# Project funding prediction for Donors Choose

Ghislene Adjaoute
gnadjaoute@ucsd.edu
Rady School of Management

Gino Slanzi
gslanzir@ucsd.edu
Rady School of Management

Ellen Walsh
epwalsh@ucsd.edu
Rady School of Management

## ABSTRACT

DonorsChoose is a non-profit organization that facilitates donations to classroom projects for public schools within the US. This analysis utilized two datasets with records from 2015-2016, which can be downloaded directly from the company website. A random sample of 50,000 projects data and their application essays were used to predict whether a project was funded or not, using different projects and text features. Our results show that with XXX model, if a project would be funded could be predicted with an xx%. accuracy.

## KEYWORDS

prediction, classification, text mining, sentiment analysis

## 1 INTRODUCTION

Donors Choose is an organization that allows teachers to apply for different funds based on their class needs. They provide an API [1] to connect and extract data and also they have open data resources available for direct downloading [2].

For this work, Projects and Essays datasets were used. The Projects dataset includes 1,108,217 observations of 44 variables to represent the school, teacher, location, project, grade level, funding status, and dates. The Essays dataset includes the same observations of 4 variables to represent the project ID, funding status, title, and text from the grant application for each project. Other information regarding donations, resources and payment type was available too but disregarded for this analysis.

A sample of 50,000 projects were used for simplifying computation times, by taking a balanced random set of funded and not funded projects IDs and joining them with their essay texts.

This paper is structured as follows: a brief description of the predictive task is in part 2, literature review according to this topic is provided in part 3, the model and results are in part 4, and part 5 provides the main conclusions.

## 2 PREDICTIVE TASK

The identified goal is to predict whether a project funding application through Donors Choose is successful with the available information.

The projects dataset's Funding Status variable tells whether a project was funded or not. It has four different values: "Completed" for projects that received full funding; "Expired" projects are ones that expired before donations were made, "Reallocated" means that the donations were moved towards another project and "Live" are those that hasn't been funded yet. Using this information we can label projects as 1 or 0 depending if they were funded or not.

In order to achieve this task different features were computed for using them as input for classification models. More detail in modeling is presented in the Model and Results section.

## 3 LITERATURE REVIEW

This existing dataset came straight from the DonorsChoose website and is readily available for public use to promote discussion and collaboration among researchers to gain insight and draw conclusions on how public education can be improved in the United States. There is in-depth analysis available on Kaggle exploring donor preferences and project classification. There is extensive analysis available on projects that get funding and projects that don't get funding. This project aims to expand further and predict if a project will get funded or not before there is a decision.

An important input for this funding prediction was text. The article "Text Mining Infrastructure in R" outlines [2] the structure we used to manipulate the text in the grant application essays. The approach taken creates a compilation of text, called a Corpus, to which text mining and natural language processing can be run to develop insights. The R package tm is used to utilize this text mining framework. The reason we followed a similar approach to this explanation is because the DonorsChoose dataset includes millions of rows of data. Reading each essay line by line continued to crash our systems. The corpus approach imported the essays from our file and configured the words into a term-document matrix with the rows representing a document, the columns representing a word, and the values representing the term frequency, which is all stored into memory. Important processes we followed are summarized, such as removing punctuation, stop words, numbers, and whitespace, and assigning lower-case capitalization to all text to streamline format. This paper followed similar approaches to complete an in-depth analysis on the R-devel mailing list data on newsgroup postings, compiling 4583 text documents. A term document matrix was designed and the top 3 threads within the mailing list were then identified, which allows users to search for a key-word and observe how many entries contain it. This paper focuses on identification rather than prediction.

A similar approach has been followed by Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah a, and David Chek Ling Ngo [5] to use text data as an input to predict a binary outcome. Text mining and sentiment analysis were conducted on social media posts and web-based news to predict gain/loss in financial markets, in a similar format to our prediction model. The sentiment analysis of financial online posts resulted in classification accuracies as low as 52% and as high as 91.5%, depending on the magnitude of the word. This method was able to achieve a higher accuracy than our prediction because our prediction was limited by the NRC library. Similar conclusions were drawn that the text mining process should be broken down into feature selection, representation, and reduction, after preliminary analysis.

[1] https://www.donorschoose.org/data
[2] https://research.donorschoose.org/t/download-opendata/33

A state-of-the-art method currently employed for studying word representation and extracting meaning from text data is explained by Jorge A. Balazs and Yutaka Matsuo [1] as a vector gate. Sentiment can be difficult to classify and following dictionaries results in limited prediction models; this paper reached the conclusion that a learned vector gate is the best approach to identify the true sentiment of text data. A learned vector gate takes sentiment analysis one step further by integrating word-level representation with character-level representation. Simultaneously accounting for the dimensions of a word and the characters within that word adds an additional layer of analysis and derives more insight than word frequencies alone.

## 4 MODEL AND RESULTS

In this section a detailed description of the modeling process is provided. Two different approaches were taken, as we had two sources of data: projects and essays.

### 4.1 Projects Data

An initial exploratory analysis was conducted on this dataset. The main findings are the following: The probability of project funding has an inverse relationship with the size of the project's goal. The projects from schools with the highest poverty level received the most funding. The project resource most highly funded across all poverty levels were trips. The average target funding for each project based on resource type varied highly across poverty levels. The states with the most funded projects were California, Texas, Illinois, and New York. The amount of students reached had no difference for funded and not funded projects. The success rate varies depending on the month of the application. The majority of the applications tend to focus on Literacy, Literature & Writing and Mathematics.

The features we generated from the Projects dataset were both numerical and categorical, so for the latter we transformed them using One-Hot encoding. The final features set corresponded to: funding goal, students reached, primary focus grade level, poverty level, resource type, eligible for double your impact match, eligible almost home match, teacher for America, school state, month of application.

### 4.2 Essays Data

For the Essays data we conducted Sentiment Analysis and TF-IDF using R in order to generate features for the models.

We initially used Corpus object methodology to reduce the complexity of the text rather than using the a tokenizer function (unnest_tokens) as our dataset was too large. Corpus object in the tm package are preferred for large data sets as they “boost performance and minimize memory pressure [2]. Within a corpus object, we were able to use the tm-map() function to transform our data to reduce the numbers of characters (e.g., lowercase all text, removing stopwords, punctuation and numbers). We finally converted the Corpus object to a Document Term Matrix object from tm and removed terms that had high sparsity.

We applied sentiment analysis using the NRC lexicon (developed by Saif Mohammad and Peter Turney) and the tidytext package on our document term matrix to detect the primary sentiments presents in each essay to estimate whether a strong particular sentiment results in funding [4]. The NRC lexicon model contains of English words that fall into the positive, negative sentiments as well as emotions sentiments. Specifically, the NRC lexicon categorizes words into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust [6].

This sentiment analysis process provided 10 features for each essay representing the amount of words for each emotion that were contained in each one.

Additionaly, we computed TF-IDF (term frequence – inverse document frequency), a statistic which measures the importance and uniqueness of a word into a document in a corpus, obtaining 54 more features (only for the words that matched the lexicon and were not sparse). Finally, the length of each essay was calculated too.

### 4.3 Classification model

Several classification attempts were performed to obtain reasonable results in terms of accuracy and BER. For this we divided our datasets into Train and Test with the 70% and the 30% of the observations respectively.

First we tried using only the Projects features and a Logistic Regression for different values of a regularizer term. The best performance was when we used the 0.1 as regularizer, with a Training accuracy = 0.6478; Training BER = 0.352, Test accuracy = 0.6415; and Test BER = 0.3586.

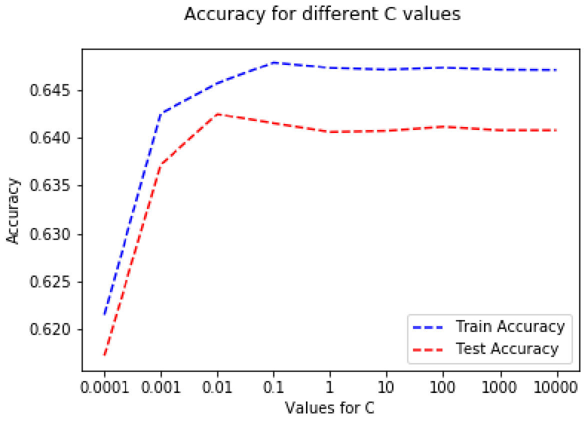

**Figure 1: Accuracy for Projects data features**

Using only text features the results were even worse: for regularizer 1 the Training accuracy = 0.54 and BER = 0.44 as seen in Figure 2.

In order to improve this model we tried training using all features at once. The results were similar to the first approach so we conducted PCA for selecting features. We compared the performance of the models using different numbers of dimensions (from 5 to 120), and the best results were given by 95 components (Training accuracy = 0.6458; Training BER = 0.354; Test accuracy = 0.6396; Test BER = 0.3605).

The last attempt was using Deep Learning using Keras. We trained several neural networks using different combinations of

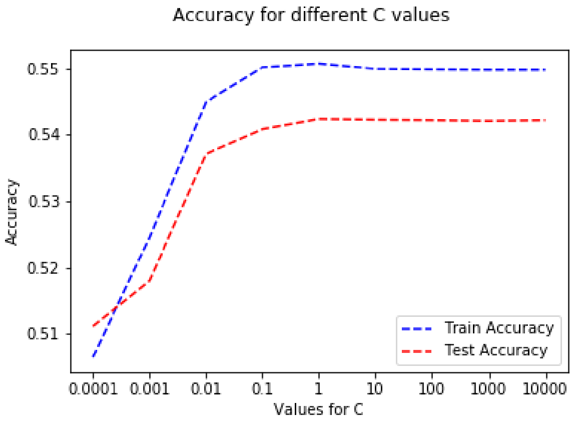

**Figure 2: Accuracy for Essays data features**

hidden layers and neurons. We were able to increase the Training accuracy to 6̃7% and the Test accuracy to 65%, by using 3 hidden layers with 8 neurons each.

## 5 RESULTS AND DISCUSSION

As we can see in the previous section, the results of this work were not as promising as expected. This could be understood as the data we selected for modeling the funding was not as explicative as it seemed in the exploratory analysis, or maybe further transformations should have been made from them.

We hoped that text-mining would provide additional information to enhance or "lift" our model [3]. When running our prediction model with only text features we found a low accuracy of 54.2% using Logistic Regression. Further steps to improve the model would be to apply topic-modeling, specifically Latent Dirichlet allocation (LDA) where we would find "the mixture of words that is associated with each topic of the essays, while also determining the mixture of topics that describe each [essay]" [6]. We could also gather information from grant-writers' experts and grant-review committees through interviews to get insights on specific features of the text they are looking for when determining approval decision for a grant application.

Finally, we could extract keywords from grant application instructions and mission statements text data to create a lexicon library and measure similarity between lexicon and essays.

## REFERENCES

[1] Jorge Balazs and Yutaka Matsuo. 2019. Gating Mechanisms for Combining Character and Word-level Word Representations: an Empirical Study. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, Minneapolis, Minnesota, 110–124. https://doi.org/10.18653/v1/N19-3016

[2] Ingo Feinerer, Kurt Hornik, and David Meyer. 2008. Text Mining Infrastructure in R. *Journal of Statistical Software, Articles* 25, 5 (2008), 1–54. https://doi.org/10.18637/jss.v025.i05

[3] Gary Miner, Dursun Delen, John Elder, Andrew Fast, Thomas Hill, and Robert A. Nisbet. 2012. Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. (2012), xxiii – xxiv. https://doi.org/10.1016/B978-0-12-386979-1.05001-5

[4] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. arXiv:cs.CL/1308.6242

[5] Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. 2014. Text mining for market prediction: A systematic review. *Expert Systems with Applications* 41, 16 (2014), 7653–7670.

[6] Julia Silge and David Robinson. 2017. Text mining with R : a tidy approach. (2017). http://shop.oreilly.com/product/0636920067153.do