IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 6 - V7I6-1345)
Available online at: https://www.ijariit.com

Automating donorschoose.org application screening and building recommendation system using NLP, ensemble methods, and Deep Learning

Adit Shrimal
aditshrimal7@gmail.com
Great Lakes Institute of Management, Chennai, Tamil Nadu

Shanta Narayanan
shants.9@gmail.com
Great Lakes Institute of Management, Chennai, Tamil Nadu

Snehal Sanap
snehal.1993sanap@gmail.com
Great Lakes Institute of Management, Chennai, Tamil Nadu

Trupti Rane
ranetruptiv@gmail.com
Great Lakes Institute of Management, Chennai, Tamil Nadu

ABSTRACT

This paper aims to focus on building a solution for a non-profit organization in automating a task which currently requires a lot of manual effort and is therefore time consuming. We intend to solve this problem with the help of Artificial Intelligence - Natural Language Processing and Deep Learning.

Keywords— *Artificial Intelligence; Machine Learning; Deep Learning; Random Forest Classifier; Neural Network; Natural Language Processing; Donors Choose; Application Screening.*

1. INTRODUCTION

Donors Choose is an online donation marketplace that connects teachers to donors. From the Donors Choose's about page, "Public school teachers post classroom project requests which range from pencils for poetry to microscopes for mitochondria," and individual donors donate 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.

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

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 that 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

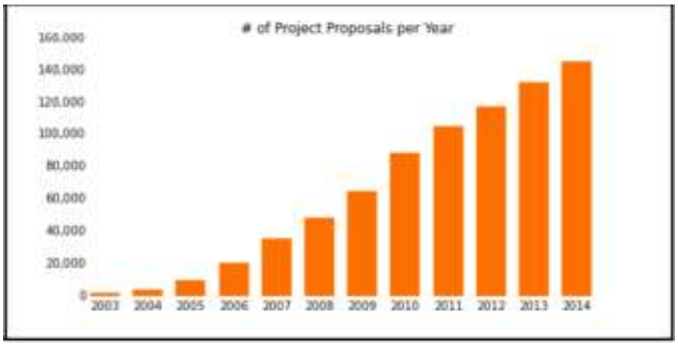


Fig 1: Submitted proposals

According to Donors Choose, the screening process being labor-intensive, 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.

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.

2. MATERIALS AND METHODS

Data

The datasets are provided by Donorchoose.org Kaggle [7]. The datasets have various resources; among which Projects, Teachers, Donors, Donations are extensively worked upon for this paper. Projects dataset contains details related to the project essay, grade category, subject categories/sub-categories, title, submission data time, teacher id, project is approved. Donors dataset contains data related to donor city, state, zip and whether the donor is teacher. Donation dataset is dataset linking the projects and donors' dataset, mainly containing data of project id and donor id, also donation amount and its date and time.

Pre-processing of the Data

After the reading from the Projects, Donors, Donations and Teachers datasets, the data is pre-processed before using it for modelling. For Project dataset, the columns project_essay_1, project_essay_2, project_essay_3, project_essay_4 are merged to form a single column 'project_essay'. New column is added as project_essasy_len which would contain total length of the 'project_essay'. For Projects dataset, the string column containing empty is replaced with NaN and numerical column is replaced with 0. Also, all the string columns are converted into lower case. Then the columns 'project_subject_categories', 'project_title', 'project_subject_subcategories', 'project_resource_summary', 'project_essay' are merged into single column 'project_descp'. The 'project_descp' column is cleaned using re library (regex). The numeric data is scaled using MinMaxScaler and stored in different dataframes

Mean encoding technique is used to convert the categorical features into numerical.

As Donations dataset has no missing values, no modification is done to it. Donors dataset has missing City and zip details for the donors, which is retained in the beginning and columns are dropped later when they are not considered as contributing factors for building the model

Then, both donors, donations and projects datasets are merged and combined into one using inner join. From this new dataset, the text features are considered for creating the word2vec embeddings. The titles and essays are then pre-processed by substituting the apostrophe/short words in both these columns into proper phrases and removing the lease contributing words by using stop words.

Methods used

LSTM

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. LSTM works using feedback connections, unlike standard feedforward neural networks. Other than processing single data points such as images, it can also process entire sequences of data such as speech or video.

CNN

A Convolutional Neural Network (CNN) is a Deep Learning algorithm, generally used in computer vision. But lately, it has been also been applied on NLP tasks, eventually providing promising results

© 2021, www.IJARIIT.com All Rights Reserved

Random Forest Classifier
Random Forest Classifier is an ensemble learning method. It can be used for classification problem, regression problem and other tasks that operate by creating several decision trees at training time and outputting the class that aggregates the votes from the individual trees to select final class

Light GBM
Light GBM is a gradient boosting framework that uses a tree-based learning algorithm. Other tree-based algorithm grows horizontally i.e. level-wise while Light GBM grows vertically i.e tree leaf-wise.

Follow The Regularized Leader (FTRL)
The loss function is a function of w and x, where w vector relates to the weights and x vector relates to the feature vector. The loss for all train data is calculated and summed up. As the weight 'w' which was best fit for all previous iterations is used, this algorithm is thus called Following The Leader algorithm. In this algorithm, the weight's (w) growth is unchecked, which causes large values of weights to be chosen in order to reduce the loss. To restrict from choosing the large values of weights, a regularization term is added to loss function and the sum is optimized. This algorithm is called Follow The Regularized Leader algorithm

Neural Networks
A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so, the network generates the best possible result without needing to redesign the output criteria.

Recommendations techniques
Recommendation techniques are nothing, but the technique wherein we suggest products/services to the end users based on some past behavior. Content-based and collaborative filtering-based recommendation technique would be used in this use case.

Content based recommendations
This method uses only information about the description and attributes of the projects donors have previously donated to when modelling the donor's preferences. In other words, these algorithms try to recommend projects that are similar to those that a donor has donated to in the past. In particular, various candidate projects are compared with projects the donor has donated to, and the best-matching projects will be recommended.

Collaborative filtering-based recommendations
Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. These predictions are specific to the user, but use information gleaned from many users. Applications of collaborative filtering typically involve very large data sets

Methodological steps
The kernel parameters such as quick_run, max_features, embed_size , dpcnn_folds, batch_size and epochs are set. The pre-processed data is then loaded to apply the model.

There are 4 models designed
1. LSTM Model
2. LSTM with filtered input
3. LSTM + CNN
4. Ensemble

LSTM Model

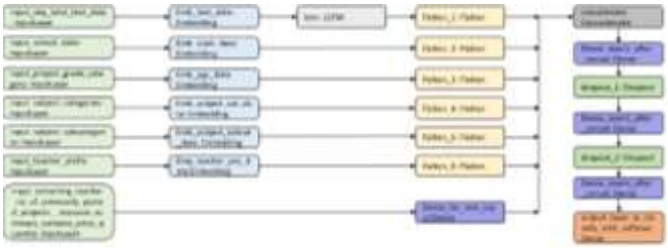


Fig 2: LSTM Model

First Input is text data columns like project essay, on which embedding is done and later it is passed as input to LSTM. The output of the LSTM is then flattened. For the next 5 input as per the fig., various columns like school state, project grade category, subject categories, subject sub categories, teacher prefix are passed through embedding layer individually and then flattened.

The last input contains remaining columns from the train data and resource data which passed through Dense layer. All the output of the above steps is then concatenated and passed through series of Dense and dropout layer and finally through output dense layer with softmax

LSTM Model with filtered input
For this Model, the same model as Model LSTM is built with different first input layer. As first input layer, only some words from total text data (project essay) is be used. It is then filtered as below
1. Train the TF-IDF on the Train data
2. Get the IDF value for each word from the train data.
3. Remove the low IDF value and high IDF value words from data as this doesn't give much information

Model – LSTM + CNN Model

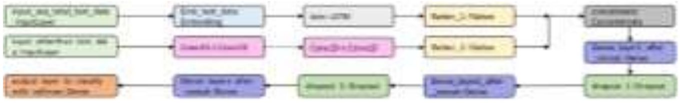


Fig 3: LSTM + CNN Model

The first input is the project essay text column. On this input, embedding layer is applied to get word vectors. Here, predefined glove word vectors are used. The embedding layer output is then passed through LSTM layer which is then flattened

For the second input, all Categorical values are converted to one hot encoding vectors and then all these vectors are concatenated. All the concatenated input is then passed through CNN 1D layer. The output of CNN 1D layer is then flattened

All the output of the above steps is then concatenated and passed through series of Dense and dropout layer and finally through output dense layer with softmax

Ensemble +
As part of the final model, following algorithms are used
- Random Forest Classifier

- Light Gradient Boosting Machine
- Follow the Regularized Leader (FTRL)
- Neural Networks

The main idea of constructing a predictive model by combining different models can be schematically illustrated as below:

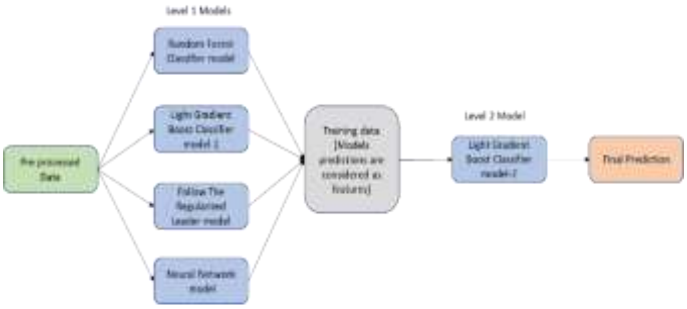

Level 1 Models
Random Forest Classifier model
Light Gradient Boost Classifier model-1
Follow The Regularized Leader model
Neural Network model
Pre-processed Data
Training data [Models predictions are considered as features]
Level 2 Model
Light Gradient Boost Classifier model-2
Final Prediction

Figure 4 Final Model

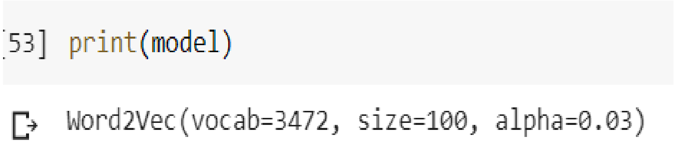
For recommending to Donors
In our case, we are fetching these recommendations based on the word2vec embeddings generated for the text attributes of the dataset.

Obtaining word2vec Embeddings
Now, let's say we have a bunch of sentences and we extract training samples from them in the same manner. We will end up with training data of considerable size.

Suppose the number of unique words in this dataset is 5,000 and we wish to create word vectors of size 100 each. Then, with respect to the word2vec architecture given below:

V = 5000 (size of vocabulary)
N = 100 (number of hidden units or length of word embeddings)

Summary of the model



```
[53] print(model)

Word2Vec(vocab=3472, size=100, alpha=0.03)
```

Visualize word2vec Embeddings

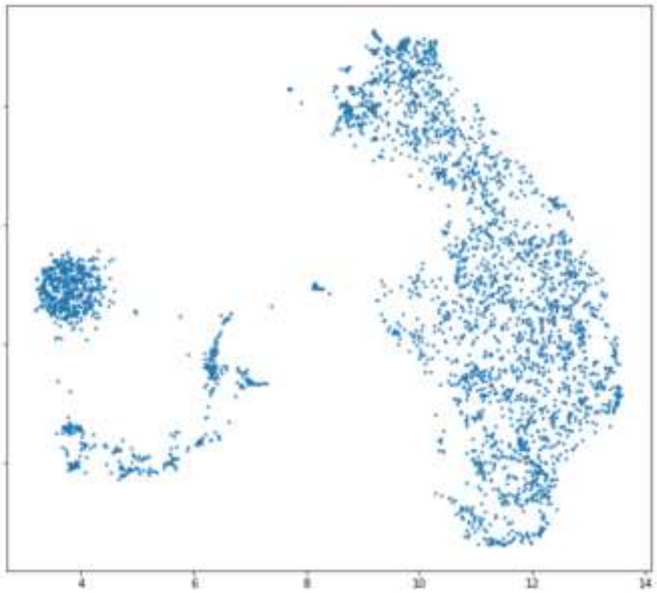
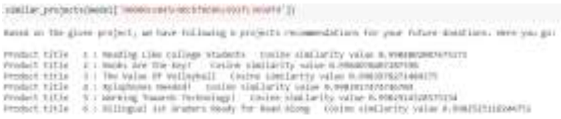


Fig 5: word2vec plot

Every dot in this plot is a project. As you can see, there are several tiny clusters of these data points. These are groups of similar projects.

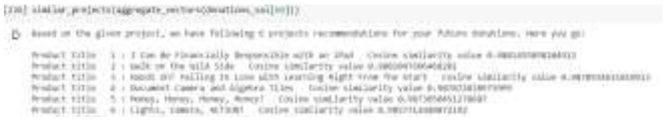
Now that, we have the word2vec embeddings, let's start recommending products

After finding similar projects based on cosine similarity, following output is obtained:



```
similar_projects(model['[illegible]'])

Based on the given project, we have following 6 projects recommendations for your future donations. Here you go:

Product title   1 : Reading Like College Students   Cosine similarity value [illegible]
Product title   2 : Books Are The Key!   Cosine similarity value [illegible]
Product title   3 : The Value Of Volleyball   Cosine similarity value [illegible]
Product title   4 : Xylophones Needed!   Cosine similarity value [illegible]
Product title   5 : Working Towards Technology!   Cosine similarity value [illegible]
Product title   6 : Bilingual 1st Graders Ready for Read Along   Cosine similarity value [illegible]
```

One simple solution is to take the average of all the vectors of the projects the donor has donated for so far and use this resultant vector to find similar products.

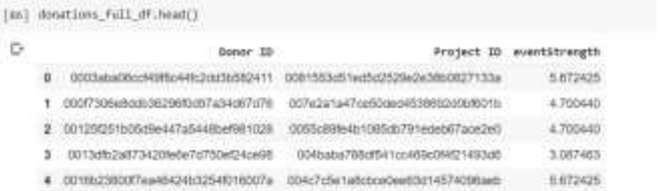


```
[130] similar_projects(aggregate_vectors(donations_[illegible]))

Based on the given project, we have following 6 projects recommendations for your future donations. Here you go:

Product title   1 : I Can Be Financially Responsible with an iPad   Cosine similarity value [illegible]
Product title   2 : Walk on the Wild Side   Cosine similarity value [illegible]
Product title   3 : [illegible] Falling in Love with Learning Right from the Start   Cosine similarity value [illegible]
Product title   4 : Document Camera and Algebra Tiles   Cosine similarity value [illegible]
Product title   5 : Money, Money, Money, Money!   Cosine similarity value [illegible]
Product title   6 : Lights, Camera, ACTION!   Cosine similarity value [illegible]
```

Collaborative filtering
This method makes automatic predictions (filtering) about the preference of a donor by collecting preferences from many other donors (collaborating). It predicts what a donor will donate based on what projects other similar donors have donated to.

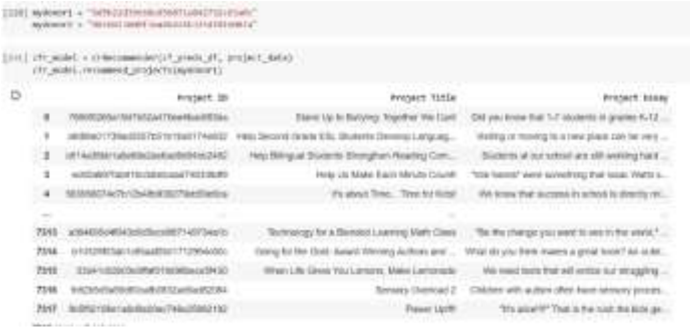
Here also, we used the merged dataset:

Now, let's get the relative strength between the donors and their donations. In short, there are some donors who donate to multiple projects. The amount donated also varies. Let's try to capture that with something called as event strength



```
[85] donations_full_df.head()

                            Donor ID                        Project ID  eventStrength
0   0003aba06ccf49f8c44fc2dd3b582411  0081553d51ed5d2529e2e38b0827133a       5.672425
1   000f7306e8cdb3829680d57a34d67d76  007e2a1a47ce50ded4538882d06f601b       4.700440
2   00125f251b06d9e447a5448bef681028  0055c89fe4b1085db791edeb67ace2e0       4.700440
3   0013dfb2a873420fe6e7d750ef24ce98  004baba788df541cc469c0f4f21493d8       3.087463
4   0018b23800f7ea48424b3254f016007a  004c7c5e1a8cbca0ee63d145740f8aeb       5.672425
```

After applying Collaborative filtering recommender, we get following output



```
[118] mydonor1 = "[illegible]"
      mydonor2 = "[illegible]"

[[illegible]] cfr_model = CFRecommender(cf_preds_df, project_data)
      cfr_model.recommend_projects(mydonor1)

                     Project ID                               Project Title                            Project Essay
0     [illegible]                  Stand Up to Bullying: Together We Can!   Did you know that 1-7 students in grades K-12 ...
1     [illegible]    Help Second Grade ESL Students Develop Languag...   Moving or moving to a new place can be very ...
2     [illegible]     Help Bilingual Students Strengthen Reading Com...   Students at our school are still working hard ...
3     [illegible]                   Help Us Make Each Minute Count!   "Idle hands" were something that Isaac Watts s...
4     [illegible]                     It's about Time... Time for Kids!   We know that success in school is directly re...
...   ...                                                         ...                                               ...
7143  [illegible]            Technology for a Blended Learning Math Class   "Be the change you want to see in the world." ...
7144  [illegible]      Going for the Gold: Award Winning Authors and ...   What do you think makes a great book? As a st...
7145  [illegible]      When Life Gives You Lemons, Make Lemonade   We need tools that will entice our struggling ...
7146  [illegible]                                Sensory Overload 2   Children with autism often have sensory proces...
7147  [illegible]                                         Power Up!!!   "It's alive!!!" That is the rush the kids ge...
```

Results

For classifying whether project can be approved, the initial benchmark acquired by applying LSTM, LSTM with filtered output and LSTM + CNN was 0.701, 0.728, 0.744 , respectively. After implementation of the final model, the ROC_AUC score is 0.8084 or 80.84%

3. DISCUSSION AND CONCLUSION

Summary

The Donorchoose.org which is a non-profit organization helps a big deal in connecting the donors to the relevant projects, that they can donate for. While the project is posted by the teacher on the website, there is a step by step manual process to validate the details presented in there, before the project starts appearing on the website. This is done by the volunteers based on the details furnished for project essays. Manual essay screening involves eyeing the important features like resource description, student experience description and to make sure that the privacy of the students or schools are maintained by following certain norms. Details given as a part of resource description needs to have the exact resource required by the student and teacher should have explained how would the student use the resource and how can it better the experience of learning for the student under the part student experience description. Donorchoose screeners check for missing data or violations for these areas and send the application back to the teacher or further details.

Our research and modelling using machine learning has shortened this lengthy manual process. Using all the details provided, by the teacher, in the various essays' sections, we have predicted whether a project will be approved by the screeners or not. Our work majorly focuses on the description given by the teacher for a project which is used in the context of artificial intelligence to generate a view whether it is a prospective project for getting approved and be listed in the website. We have used natural language processing in combination with CNN, neural networks and many ensembles put together to achieve a good accuracy in this whole activity.

One of the top challenges faced by any online crowdfunding websites is retention of the donors who contributed for the wellbeing of the people. Donorchoose is no different and faces a major challenge in having repeat donors when it comes to having their new projects funded by their same old donors. So, our work has contributed in this direction by building a recommender system that tries to connect the donors to the projects that focus their relevant areas of interests. Using their past donations, we have learnt a lot about the donors to understand what they will be interested in and their way forward. Looking at the various combination of the donors and their regular areas of donations, we have drawn conclusions, what are the potential projects for a donor. This would keep the donor motivated and he or she will get a feel of a customized approach when they login to the website. A model like ours could prove to be very extremely useful for crowdfunding platforms and non-profit organizations to efficiently target fundraising campaign efforts.

Implications:

Our first model that helps in auto-approval of the application to the website saves immense human efforts and hence those volunteers can spend their time on more nuanced and detailed project vetting processes, including doing more to help teachers develop projects that qualify for specific funding opportunities. Apart from this, we have eased the job of donors by giving them top recommendations for other projects they might be interested in, based on their past donations. Thus, we have also been successful in connecting the donors with the projects that most inspire them. Through our exploratory analysis, we have shown the 10 percent of donors are teachers themselves and further distribution of this based on titles is also shown. Our graphs also show that California is the state that received maximum donation whereas Wyoming is the one to receive the least of donations. We have drawn many more such insights from our research based on the data.

Future work and limitations:

Our model uses the evolved LSTM for predictions with ensembles, but we haven't explored the transfer learning part of NLP like BERT which is so popular for language-based tasks. This could be an area we could explore further. As a part of predicting whether a project will be funded or not, we have only taken into consideration the title and essays, whereas, demographics of where project is posted may also have an effect of its chances of being funded. In the future, our models can essentially cover that area too. In case of recommendations, we have used the content based and collaborative filtering, whereas evolved ecommerce websites use hybrid filtering, filtering based on demographics, gender etc. for amazing recommendations. Our models don't cover popularity-based recommendations and that is why it would be difficult to recommend to someone who is brand new to Donor choose website.

4. REFERENCES

[1] D. Reddy, "Factorization machines and follow the regularized leader for dummies," Medium, 21-May-2019. [Online]. Available: https://medium.com/@dhirajreddy13/factorization-machines-and-follow-the-regression-leader-for-dummies-7657652dce69. [Accessed: 31-Oct-2020].

[2] ranliu, "Donor-project matching with recommender systems," Kaggle.com, 19-May-2018. [Online]. Available: https://www.kaggle.com/ranliu/donor-project-matching-with-recommender-systems. [Accessed: 31-Oct-2020].

[3] T. Althoff and J. Leskovec, "Donor retention in online crowdfunding communities: A case study of DonorsChoose.Org," Proc. Int. World Wide Web Conf., vol. 2015, pp. 34–44, 2015.

[4] G. Edenhofer, A. Collins, A. Aizawa, and J. Beel, "Augmenting the DonorsChoose.Org corpus for meta-learning," AMIR@ECIR, 2019.

[5] T. Yiu, "Understanding random forest - towards data science," Towards Data Science, 12-Jun-2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2. [Accessed: 31-Oct-2020].

[6] P. Mandot, "What is LightGBM, How to implement it? How to fine tune the parameters?," Medium, 17-Aug-2017. [Online]. Available: https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc. [Accessed: 31-Oct-2020].

[7] "Data Science for Good: DonorsChoose.org." .

[8] P. Joshi, "Build a recommendation system using word2vec in Python," Analyticsvidhya.com, 30-Jul-2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/07/how-to-build-recommendation-system-word2vec-python/. [Accessed: 31-Oct-2020]