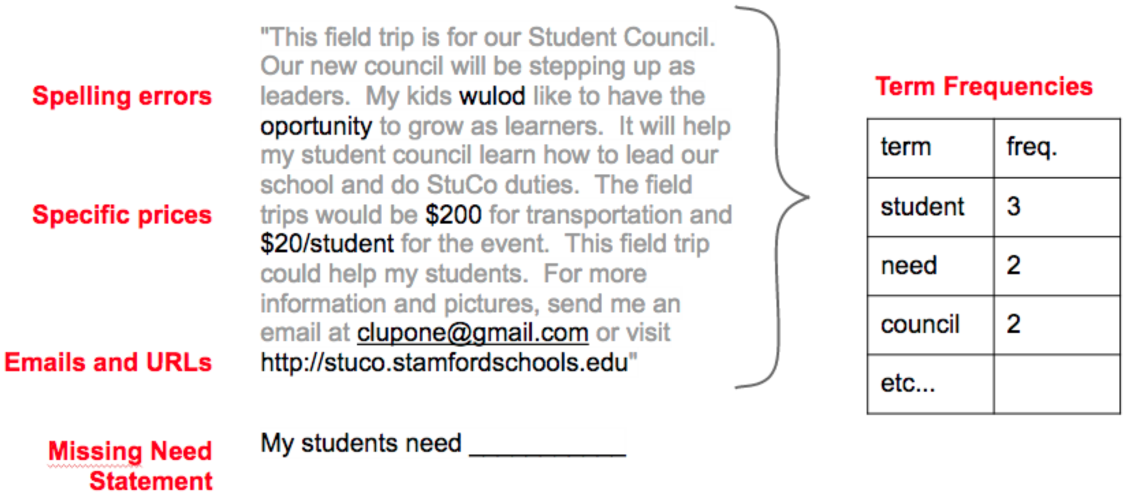
- TFIDF produces a sparse matrix that weights each word in the word lists. Words that appear frequently in a document are weighted higher. Words that appear frequently in all documents are weighted lower. The weights are normalized and smoothed by the Scikit-Learn package.
- TFIDF required use of sparse array formats. Another lesson learned: never use boolean indexing on a scipy sparse matrix.
- Predictive information could not be gleaned from terms that appeared only once across all documents; they were removed from consideration.

Additional features were engineered as follows:

- *Essay length (Numerical).* In general, essays that are descriptive and written more in detail are likely to be approved. The length of each essay is a good proxy for this criterion.
- *Missing fields (Binary).* Proposals are only approved if essay, need statement, and title are complete. Any project proposal with a missing field presents a strong signal for additional review and possible rejection.
- *All caps.* Essays that include a lot of "shouting" are generally undesirable. This can be an indicator of poorly written essays and possible spams.
- *Mentions of amount of money requested in essays.* Teachers are not allowed to manually put in the amount of money requested. This amount should be calculated automatically based on the materials requested. Both the count of and binary indication of "$" sign is used as an indicator of whether the essay contains any mention of amount of money requested.
- *Email addresses present in essays.* Teachers are not allowed to include email addresses in the essays. The count of "@" symbol is used as a proxy for emails present in the essays.

[2] http://tartarus.org/~martin/PorterStemmer/

- *URLs present in essays.* Website links are also not allowed in essays. We counted the number of times any of the following strings show up in essays. The list of strings we searched for included: “www”, “.com”, .”org”, and “.edu”.



"This field trip is for our Student Council. Our new council will be stepping up as leaders. My kids wulod like to have the oportunity to grow as learners. It will help my student council learn how to lead our school and do StuCo duties. The field trips would be $200 for transportation and $20/student for the event. This field trip could help my students. For more information and pictures, send me an email at clupone@gmail.com or visit http://stuco.stamfordschools.edu"

My students need ____________

term	freq.
student	3
need	2
council	2
etc...	

A simplified illustration of some features we would extract from essay data. We identified proxies for prices, emails, and urls and encoded into features, and tokenized essays into unigrams.

All numerical variables were normalized and smoothed using the Scikit-Learn Normalize package. Binary variables were not normalized.

We also considered several other features such as spell-checking and topic analysis using the gensim library[3]. Ultimately, we decided against using these features. Time and data constraints were the biggest factors limiting the features we engineered.

A summary table of the features and labels we explicitly used for modeling is provided in Appendix III.

Systematic vs. Random Sampling

To split the data into training and test sets, we set a pivot date such that all proposals created before that date would belong to the training set and the rest would belong to the test set. Proposals with null ‘created date’ values were thrown out (there were only a few of these). The advantage over a random split was twofold. First, the split more closely represents the actual production environment. Secondly, splitting on a date

[3] https://radimrehurek.com/gensim/

would allow us to generate time-based features. Although we did not include any time-based features in the model we built, we believe some features could be very powerful. For example, each teacher is associated with an average of 5 projects. This suggests that many teachers are repeat users of the site for hosting projects. It intuitively makes sense that teachers who have a track record of historical successes are likely to be approved in the future and teachers that are new to the site might encounter more problems with getting their projects approved.

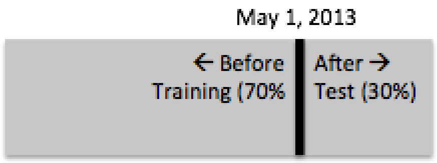
Our pivot date - May 1st, 2013 - was set to place 70% of *rejected* proposals into the training set and 30% into test. We split on rejected proposals rather than the whole dataset because the dataset is only 3% rejects - they were the limiting factor. Of the approved proposals before that date, we took a random subset so they would match the rejected proposals in number, and thereby improve modeling results.

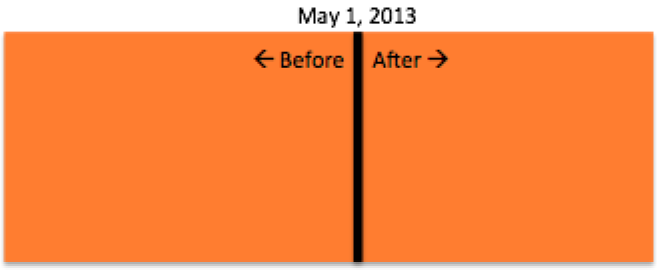
The process was as follows:

1. Split the data into approved and rejected subsets. The approved set is much larger.

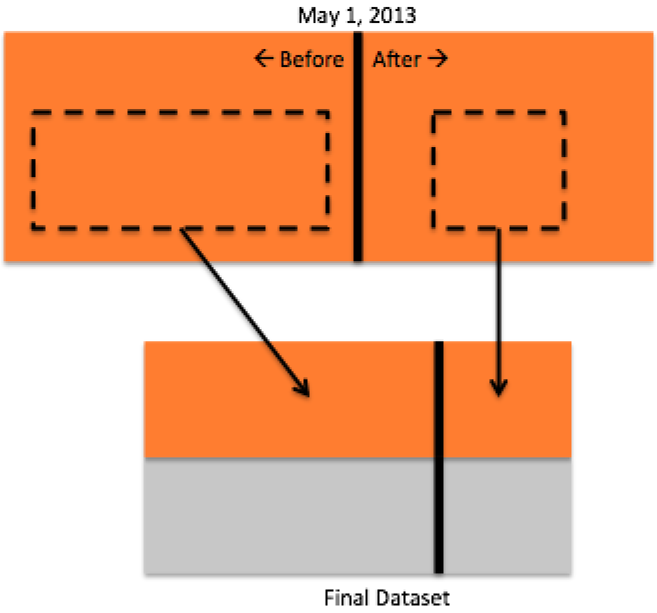
Approved
Rejected

2. Split the rejected data on a date so that 70% of cases are before that date, 30% are after. Split the approved data on the same date.





3. Downsample the approved data to match the rejected in number.



The resulting dataset is balanced for both training and test sets. Without this balancing, models would ignore feature data and universally approve proposals to achieve maximum 'accuracy' on the biased datasets.

Modeling

The auto-screener process aims to reduce the overall screening workload while satisfying Donors Choose content integrity requirements and ensuring fair screening. As a modeling problem, or goal is to build a predictor that would approve or flag project proposals accurately. This is a supervised binary classification problem.

We wanted to build a model that could be implemented, maintained, and understood easily by both technical and business stakeholders. It is not as imperative for the model to be particularly fast, though the less expensive in terms of memory and time the better. We decided to build three models for comparison: multinomial naïve bayes, logistic regression, and linear SVM. As is illustrated in the next section, logistic regression was the best solution out of the three models. Scikit-Learn was used for all classification modeling.

Naive Bayes:

- ➢ Our problem looks very similar to a spam classification problem. Thus we decided to include Naive bayes in our list of algorithms to test.
- ➢ Naive bayes is fast and scalable and handles sparsity well, which is very relevant to our problem.
- ➢ Naive bayes is not susceptible to the curse of dimensionality. The TFIDF feature is highly dimensional.
- ➢ It is relatively easy to interpret Naive Bayes model output.
- ➢ It only works well if the naive assumption holds. There is a possibility that this assumption does not hold very well with our data. Some teachers tend to post a lot of projects and others do not.
- ➢ Multinomial Naive Bayes was used.
- ➢ Grid Search Parameter: alpha, the Laplace smoothing parameter.
- ➢ Grid Search Range: 10^{-9} to 10^{2}

Logistic Regression:

- ➢ Logistic regression tends to do well in binary classification problems. Thus we added it to the mix of algorithms to test.
- ➢ Logistic regression is also fast, scalable, and suited to highly dimensional problems.
- ➢ We have naively assumed that nonlinearities are not present in our data, and thus are using a linear logistic regression model.
- ➢ Features need to be normalized prior to running a logistic regression model in order to better interpret results.
- ➢ A downside of the logistic regression model is that it can be difficult to compare regression coefficients of binary variables to those of numerical variables.
- ➢ L1 regularization was used in all logistic regression models to reduce the expected variance of our model and prevent overfitting. This is especially useful

given that we have extremely high dimensionality relative to the number of observations.

- Grid Search Parameter: C, the inverse regularization strength
- Grid Search Range: 10^{-9} to 10^{2}

SGD Linear SVM:

- The linear SVM is also suited to binary classification problems. We used the SGD Classifier package in Scikit-Learn to perform this modeling.
- SVM is highly suitable to nonlinear decision boundaries, though we did not use this feature of SVM.
- We used Hinge Loss.
- L1 regularization was used to reduce the expected variance of our model and prevent overfitting. This is especially useful given that we have extremely high dimensionality relative to the number of observations.
- Grid Search Parameter: alpha, the regularization strength
- Grid Search Range: 10^{-14} to 10^{1}

Evaluation & Insights

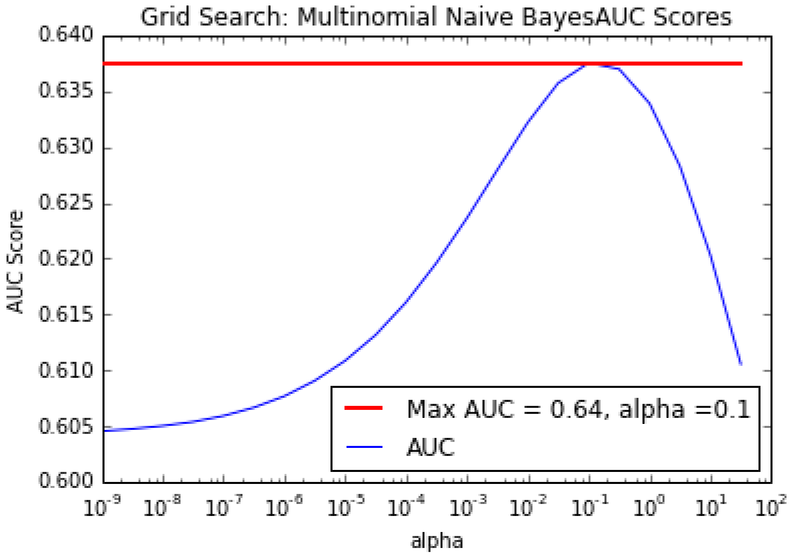
The goal of the auto-screener process is to reduce the overall screening workload while satisfying Donors Choose content integrity requirements and ensuring fair screening. Thus, an appropriate evaluation metric for model selection needs to incorporate a balance of minimizing the false positive rate while maximizing the true positive rate. In a production environment, a pre-determined decision threshold would be used to strike this balance. In our analysis, no such decision threshold has been determined. We leave it as an outstanding business question.

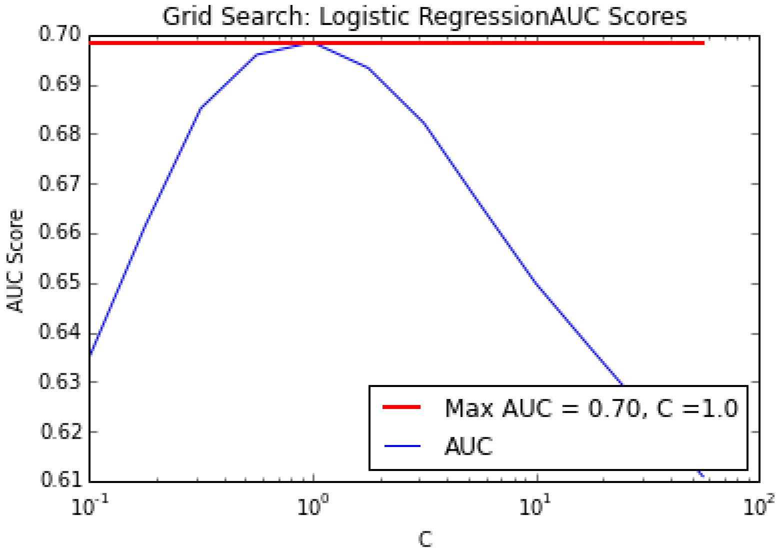
Therefore, we use AUC, which is a representation of model performance across all decision thresholds.

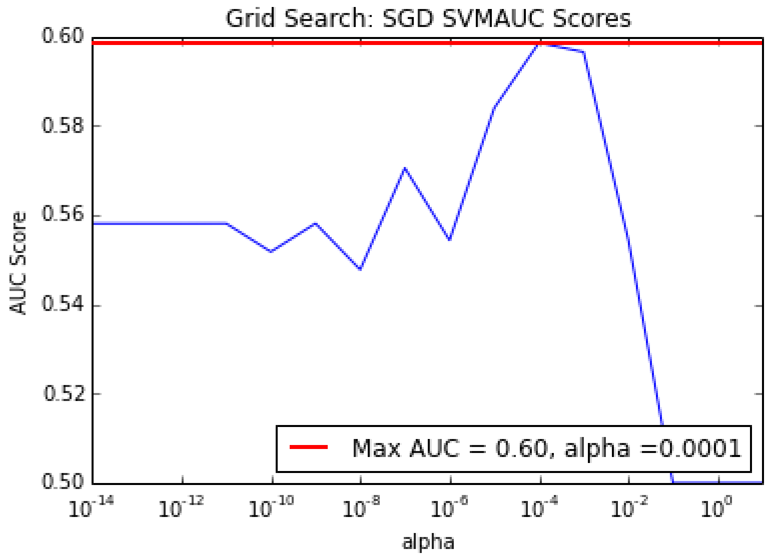
To choose the best model, we performed a parameter grid search on the Multinomial Naive Bayes, Logistic Regression, and SGD SVM classifiers. The results of the grid search are provided in the following three diagrams. Logistic regression clearly performed the best of the three alternatives.

- The naive bayes grid search showed an optimal AUC of 0.64 at alpha = 0.1
- The logistic regression grid search showed an optimal AUC of 0.7 at C = 1.0

- The SGD linear svm grid showed an optimal AUC of 0.6 at alpha = 0.0001







To understand what the ROC curve means from a business perspective, we looked at various approval thresholds and the percent of false approval rates associated with each threshold.

Good Projects Approval Rate	Bad Projects Approval Rate	Anticipated Production Results Bad Postings Rate (% Posted Projects That Are Bad)
5%	1%	0.18%
15%	5%	0.90%
25%	10%	1.80%

Given a total number of 5,600 rejections in the test set, a 1% false approval rate means 56 projects that should have been rejected were posted on the site. If this were projected to a production system of 100,000 proposals in which 18% are originally flawed, then at the 1% false approval rate:

Auto-screener performance

- 180 bad projects would be approved
- 5,100 good projects would be approved

Overall website content integrity

- Assume perfect judgment by the human reviewers
- Total 82,180 approved projects
- 0.18% of projects posted should not have been

It is challenging to tell what the actual impact of choosing each threshold would bring. The dataset that we received is heavily skewed towards approvals since only the final version of any proposal is saved in the database. Ultimately, setting the appropriate threshold is a business decision. Setting the threshold at the 25% approval rate would save 10,000 hours, but at the cost of 1.8% bad postings rate.

In a real production system, we recommend that both the approval rate of the auto screener and an estimated bad postings rate of the auto-screener would need to be monitored. Auto-approval-rate would measure workload reduction. Bad postings rate measures website content integrity. Since the bad postings rate could not be monitored directly, it would need to be inferred by sending random samples of the auto-approved proposals directly to human reviewers instead of onto the website and tracking the bad postings rate.

A natural question that arises from this analysis is: how bad are those 1.8%?   In addition to using quantitative metrics for evaluating our model, we examined rejected essays that scored 1%, 3%, 5%, and 15% probability of approval using the model. Overall, the ranking makes sense intuitively.

The essays that the model predicted with 1% approval probability contained short nonsensical phrases on repeat. It is safe to assume these are spam proposals that should be rejected. The 1% probability corresponds to this intuitively.

The rejected essays that the model predicted with 5% approval probability are already well written with only minor grammatical errors. In the example below, it is pretty hard to tell whether the essay would have been approved or not. Often, the rejected essays with higher approval probabilities tend to have very subtle faults that would be difficult for a computer algorithm to pick up. These included minor grammatical errors, thematic issues, and vague descriptions.

Sample Essay with 1% approval probability

My students need My students need My students need My students need My students need My students need My students need My students need My students need My students need My students need My students need My students need My students need .... [and so on for 20 lines]

Sample Essay with 5% approval probability

Experiences are what shape our future. As a teacher, I wish to improve and enhance the lives of my students as much as possible. Next year, I will take 12 students on a once in a lifetime trip to Spain. These students have been studying Spanish for 2-3 years but have never studied abroad. \r\\n\r\\nMost of my students have never been out of their

> state/country. These students are hardworking, dedicated, and brilliant in every way possible. They come from excellent families and have a thirst for knowledge and will thrive on the ability to experience and share that knowledge with their friends and families. They can\xc3\x83\xc2\xa2\xc3\x82\xc2\x80\xc3\x82\xc2\x99t stop dreaming about learning Spanish in Spain.can \r\\n\r\\nThe experiences that they will be exposed to while traveling will serve as life-long lessons and will, undoubtedly, change their perspective on themselves and their place in the world. In Spain, we will visit Madrid and Barcelona. We will visit the Prado, learn to make paella, visit the Royal Palace and many more amazing sites. Students will take language classes. This will be a trip that they will never forget! I am requesting assistance with partial funding of their trip. Parents are contributing to the cost of the trip, and in addition, we would like to help each student with the cost through Donors Choose.org. This will help alleviate some of the costs to the students and their parents. We are working hard to fund raise throughout the community for the remaining balance of the students\' trip by having car washes, auctions, yard sales, and Spanish food nights.   \r\\n\r\\nIn Spain, our students will witness new cultures first hand and experience history typically only learned in a textbook. While students will remember some material just long enough to succeed on a test, the memories of this trip will remain with them forever. Thank you for taking the time to read about our trip. We are working hard to fund raise. Any help is appreciated. Thank you for making a difference!

Since the review criteria included some specific rules, we also examined approval and rejection statistics based on these rules. We looked at projects that included website URLs, email addresses, and specific prices. The statistics show that proposals including these elements are positively correlated with rejections, though these are not rules that guarantee rejection. We believe the correct way to interpret this is that projects containing these elements should be flagged for manual review instead of returned to the teachers. We would also like to point out that even though these are indicative criteria to check for, they account for only a very small percentage of all projects. In total, they represent roughly 0.022% of all rejected projects in the dataset we have. This, however, is not representative of what the proportions are in reality. As we have mentioned before, the dataset we are using only contains final revisions, which means there could be many more projects that initially contained urls, emails, and prices. Regretfully, we do not have sufficient information to estimate the actual number.

Overall, the model we have built represents our best attempt at creating a supervised classification model. Our evaluation shows that the model makes sense intuitively and has a decent performance. It has yet to be determined the exact business value this would bring. The analysis of the business case depends largely on access to a representative dataset of the production environment as well as anticipated costs for making changes to the production environment. Solely based on the data sources we have, we can help Donors Choose save roughly 3,000 hours of volunteer work per year with a cost of roughly 5% false approvals and 0.9% overall bad postings rate. On this note, we acknowledge there is still a lot of room for improvement and much more in-depth analysis is required. We believe what we have learned here will prove to be valuable for any subsequent undertaking of the project.

Steps for Improving the Model

Although the model provides decent predictive accuracy, there is ample room for improvement. Given the data challenges mentioned before, we were, unfortunately, unable to try several things that we had reason to believe would have improved the model. These are as follows:

Features

- Use a spell-check package like pyEnchant[4] to count the number of misspelled words.
- Teachers are told that their need statement should match the list of resources requested. We could measure the overlap between words in the need statement and resources requested.
- The number of project proposals approved or rejected in the past, for a given teacher or school, could be predictive.
- Lemmatization, rather than simple stemming, to reduce dimensionality of essays. Lemmatization uses context to appropriately collapse word variants into one root word for analysis.
- Topic Modeling such as LDA to convert essays into lists of topics. The topics may have been more meaningful features than the lists of words themselves.

Other Model Selection Strategies

- We implemented L1 regularization for feature selection, but could try models with more/fewer features to better understand the significance of each.
- Bootstrapping the data to a larger set on which to run our models, to see if we are using enough data.

Even with these improvements, our model would still be limited by the quality of the data. Without access to the revision history of proposal essays, the model is denied a wealth of information about what causes a proposal to be rejected. If costs allow, we recommend that Donors Choose keep a record of the entire proposal revisoin history, even if they are later approved after revision.

[4] http://pythonhosted.org/pyenchant/

Recommendations for Deployment

Before additional modeling and testing is performed, we recommend that some key questions should be discussed and answered.

- Decision Threshold: Assuming the performance results developed in this analysis are representative of a production system, what is an appropriate decision threshold? What bad postings rate would be acceptable given the projected savings in volunteer effort?
- Initial and long-term implementation costs: Modifications to production environments tend to be expensive and disruptive. Therefore, determine order of magnitude cost, schedule, and effort estimates for auto-screening modifications to the production environment. This includes both labor, hardware, and software costs for initial deployment as well as long term operations and maintenance. Would the costs justify the anticipated savings?
- Coordinate with other internal initiatives: It may be that Donors Choose has other anticipated modifications to the production environment. How could the auto-screener project be rolled into those? Are there potential conflicts?

These questions should be answered prior to additional model testing and development is performed. Should Donors Choose decide that the auto-screening initiative would be worthwhile, the next step is to get better data and continue to build the model. For this next testing and development phase, we propose the following:

- PROD scraping: Since it is desirable to avoid making changes to the production system, we recommend scraping the production database on a frequent basis to capture proposal revision history. The database of scraped data need not be maintained to production system standards, such as 24/7 reliability, and thus we anticipate that there are cost-effective solutions to achieve this. Appropriate syncing and logic routines would need to be put in place to make sure that the scraper is grabbing the 1st revision after a teacher has submitted and prior to subsequent revisions.
- In-parallel testing: With the scraper in place, a framework for parallel testing should be put in place. This means testing the auto-screener on incoming scraped data and assessing its performance against the AUC, auto-approval-rate, and bad-postings-rate metrics.
- In-parallel model development: The incoming scraped data should be used to build better models. The additional feature engineering identified in the section

"Steps for Improving the Model" should also be explored for incorporation into the model.

- Revision Loop Considerations: Once a proposal has been flagged for revision, should subsequent revisions again go through the auto-screener or directly to a human reviewer. This is a question that we were unable to answer in our analysis, given the aforementioned data challenges. It is a question that should be addressed during the in-parallel model development.

Detailed production planning need not be considered until after in-parallel model testing and development has been completed. It is worth pointing out now how the performance of the production auto-screener would be evaluated on an ongoing basis.

- Metrics: The production system would need to measure auto approval rate and bad postings rate. Auto approval rate is simple to calculate, since it is simply the percentage of proposals approved by the auto screener.
- Measuring bad postings rate: The bad postings rate would need to be inferred through statistical measurement. Random samples of the auto-approved proposals could be sent to human screeners instead of to the website. The approve/reject decision by the human screener would be tracked. The sampled bad postings rate would be the proxy metric.

There are important ethical considerations that Donors Choose should note. Fair treatment of proposals is important. Currently, Donors Choose achieves fair treatment by limiting what human screeners see. In addition, all proposals take 2-4 weeks to review. This dynamic changes with an auto-screener.

- Change in review time: With an auto-screener, the review time changes from either immediate to 2-4 weeks. Auto-approved proposals are reviewed immediately. Flagged reviews require human processing time. This introduces a potential for unfair treatment of some groups of proposals.
- Limiting auto-screener features: We recommend that the features used by the auto-screener be limited to only information that is seen by the human reviewers. This would exclude the project metadata.
- Monitor auto-screener decisions: The types of proposals that are flagged for review should be reviewed by an analytical team independent of the human screeners. This team should look at the proposal project metadata to see if, for example, certain subjects or classrooms from certain poverty levels are treated differently by the auto-screener and try to understand why this is happening.

The question of whether or not the auto-screener is making the right decision on ethical grounds must be treated on a case by case basis.

It is often the case in problem solving that we go through complicated analyses to come to a realization that the best solutions are the simple ones. Throughout our analysis we identified some non-data-mining solutions to improve the screening process.

- Auto-verification: Our analysis suggests that a number of projects that included external links, email addresses, and currency symbols were rejected. Even without deploying our full model, Donors Choose could reduce screener workload by adding an automatic verification step to the proposal submission process. If the system detects use of email addresses, external links, or currency symbols in a draft proposal, teachers should be warned and reminded of the policies before submission.
- Flag button: Donors Choose could include a “flagging” button on published projects to allow users of the site to flag projects that they believe might be problematic. Even human screeners are not perfect, and this button might help improve the integrity of the website, regardless of whether or not an auto-screener is put in place.

Although there are still many questions to be answered and the model is not yet ready for deployment, we garnered valuable insights that calls for immediate actions and helps the organization better understand its business and processes. Data mining is never a straightforward path to an answer. It is always a process of constant learning and improvements. We believe we have done just that in this project. We look forward to future developments along this initiative.

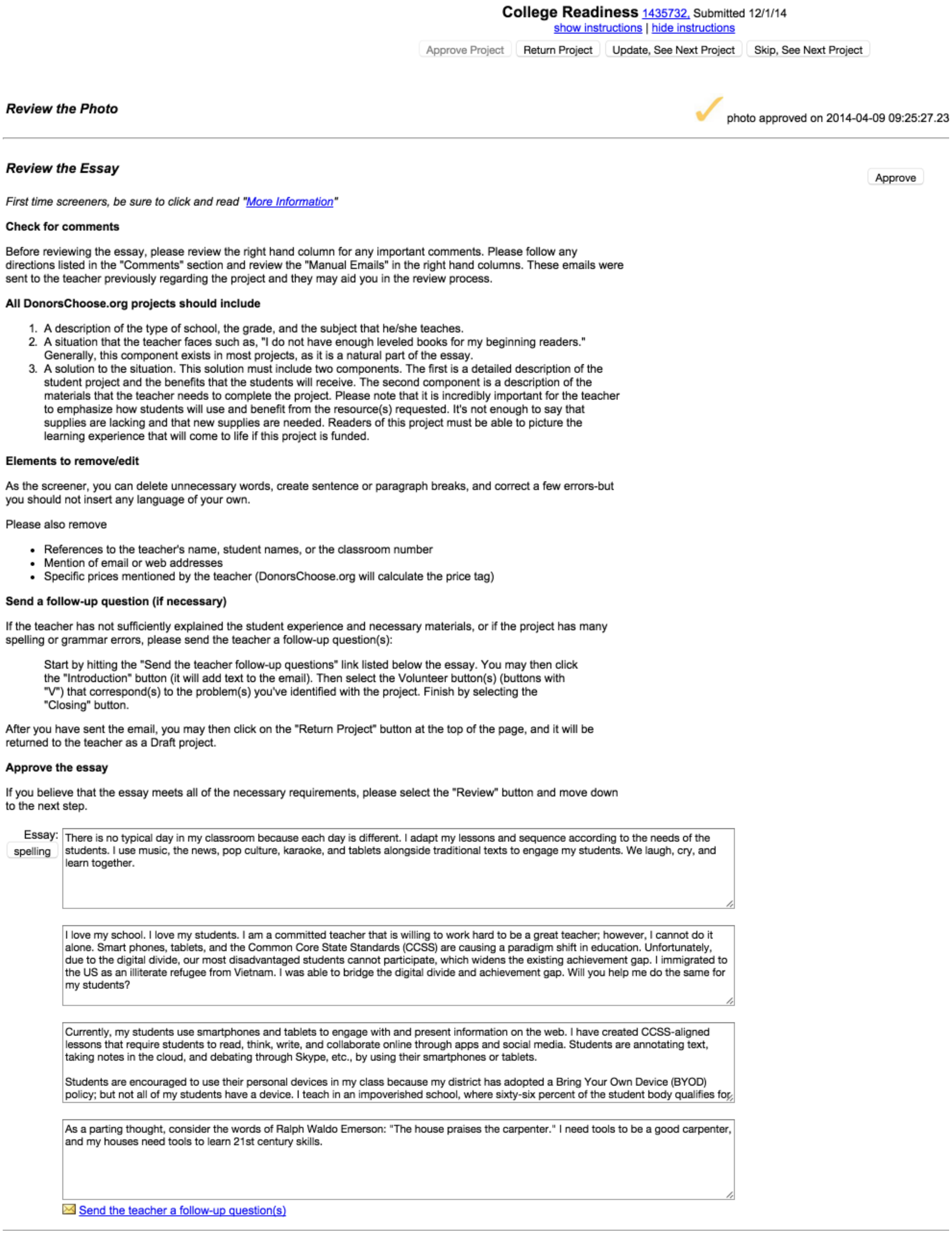
Acknowledgments

We would really like to thank Vladimir Dubovskiy, the Head of Data Science at Donors Choose, for working with us throughout the project. Vlad was kind enough to suggest the idea, offer his time to get the data we needed, and meet with us in person to discuss project specifics. We would also like to thank Professor D’Alessandro for a great semester and challenging us to tackle this project.

References

1. All coding was performed in Python 2.7. https://www.python.org/
2. The Python scientific modeling distribution SciPy was used extensively for data wrangling, feature engineering, modeling, and analysis. http://www.scipy.org/
3. The Natural Language Toolkit NLTK was used for text processing. http://www.nltk.org/
4. The NYU DSGA-1001 Introduction to Data Science course lecture slides by Brian D'Alessandro were used for understanding of model selection and model results interpretation.
5. All information about Donors Choose derived from either DonorsChoose.org or from a direct communication from DonorsChoose.

Appendix I Sample Volunteer Review Screenshot



College Readiness 1435732, Submitted 12/1/14

Review the Photo

✓ photo approved on 2014-04-09 09:25:27.23

Review the Essay

First time screeners, be sure to click and read "More Information"

Check for comments

Before reviewing the essay, please review the right hand column for any important comments. Please follow any directions listed in the "Comments" section and review the "Manual Emails" in the right hand columns. These emails were sent to the teacher previously regarding the project and they may aid you in the review process.

All DonorsChoose.org projects should include

1. A description of the type of school, the grade, and the subject that he/she teaches.
2. A situation that the teacher faces such as, "I do not have enough leveled books for my beginning readers." Generally, this component exists in most projects, as it is a natural part of the essay.
3. A solution to the situation. This solution must include two components. The first is a detailed description of the student project and the benefits that the students will receive. The second component is a description of the materials that the teacher needs to complete the project. Please note that it is incredibly important for the teacher to emphasize how students will use and benefit from the resource(s) requested. It's not enough to say that supplies are lacking and that new supplies are needed. Readers of this project must be able to picture the learning experience that will come to life if this project is funded.

Elements to remove/edit

As the screener, you can delete unnecessary words, create sentence or paragraph breaks, and correct a few errors-but you should not insert any language of your own.

Please also remove

- References to the teacher's name, student names, or the classroom number
- Mention of email or web addresses
- Specific prices mentioned by the teacher (DonorsChoose.org will calculate the price tag)

Send a follow-up question (if necessary)

If the teacher has not sufficiently explained the student experience and necessary materials, or if the project has many spelling or grammar errors, please send the teacher a follow-up question(s):

> Start by hitting the "Send the teacher follow-up questions" link listed below the essay. You may then click the "Introduction" button (it will add text to the email). Then select the Volunteer button(s) (buttons with "V") that correspond(s) to the problem(s) you've identified with the project. Finish by selecting the "Closing" button.

After you have sent the email, you may then click on the "Return Project" button at the top of the page, and it will be returned to the teacher as a Draft project.

Approve the essay

If you believe that the essay meets all of the necessary requirements, please select the "Review" button and move down to the next step.

Essay:

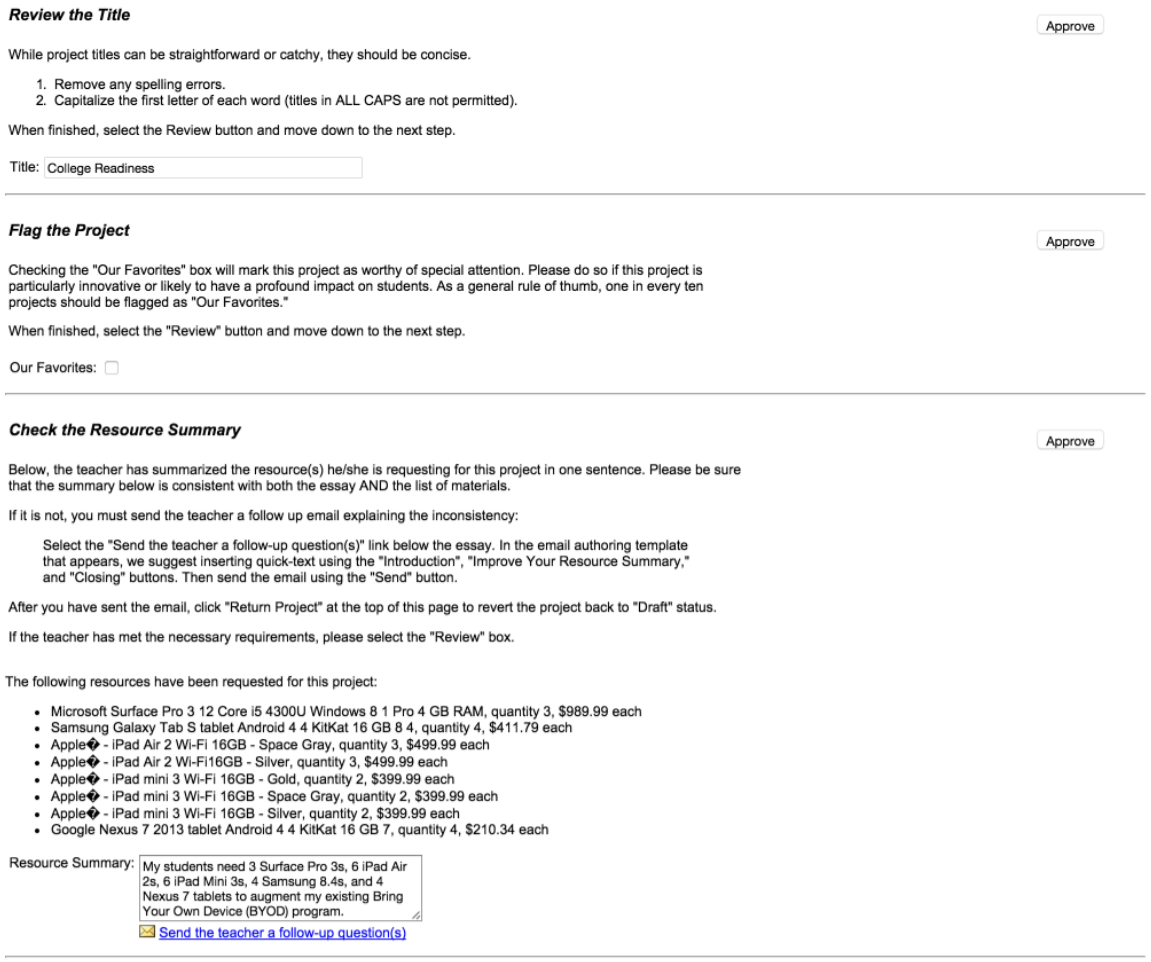
There is no typical day in my classroom because each day is different. I adapt my lessons and sequence according to the needs of the students. I use music, the news, pop culture, karaoke, and tablets alongside traditional texts to engage my students. We laugh, cry, and learn together.

I love my school. I love my students. I am a committed teacher that is willing to work hard to be a great teacher; however, I cannot do it alone. Smart phones, tablets, and the Common Core State Standards (CCSS) are causing a paradigm shift in education. Unfortunately, due to the digital divide, our most disadvantaged students cannot participate, which widens the existing achievement gap. I immigrated to the US as an illiterate refugee from Vietnam. I was able to bridge the digital divide and achievement gap. Will you help me do the same for my students?

Currently, my students use smartphones and tablets to engage with and present information on the web. I have created CCSS-aligned lessons that require students to read, think, write, and collaborate online through apps and social media. Students are annotating text, taking notes in the cloud, and debating through Skype, etc., by using their smartphones or tablets.

Students are encouraged to use their personal devices in my class because my district has adopted a Bring Your Own Device (BYOD) policy; but not all of my students have a device. I teach in an impoverished school, where sixty-six percent of the student body qualifies for

As a parting thought, consider the words of Ralph Waldo Emerson: "The house praises the carpenter." I need tools to be a good carpenter, and my houses need tools to learn 21st century skills.



Review the Title

While project titles can be straightforward or catchy, they should be concise.

1. Remove any spelling errors.
2. Capitalize the first letter of each word (titles in ALL CAPS are not permitted).

When finished, select the Review button and move down to the next step.

Title: College Readiness

Flag the Project

Checking the "Our Favorites" box will mark this project as worthy of special attention. Please do so if this project is particularly innovative or likely to have a profound impact on students. As a general rule of thumb, one in every ten projects should be flagged as "Our Favorites."

When finished, select the "Review" button and move down to the next step.

Our Favorites: □

Check the Resource Summary

Below, the teacher has summarized the resource(s) he/she is requesting for this project in one sentence. Please be sure that the summary below is consistent with both the essay AND the list of materials.

If it is not, you must send the teacher a follow up email explaining the inconsistency:

> Select the "Send the teacher a follow-up question(s)" link below the essay. In the email authoring template that appears, we suggest inserting quick-text using the "Introduction", "Improve Your Resource Summary," and "Closing" buttons. Then send the email using the "Send" button.

After you have sent the email, click "Return Project" at the top of this page to revert the project back to "Draft" status.

If the teacher has met the necessary requirements, please select the "Review" box.

The following resources have been requested for this project:

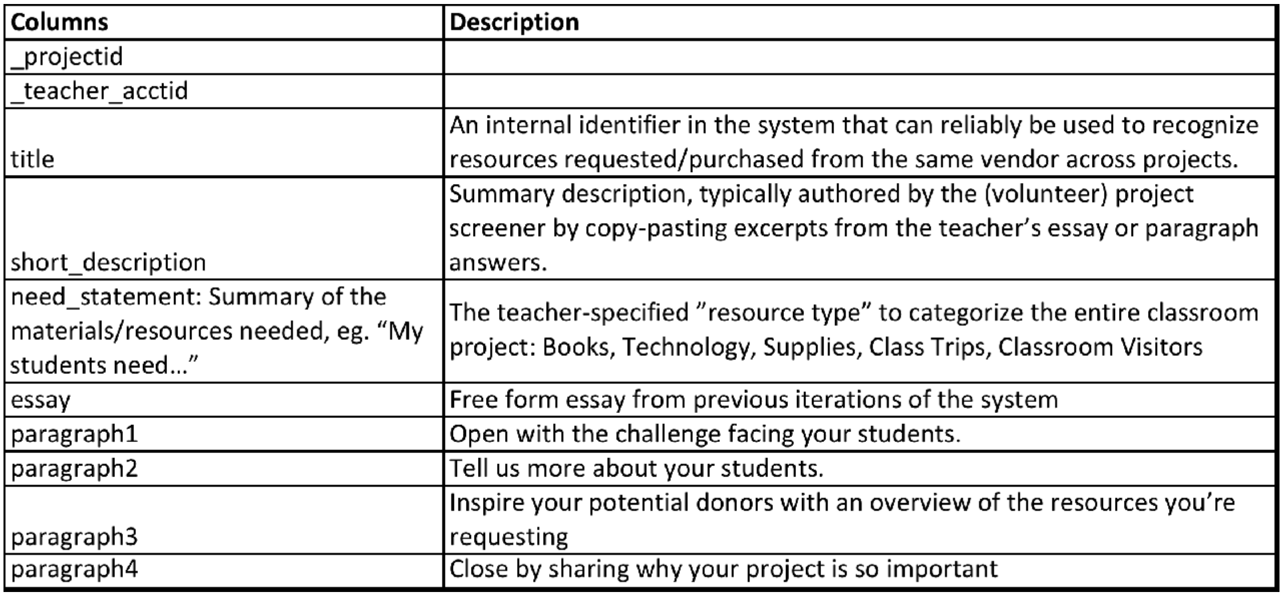
- Microsoft Surface Pro 3 12 Core i5 4300U Windows 8 1 Pro 4 GB RAM, quantity 3, $989.99 each
- Samsung Galaxy Tab S tablet Android 4 4 KitKat 16 GB 8 4, quantity 4, $411.79 each
- Apple� - iPad Air 2 Wi-Fi 16GB - Space Gray, quantity 3, $499.99 each
- Apple� - iPad Air 2 Wi-Fi16GB - Silver, quantity 3, $499.99 each
- Apple� - iPad mini 3 Wi-Fi 16GB - Gold, quantity 2, $399.99 each
- Apple� - iPad mini 3 Wi-Fi 16GB - Space Gray, quantity 2, $399.99 each
- Apple� - iPad mini 3 Wi-Fi 16GB - Silver, quantity 2, $399.99 each
- Google Nexus 7 2013 tablet Android 4 4 KitKat 16 GB 7, quantity 4, $210.34 each

Resource Summary: My students need 3 Surface Pro 3s, 6 iPad Air 2s, 6 iPad Mini 3s, 4 Samsung 8.4s, and 4 Nexus 7 tablets to augment my existing Bring Your Own Device (BYOD) program.

Send the teacher a follow-up question(s)

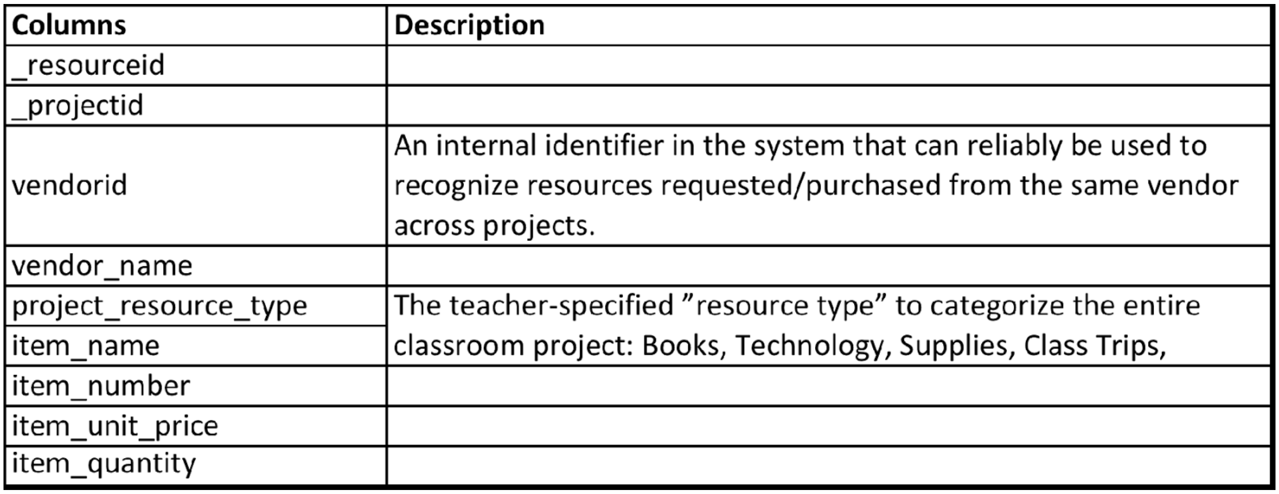
Appendix II Dataset Documentation

Project Essay Data



Columns	Description
_projectid	
_teacher_acctid	
title	An internal identifier in the system that can reliably be used to recognize resources requested/purchased from the same vendor across projects.
short_description	Summary description, typically authored by the (volunteer) project screener by copy-pasting excerpts from the teacher’s essay or paragraph answers.
need_statement: Summary of the materials/resources needed, eg. “My students need…”	The teacher-specified ”resource type” to categorize the entire classroom project: Books, Technology, Supplies, Class Trips, Classroom Visitors
essay	Free form essay from previous iterations of the system
paragraph1	Open with the challenge facing your students.
paragraph2	Tell us more about your students.
paragraph3	Inspire your potential donors with an overview of the resources you’re requesting
paragraph4	Close by sharing why your project is so important

Project Resource Data



Columns	Description
_resourceid	
_projectid	
vendorid	An internal identifier in the system that can reliably be used to recognize resources requested/purchased from the same vendor across projects.
vendor_name	
project_resource_type	The teacher-specified ”resource type” to categorize the entire classroom project: Books, Technology, Supplies, Class Trips,
item_name	
item_number	
item_unit_price	
item_quantity	

Project Data

Type	Feature Name	Description
Binary	school_charter	They are flagged with 1 if the project is associated with the type of school specified, and a 0 if not.
	school_magnet	
	school_year_round	
	school_nlns	
	school_kipp	
	school_charter_ready_promise	
	teacher_prefix	Dr., Mr., Mr. & Mrs., Mrs., Ms. These are transformed into binary data.
	teacher_teach_for_america	Teach For America program (0 = no, 1 =yes)
	teacher_ny_teaching_fellow	Teaching Fellows program (0 = no, 1 =yes)
	primary_focus_subject	Subject (e.g. Literacy, Music, Math...). These are transformed into binary data.
	primary_focus_area	Area of study (e.g. Literacy & Language...). These are transformed into binary data.
	secondary_focus_subject	Secondary subject (similar to primary)
	secondary_focus_area	Secondary area of study (similar to primary)
	resource_type	Books, Technology, Supplies, Trips, Visitors, others. These are transformed into binary data.
	poverty_level	high, low, minimal, unknown. These are transformed into binary data.
		Grades PreK-2, Grades 3-5, Grades 6-8, Grades 9-12. These are transformed into binary data.
	grade_level	
	school_metro_mv	Missing values. They are flagged with 1 if the value is missing, and a 0 if not.
	primary_focus_subject_mv	
	primary_focus_area_mv	
	secondary_focus_subject_mv	
	secondary_focus_area_mv	
	resource_type_mv	
	grade_level_mv	
	need_statement_mv	
	short_description_mv	
	email	This is flagged with 1 if the essay contains email address, and a 0 if not.
	urls	This is flagged with 1 if the essay contains URL, and a 0 if not.
Numerical	vendor_shipping_charges	These fields are available under Project Details on a project page, breaking down the cost of a project into its parts.
	sales_tax	These fields are available under Project Details on a project page, breaking down the cost of a project into its parts.
	payment_processing_charges	
	total_price_excluding_optional_support	
	total_price_including_optional_support	
	essay_len	Number of words in an essay
	totalcaps	Total number of capital letters
	maxcaps	Number of consecutive capital letters
	dollarcount	Number of '$' sign in an essay
	essay	TF-IDF Matrix generated by Scikit-learn TFIDF Vectorizer.
Label	rejected	0 = approved, 1 = rejected

Appendix III Dataset Documentation

Type	Feature Name	Description
Label	approved	1 = approved, 0 = rejected
Binary	need_statement_mv	Missing values. They are flagged with 1 if the value is missing, and a 0 if not.
	short_description_mv	
	essay_mv	
	dollar_bool	This is flagged with 1 if the essay contains “$”, and a 0 if not.
	email_bool	This is flagged with 1 if the essay contains email address, and a 0 if not.
	urls_bool	This is flagged with 1 if the essay contains URL, and a 0 if not.
Numerical (Normalized)	essay_len	Number of words in an essay
	totalcaps	Total number of capital letters
	maxcaps	Largest string of consecutive capital letters (spaces and symbols ignored)
	dollarcount	Number of times '$' appears in essay
	TFIDF – essay	TF-IDF Matrix generated by Scikit-learn TFIDF Vectorizer.

Appendix IV Team Member Contributions

Charlie Guthrie

- Lead on communications with Donors Choose
- Data Wrangling
- Train/Test Splitting & Down Sampling
- Model Results Interpretation
- Feature Engineering and Visualization
- Presentation
- Write-Up

Justin Mao-Jones

- Data Wrangling
- Modeling
- Model Results Analysis
- Feature Engineering and Visualization
- Presentation
- Write-Up
- Code Organization

Yasumasa Miyamoto

- NLP Feature Engineering (TFIDF)
- Exploratory Feature Engineering
- Model Results Interpretation
- Visualizations for Presentation
- Visualizations for Write-Up
- Tables for Write-Up

Lucy Wang

- Feature Engineering (Stemming & Lemmatizing)
- Data and document organization
- Model Results Interpretation
- Presentation Development
- Write-Up