ISyE 6740 - Fall 2021
Final Project Report

Team Member Name: Chai Han Xie (hchai30)

Project Title: Ranking Projects for a Donation Website

Problem Statement

DonorsChoose.org is an online website that helps to link donors with projects to help students in need. These projects are submitted by teachers, alongside information such as the need statement and the materials needed. When a project reaches its funding goal, DonorsChoose.org will ship the requested materials to the school.

Over the years, the number of projects submitted to the website have been rising steady (see Figure 1), making it increasingly difficult for donors to sift through all the projects to find something that they want to support. Even with a filter function, donors might have to look through a long list for popular subjects or areas, and hence could be discouraged from donating.

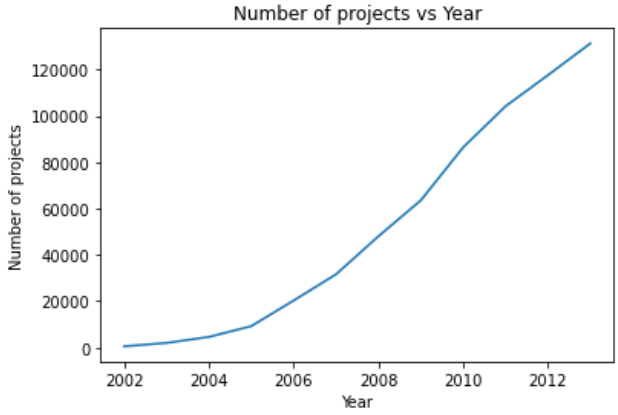

Figure 1: Number of projects on DonorsChoose.org have been increasing over the years.

It is therefore imperative to develop a model to help donors find projects that are potentially of exceptional quality more easily on the website. This would hopefully improve the number of donations, the number of successfully funded projects, and most importantly the number of students that will benefit from the donations.

To help with this, DonorsChoose.org has identified a subset of projects that they term as *exciting*. These projects has met the following requirements selected by DonorsChoose.org.

- $fully\_funded$: The project was fully funded.

- *at_least_1_teacher_referred_donor*: At least one donor donated to the project because of a shared link by the teacher. This type of donors will be referred to as teacher-referred donors.
- *great_chat*: The comment thread of the project had a higher than average percentage of donors leaving an original message.
- *at_least_1_green_donation*: The project received a donation via credit card, PayPal, Amazon, or check.
- *one_or_more*: In addition to every requirement above, the project also met one or more of the following requirements:
  - *three_or_more_non_teacher_referred_donors*: The project received donations from three or more teacher-referred donors.
  - *one_non_teacher_referred_donor_giving_100_plus*: A non-teacher-referred donor donated more than $100.
  - *donation_from_thoughtful_donor*: The project received a donation from a list of around 15 donors. The list was curated by DonorsChoose.org.

Data Source

There are three main dataset, each containing different information about a project.

essays.csv contains the essays and text provided by the teacher when submitting a project onto DonorsChoose.org. The data fields in **essays.csv** can be found in Table 1.

projects.csv contains most of the information about the project (e.g. resource type requested, total price of project), the teacher (e.g. prefix, qualifications), and the school that the teacher is from (e.g. city school is in, poverty level of school). The data fields in **projects.csv** can be found in Table 2.

outcomes.csv contains the requirements that DonorsChoose.org specified for exciting projects, as mentioned in the Problem Statement. This includes a *is_exciting* feature that aggregates the requirements.

The three dataset can be obtained from Kaggle (*https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose/data*).

Data field	**Description**
projectid	Unique id for each project
teacher_acctid	Unique id for the teacher who created the project
title	Text field for the title of the project
short_description	Text field for a short description of the project
need_statement	Text field for the need statement of the project
essay	Text field for an essay about the project

Table 1: Data fields in **essays.csv**.

Data field	Description
projectid	Unique id for each project
teacher_acctid	Unique id for the teacher who created the project
schoolid	Unique id for the school where the teacher is from
school_latitude	Latitude of school
school_longitude	Longitude of school
school_city	City of school
school_state	State of school
school_zip	Zip of school
school_metro	Metro of school
school_district	District of school
school_county	County of school
school_charter	Whether school is a public charter school
school_magnet	Whether school is a public magnet school
school_year_round	Whether school is a public year round school
school_nlns	Whether school is a public nlns school
school_kipp	Whether school is a public kipp school
school_charter_ready_promise	Whether school is a public ready promise school
teacher_prefix	Prefix of teacher
teacher_teach_for_america	Whether teacher is from Teach for America
teacher_ny_teaching_fellow	Whether teacher is from New York City Teaching Fellows
primary_focus_subject	Main subject for the project
primary_focus_area	Main area for the project
secondary_focus_subject	Secondary subject for the project
secondary_focus_area	Secondary area for the project
resource_type	Main type of resource requested by the project
poverty_level	Poverty level of school
grade_level	Grade level for which project materials are intended
fulfillment_labor_materials	Cost of fulfilment
total_price_excluding_optional_support	Project cost excluding optional tips by donors
total_price_including_optional_support	Project cost including optional tips by donors
students_reached	Number of students benefiting from the project
eligible_double_your_impact_match	Whether a corporate partner is supporting half of the project cost
eligible_almost_home_match	Whether a corporate partner is sponsoring US$100
date_posted	Date the project was submitted

Table 2: Data fields in **projects.csv**.

Methodology

We will focus on the **projects.csv** dataset as input data, given that the data is structured and there is a notable number (24) of features available. In addition, no projects submitted before 2010 is labelled by DonorsChoose.org as exciting. As such, we shall omit all projects before 2010, which still leaves us with a sizable amount of 484,371 projects (from 664,098 projects).

The remaining sections in this report will be as follows:

- **Feature selection and engineering**: We will identify important features that could be useful in predicting if a project is exciting, and reduce the number of features to a more manageable size. We will also explore whether it is possible to obtain more useful features by combining features, extracting information from existing features (e.g. month from the date posted), or creating new features (e.g. the number of past projects from the same teacher that have been labelled as exciting).
- **Model fitting and evaluation**: After we have identified useful features, we will use these features as inputs to classification models that we have learnt in this course, such as logistic regression, random forest classifiers, and neural networks. We will then compare these models based on a set of evaluation metrics.
- **Potential improvement**: Rather than fitting a single model to predict *is_exciting*, we can create multiple sub-models to predict each of the requirements instead. The output from each model could then be combined to form an overall prediction of whether the project is exciting.

Feature Selection and Engineering

To have a sense of the original features' usefulness, the mutual information (MI) of each feature with the label *is_exciting* is computed and shown in Figure 1.

MI of each feature X_i with the label Y $I(X_i, Y)$ quantifies the amount of information obtained about Y or the reduction in uncertainty of Y by observing X_i. Unlike measures such as correlation coefficient, MI is not limited to linear dependence, and can be computed from the entropy of Y $(H(Y))$ and the conditional entropy of Y given X_i $(H(Y|X_i))$.

$$I(X_i, Y) = H(Y) - H(Y|X_i)$$

Figure 2 shows that most of the original variables provide little information about the label, with only 6 variables having a MI of above 0.05. Among the variables, the cost of fulfilment, which measures the project's price excluding any portion of the donation that goes to supporting DonorsChoose.org, provides the highest amount of MI. This is followed by features about the school's location, such as latitude/longitude, district, city, county, and state.

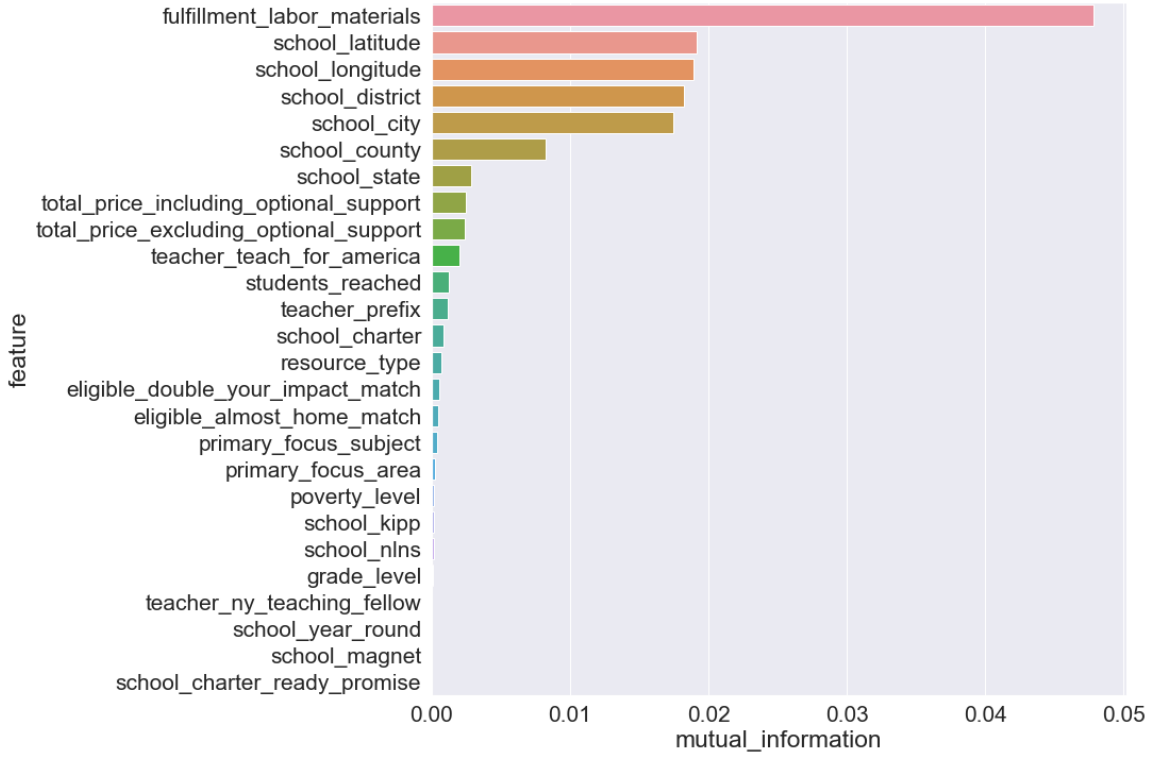

Figure 2: Mutual information of original features against label *is_exciting*.

Hence we will explore whether it is possible to obtain more useful features by creating new features from other information of the project. We will also convert both longitude and latitude into categories, and deploy one-hot encoding for all categorical variables. The features explored are listed below:

Project

- *price_relative_to_focus_area*: The project cost excluding optional tips divided by the median price of the project's focus area.
- *date_posted_relative*: The number of days since the first project in the training data (1 Jan 2010) is posted.

Teacher

- *num_prev_col*: The number of projects that the teacher has posted prior to the current project.
- *num_prev_exciting_col*: The number of exciting projects that the teacher has posted prior to the current project.
- *pct_prev_exciting_col*: The percentage of exciting projects by the teacher prior to the current project. If the teacher has not posted any projects before, this will be 0.

School

- *pct_exciting*: The percentage of exciting projects from the school prior to the current project. As the *is_exciting* label is used to create the feature, this feature is created using a training set and the MI is computed using a test set to prevent data leakage.
- *projects_per_year*: The average number of projects per year by the school.

Location

- *school_latitude_group*: A categorical variable from the latitude to capture how far north or south is the school.
- *school_longitude_group*: A categorical variable from the longitude to capture how far east or west is the school.
- *school_city_state*: A categorical variable that combines the city and the state of the school.
- *school_projects_per_year*: Average number of projects per school in each city-state combination.

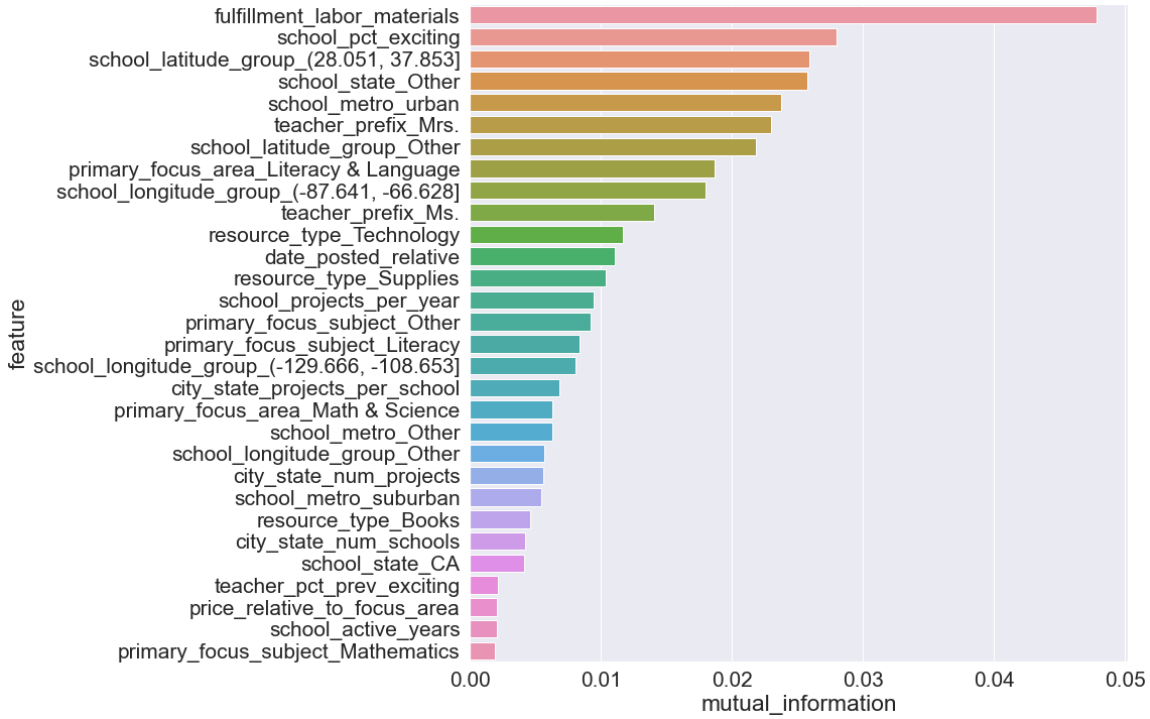

Figure 3: Mutual information of top 30 features against label *is_exciting*.

Exploring the original features and the new features mentioned above, Figure 3 shows the MI for the top 30 features. Compared to Figure 2, we see that the MI is generally higher than the original features, though the absolute values still remain low.

Moreover, we observe a slight cliff effect after the 26^{th} variable - *city_state_CA*. As such, we will only use the top 26 features to fit the models and for subsequent sections.

Model Fitting and Evaluation

For model evaluation, we split the data into the training set (for model training) and the test set (for measuring performance). Instead of a random split, we will use the latest 20% of the projects as the test set, to assess both over-fitting and possible time drift.

How we envisage using the models' output is that the projects will be sorted in descending order using the model's predicted probability of whether a project is exciting. In other words, donors will be seeing projects that the model deems to be more likely to be exciting in earlier pages.

Given this use case and the subsequent section, we will use the following evaluation metrics to evaluate the models:

- **Area under the curve of the receiver characteristic operator (AUC-ROC)**: The AUC-ROC captures the chance that the model gives higher probability to a randomly selected exciting project compared to a randomly selected non-exciting project. The higher the AUC-ROC, the better the performance of the model at distinguishing between exciting and non-exciting projects. This allows us to compare model performance across different requirements, as the proportion of projects that meet the requirement varies significantly across different requirements. When AUC-ROC = 0.5, the classifier is not able to distinguish exciting projects from projects that are not exciting.
- **Share of exciting projects among X-th percentile**: After sorting the projects based on the model's predicted probability, we look at the proportion of exciting projects within the top 10, 20, 30th percentile. A higher number would mean that the model is able to surface more exciting projects to donors at earlier pages, and this would be compared against the benchmark of random ordering - that is if we randomly order the projects, the ratio of exciting project at each page is expected to be the same as the overall share of exciting projects.

We fit 4 types of classifiers to the data: logistic regression, neural networks, random forest classifiers, gradient boosting classifiers.

- Logistic regression models the probability of an exciting project given the predicting features by linking the features through a logit link function.
- Random forest classifiers are an ensemble learning method that builds multiple decision trees independently during training. The predicted probability will then be the proportion of votes from the decision trees.
- Gradient boosting classifiers are similarly an ensemble of weak learners, but it builds one tree at a time in a forward stage-wise manner to improve the shortcomings of the previous tree.
- Neural networks are a algorithm that fits multiple layers of neurons to learn nonlinear decision boundaries.

The AUC-ROC score and the share of exciting projects in the 10^{th}, 20^{th}, 30^{th} percentile of each classifier's predicted probability are shown in Table 3.

- The AUC-ROC scores are similar across different classifiers, and all of them are slightly above 0.5. This means that the classifiers are able to distinguish exciting projects, though just slightly better than a random guess.
- Nonetheless, we note that simpler models (e.g. logistic regression, random forest and gradient boosting classifiers with less trees or depths) tend to perform better. This suggests that introducing more complex classifiers do little in helping to reduce the bias, and could introduce more variance instead.

Model	AUC-ROC	Top 10%	Top 20%	Top 30%	Overall share
Logistic regression	0.600	26.3%	24.0%	22.7%	16.5%
Random forest (5 layers, 1,000 trees)	0.595	24.2%	23.5%	22.7%	16.5%
Random forest (5 layers, 250 trees)	0.595	23.5%	23.7%	22.6%	16.5%
Neural net (3 layers)	0.592	22.5%	23.5%	22.7%	16.5%
Gradient boosting (250 trees)	0.587	24.2%	23.6%	22.9%	16.5%
Neural net (4 layers)	0.587	24.5%	23.9%	22.8%	16.5%
Neural net (5 layers)	0.587	23.6%	23.9%	22.8%	16.5%
Random forest (10 layers, 1,000 trees)	0.587	24.0%	24.1%	23.0%	16.5%
Gradient boosting (1,000 trees)	0.587	24.0%	22.9%	22.9%	16.5%
Random forest (10 layers, 250 trees)	0.586	24.0%	23.0%	23.0%	16.5%

Table 3: AUC-ROC score and share of exciting projects in the k^{th} percentile.

- Looking at the share of exciting projects in the top 10^{th} to 30^{th} percentile, sorting the projects based on the predicted probabilities from the classifiers does surface more exciting projects to the donors (22.6% to 26.3% of projects are exciting) compared to the assumed status quo of random ordering (16.5% of projects are exciting).

Next, we try to model each of the requirements separately, before multiplying the predicted probability of each requirement to obtain the predicted probability of *is_exciting*. For simplicity, we will use most of the same 26 features to model each requirement. The only exception is *pct_exciting*, where instead of *is_exciting*, it computes the percentage of projects that meet each specific requirement .

Requirement	Best model	AUC-ROC
at_least_1_teacher_referred_donor	Logistic regression	0.658
fully_funded	Logistic regression	0.639
at_least_1_green_donation	Logistic regression	0.619
great_chat	Logistic regression	0.606
one_or_more	Logistic regression	0.569

Table 4: Best classifier and AUC-ROC score for each requirement.

Table 4 shows the best classifiers for each requirement in terms of AUC-ROC score.

- Similar to *is_exciting*, the best classifier for all requirements is a simpler model, logistic regression.
- The difficulty of predicting each requirement seems to vary, with the *one_or_more* requirement being the hardest to predict. This could be because the requirement itself is a further aggregation of 3 sub-requirements.
- Most of the AUC-ROC score, with the exception of *one_or_more*, is higher than the AUC-ROC score for *is_exciting*.

Model	AUC-ROC	Top 10%	Top 20%	Top 30%
Logistic regression for *is_exciting*	0.600	26.3%	24.0%	22.7%
Ensemble (logistic regression for each requirement)	0.624	26.9%	26.3%	24.6%

Table 5: Performance comparison between best model in Table 3 vs ensemble of models in Table 4.

Table 5 compares the ensemble model's performance against the simple logistic regression model that we saw in Table 3. We see that the ensemble model is able to outperform the single logistic regression from

Table 3. The ensemble gives a higher AUC-ROC score, while the proportion of exciting projects in the top 10^{th} to 30^{th} percentile of the sorted projects is also slightly higher for the ensemble model.

Conclusion and future work

While the AUC-ROC scores for the models tested were not fantastic, we showed that it could still result in more exciting projects shown to donors, and hopefully reduce the chances that a potential donor may walk away without donating. Furthermore, an ensemble of models with each predicting a specific requirement seems to be able to improve the performance further.

Nonetheless, there are potential areas of improvement that could be explored for future work:

- Since more complex models do not seem to improve the performance, we could look into including more data and features into the model instead. One possibility is the **essays.csv** dataset, which contains text-related information on the project provided by the teacher.
- Since the model for *one_or_more* is the worst performing among all requirements, we could also explore if using an ensemble of models to predict each sub-requirement of *one_or_more* and aggregating the output could improve the prediction accuracy for this particular requirement.