DSC Capstone Project 2

Donors Choose: Donation Recommendation System

Problem: DonorsChoose.org funds classroom requests through the support of over 3 million donors. DonorsChoose.org is unique because it allows the donor to fund specific projects for specific schools, and specific teachers. Donors are able to browse projects, and fund ones that are important to them. Requests come in the form of all kinds of items, from books, to furniture, to electronics. In order to continue supporting public school classrooms, DonorsChoose.org must work to ensure that first time donors continue to make ongoing donations. Donors typically first visit DonorsChoose.org to make a one time donation, and for the sustainability of DonorsChoose.org it is important to convert one time donors into recurring donors.

Dataset: The dataset was obtained from https://www.kaggle.com/donorschoose/io. This dataset had been shared during a previously held contest. The data was initially broken up into 6 different .csv files (donations, donors, projects, teachers, resources and schools). In order to understand the data better, all data was initially examined separately.

I initially cleaned and began exploring this dataset in Jupyter notebook, primarily using pandas. I quickly discovered that using my current toolset in Jupyter notebook was not going to be effective to process the large amount of data that I was working with. I moved my work to Databricks in order to begin using PySpark. This proved to be an exciting challenge.

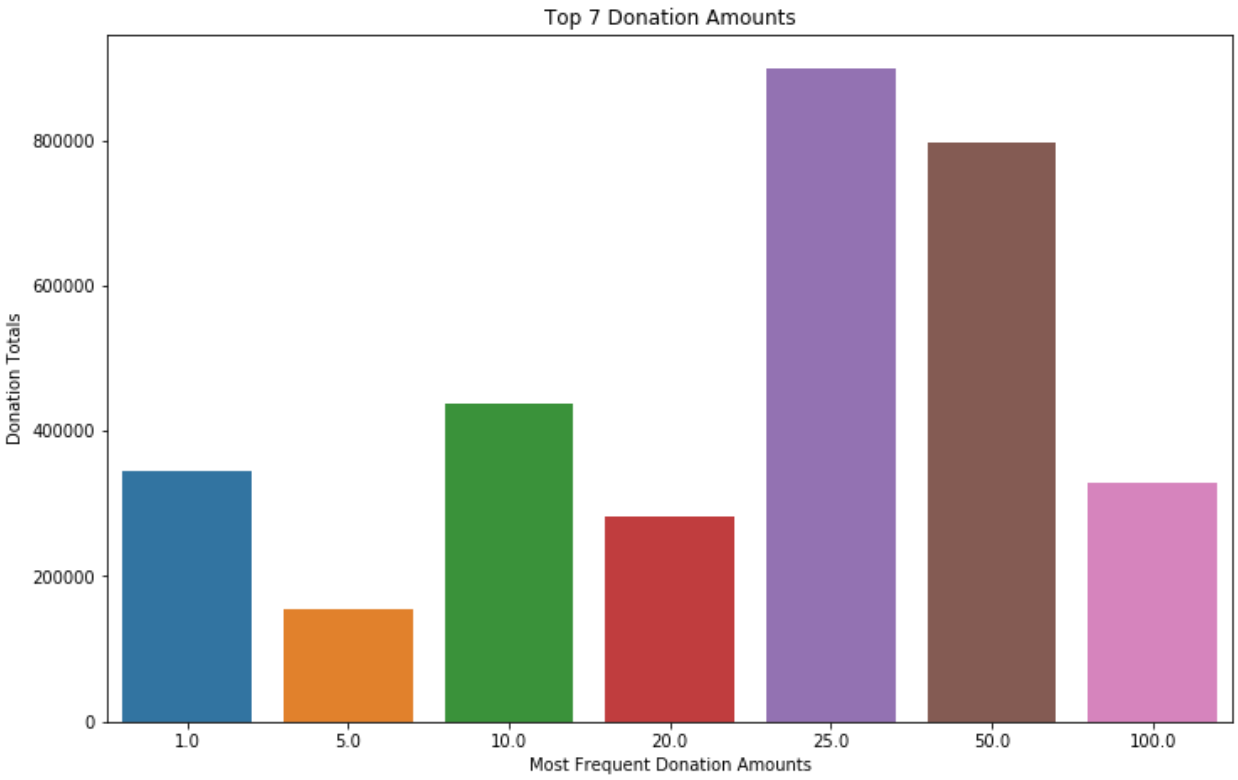
Exploratory Data Analysis

Data Cleaning: In order to process the data in Databricks, the datasets were uploaded to an S3 bucket in AWS. I used EMR to create a cluster and notebook in AWS, due to the size of the dataset. In order to prepare the Projects .csv for import, a schema needed to be created that addressed each data format type per column. Additionally, there were initially issues with multiline text and the initial data import. Text columns, like project essay, were spilling over from their respective column into neighboring columns, and sometimes into the next row depending on the length. This was addressed by ensuring that the Projects .csv had the escape key parameter changed to ‘ “ ‘, along with multiLine set to True, and the schema applied at import.

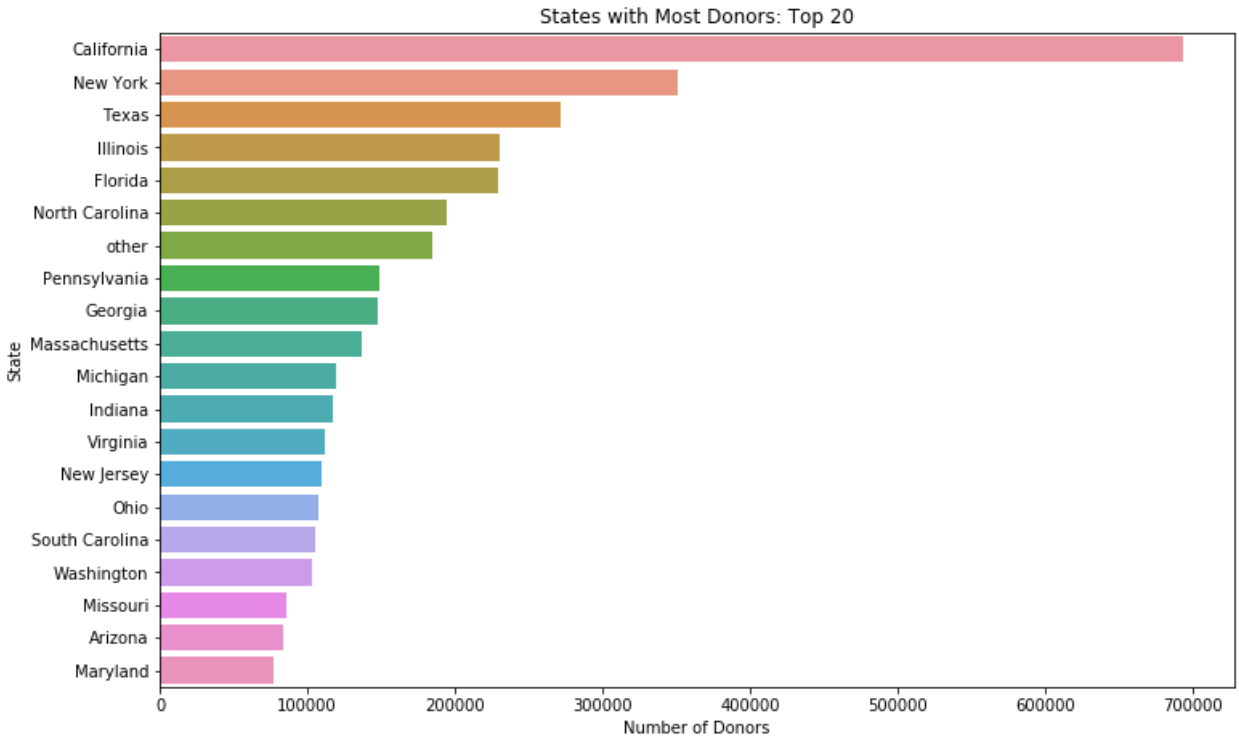
Once all of the data was read into their respective dataframes, I combined them into one dataset by joining on the ID columns (school_id, project_id, teacher_id).

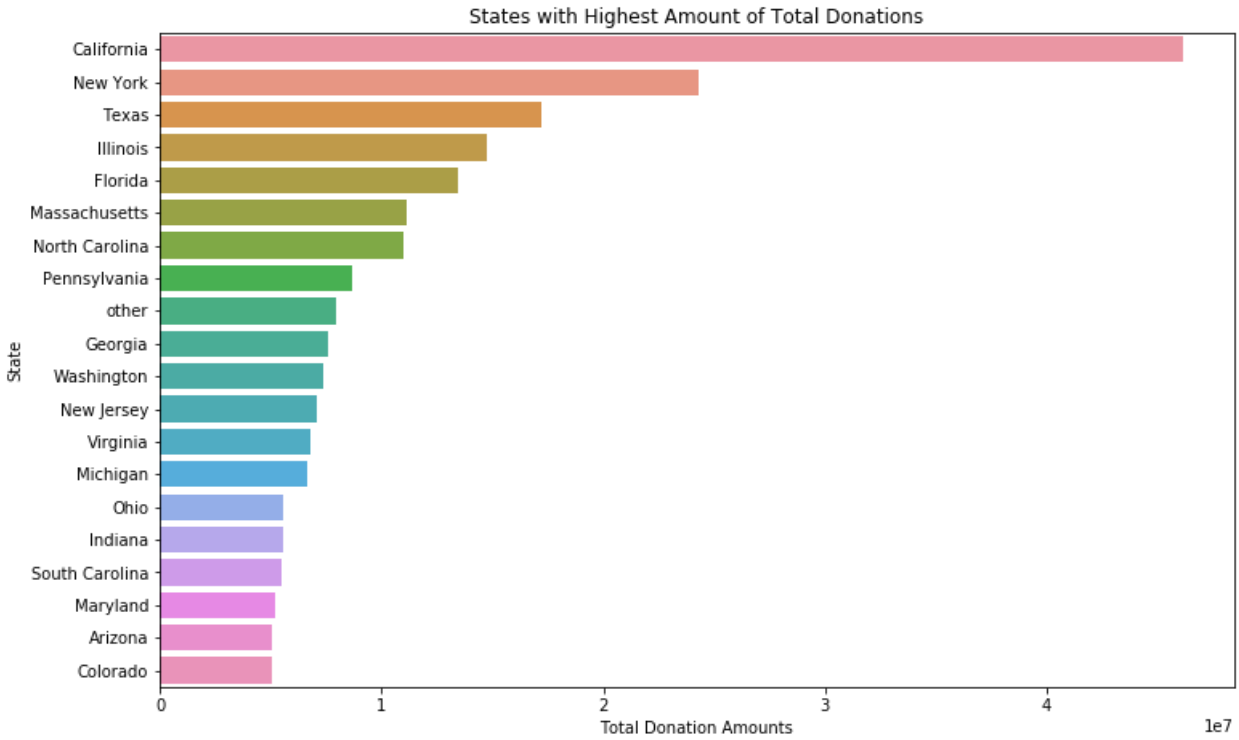
Initial Findings: In order to better understand the behavior of donors, much of the exploratory data analysis is focused on the donors, their donations, and the schools and projects that they donate to most frequently. In examining the data, I was curious who was making the highest donations, how frequently donors made recurring donations, did their donations occur at the same schools, and were the donations made for similar projects. I was also interested in determining whether there was a specific time of the year when donations increased or decreased.

- There are 2,122,640 total donations among 2,003,188 donors. 552,941 are repeating donors. This only accounts for 26% of donors! Additionally, I found that the number of recurring donors significantly decreases after donors who make more than two donations:
    - More than 2 donations - 278,039
    - More than 5 donations - 98,487
    - More than 10 donations - 40,299

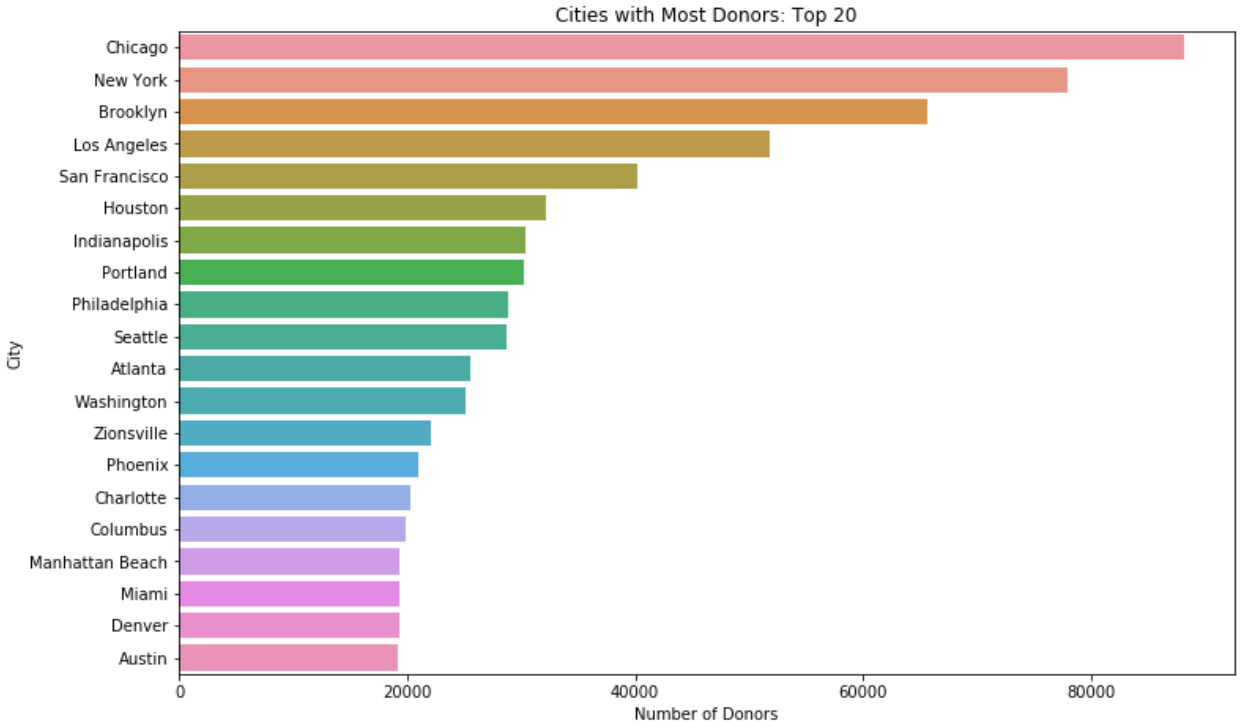


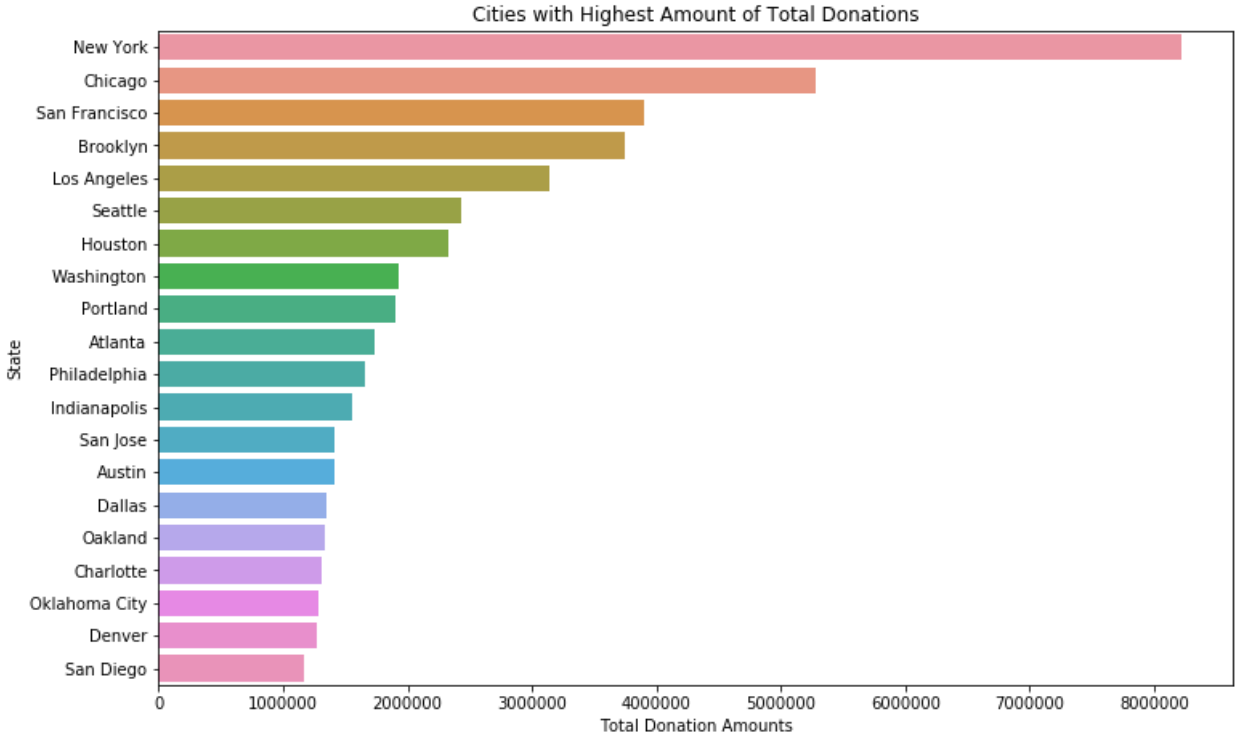
- $25, $50, $10, $1.00, $100.00, $20, and $5 are the most frequently donated amounts - it's likely that these are pre-set donation options on a donation page.
- Donors are identifiable by a randomized donor ID - based off of this, we're able to determine that the top donor has made more 18035 total donations. This top donor has donated a total of $37,121 and is from Manhattan Beach, California.
- Surprisingly, that donor ID is not the same as the donor ID who has made the most (total) donations to Donors Choose projects, totaling $1,864,016. This is an interesting find, and leads me to believe that there are donors who donate small amounts, over a long period of time and there are donors who instead donate less frequently, but with higher donations.



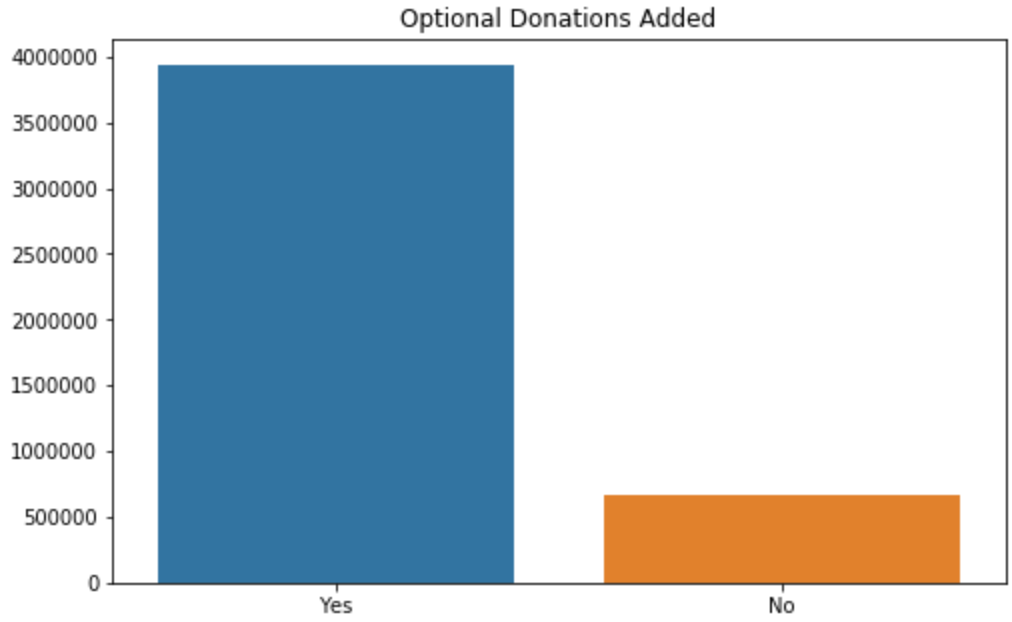


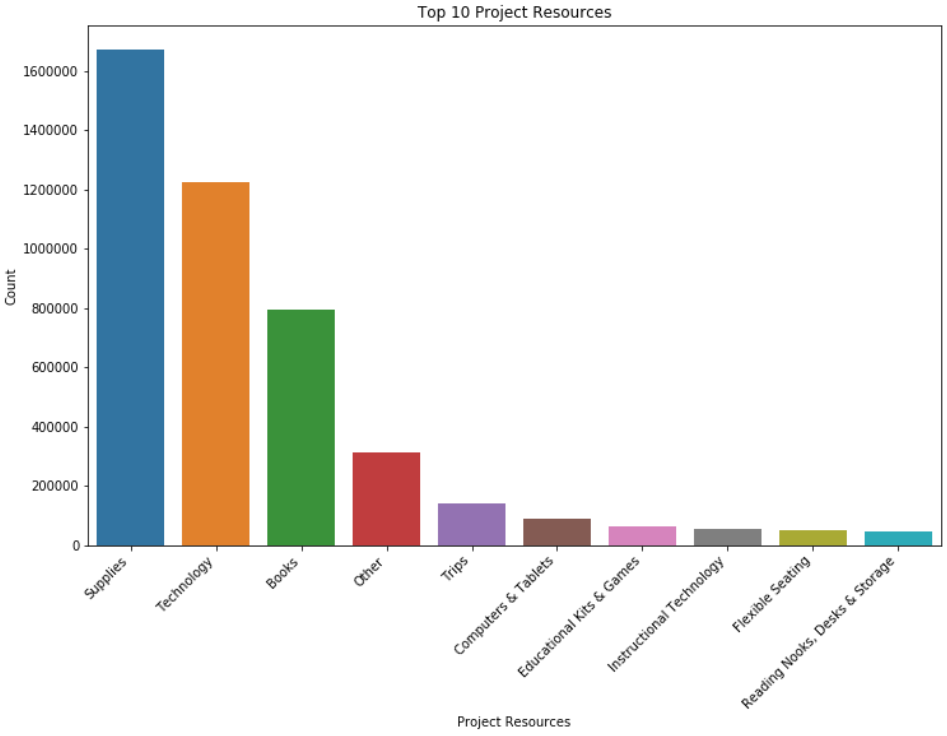
- The top three states with the most unique donors are also the same top three states that have the highest total of donations: California, New York, and Texas. Due to the size of these states, it makes sense that they would be the top three states for both the amount of donors and the highest total donations.


Cities with Most Donors: Top 20
Chicago
New York
Brooklyn
Los Angeles
San Francisco
Houston
Indianapolis
Portland
Philadelphia
Seattle
Atlanta
Washington
Zionsville
Phoenix
Charlotte
Columbus
Manhattan Beach
Miami
Denver
Austin
City
0
20000
40000
60000
80000
Number of Donors


Cities with Highest Amount of Total Donations
New York
Chicago
San Francisco
Brooklyn
Los Angeles
Seattle
Houston
Washington
Portland
Atlanta
Philadelphia
Indianapolis
San Jose
Austin
Dallas
Oakland
Charlotte
Oklahoma City
Denver
San Diego
State
0
1000000
2000000
3000000
4000000
5000000
6000000
7000000
8000000
Total Donation Amounts

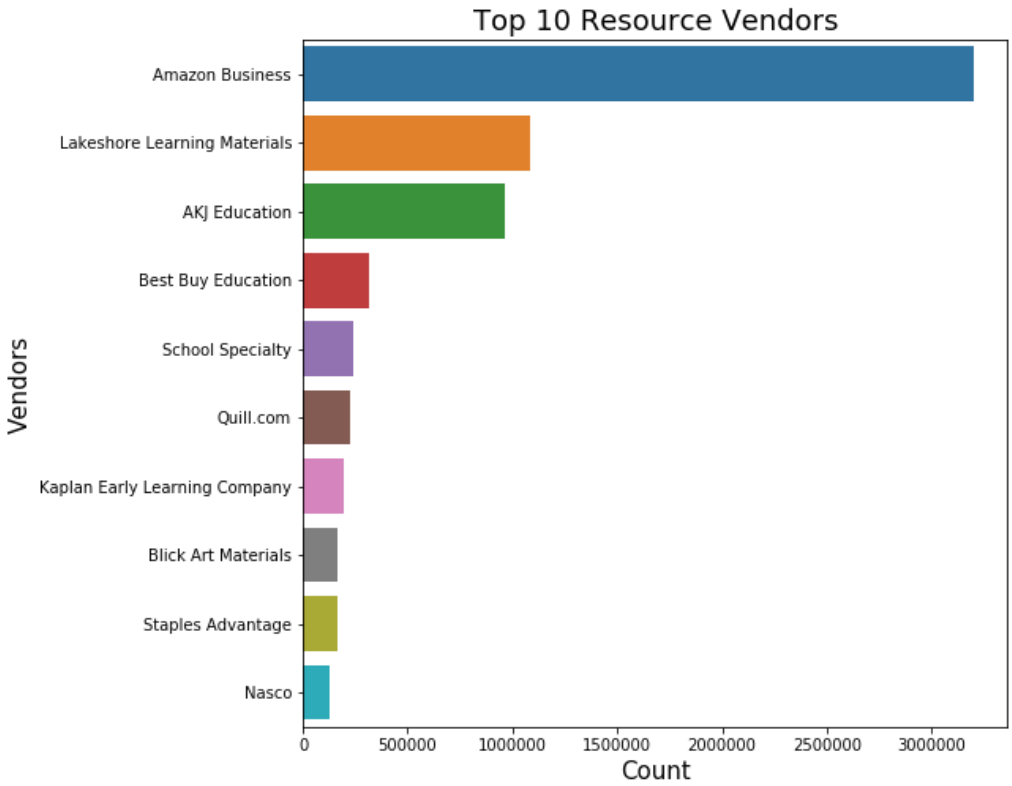
- Cities with the most donors and highest donations stray from the states slightly - the top three cities for unique donors are: California, New York and Brooklyn - while the top three cities for highest amount of donations are New York, Chicago and San Francisco.
- The majority of donors choose to add an optional donation when they are donating to a specific project. This means that they will donate 15% of their project donation to Donors Choose.

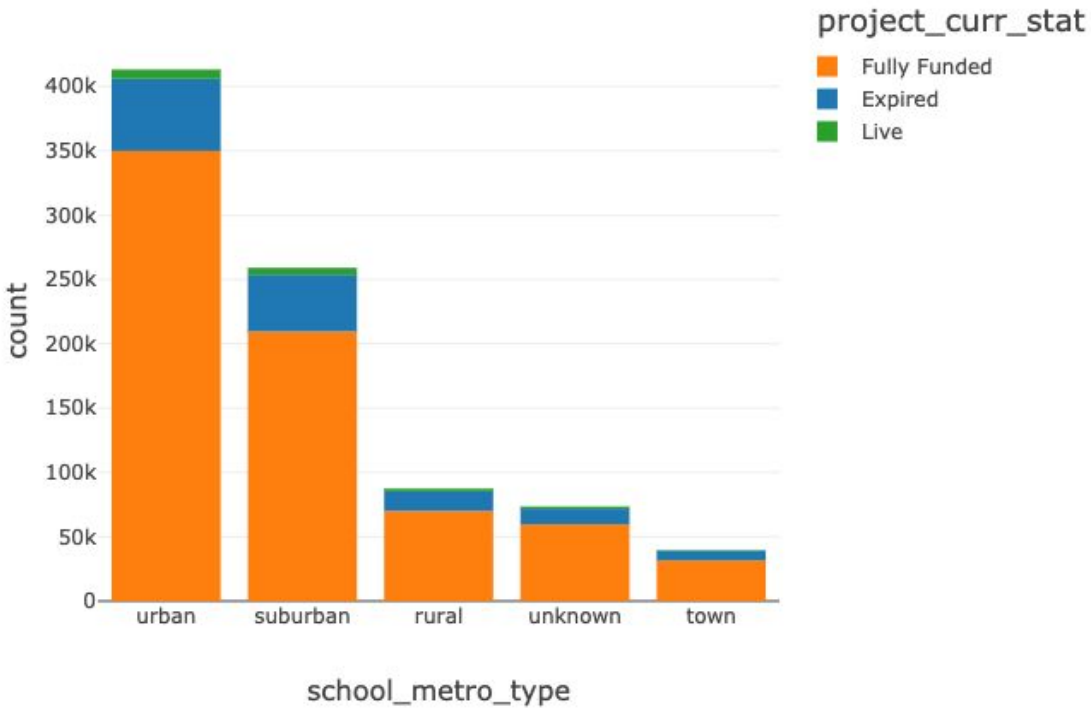




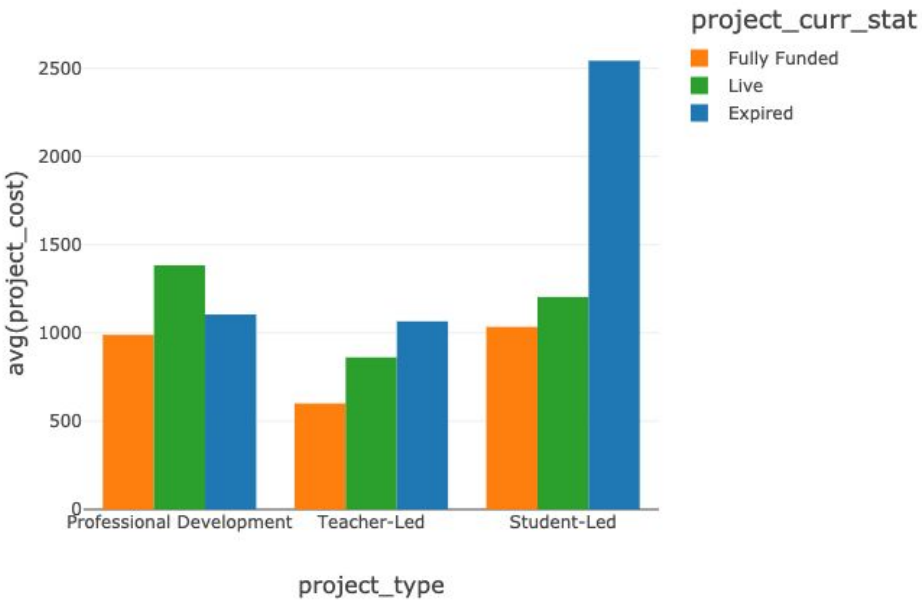
-

- Amazon leads the top resource vendors, followed by Lakeshore Learning Materials and AKJ Education.





- School areas that are in the most need are in urban areas. Schools in urban areas also have the most projects funded by donations.



- Projects led for Professional Development tend to be the most successful projects. Student led projects and projects for Professional Development have similar average project costs, however student led projects tend to expire more often then they are fully funded.

main_cat1	count ▼
Literacy & Language	15097325
Math & Science	5281522
Applied Learning	3486772
Music & The Arts	1546474
History & Civics	1372551
Health & Sports	1290419
Special Needs	1007901
Warmth	336642
null	733

- The top three project main categories are Literacy & Language, Math & Science and Applied Learning.

Summary & Next Steps: This exploratory analysis has helped to better understand donors who make one donation vs. donors who make recurring donations, and which areas one-time donors are most heavily populated in. This information, coupled with an understanding of the most requested supplies as well as areas and schools that are in the most need, provides important insight into potential future campaigns to convert one-time donors into recurring donors.

Based off of this initial exploratory data analysis, there is insight into where future donation campaigns may potentially focus in the future. The data provides a glimpse into understanding the areas of most need - urban area schools. The top cities for donors reflect this, as cities such as Chicago, New York, Brooklyn, Los Angeles, and San Francisco see the most donors. The projects most donated to, the resources most requested and the project main category are additional features which can potentially help identify one time donors who could be converted to recurring donors. It is also interesting to note that Amazon Business is the top resource provider. Due to the amount of people who use Amazon.com regularly across the country, a future partnership with Donors Choose could potentially help retain donors as well.

Modeling

The objective of this project is to match donors with new projects that may interest them based on their prior donations. In doing so, this could create momentum for donors who have possibly only donated to one project in the past. There are three different types of recommendation systems that I explored using to solve this problem:

- **Collaborative Filtering:** This method is used to make automatic predictions about project preferences that a donor may have. Based on projects that they've donated to in the past, this method will predict future behavior based on similar projects that other donors have donated to.
- **Content Based Filtering:** Alternatively, content based filtering uses the information from the description, project essay and project need statement of previous projects that a donor has donated to. The content based filtering method then predicts other projects to these donors based upon their content similarities.
- **Hybrid (mix between Content Based and Collaborative Filtering):** A hybrid approach combines collaborative filtering methods and content based filtering methods. By using a combination of project content and prior project donations, recommendations can be made for future projects.

I chose to use the Alternating Least Squares model, which can be run using either explicit ratings or implicit ratings. Explicit ratings are given when a user explicitly rates an item. For example, in the MovieLens data set, a movie title can be rated on a scale of 1 - 5. Implicit ratings, on the other hand, are created when there are no explicit ratings for an item. Implicit ratings typically stem from user behavior. For example, in the Million Songs data set, the implicit ratings come from the number of times a user listens to a particular song.

Instead of exploring and comparing the performance of two different recommendation models as mentioned above, it is instead more interesting to evaluate two collaborative filtering models that use different implicit ratings. For the Donor Choose data set, and other data sets like it, implicit ratings are the only type of rating that is able to be used in a recommendation model such as ALS. Determining which implicit ratings in a data set will lead to the most accurate predictions would be incredibly important for any company that is interested in creating a recommendation system. I chose to use the Alternating Least Squares model, which can be run using either explicit ratings or implicit ratings.

ALS Model #1 - Total Donation Amount

Pre-Process Data: In order to prepare the data for the first ALS model, it needed to be cleaned and manipulated into a new dataframe. Because our dataset does not have any sort of ranking system in which the donors can rate the projects they're donating to, I created an implicit rating by using the donation amounts per project, per donor. In order to do this I originally created a table with the donor ID, project ID, and donation amount (sum):

donor_id ▲	project_id	donation_sum
17f3f75fe1d4c9058f9a36720b312fbc	0000c0bdc0f15bd239cfffa884791a10	175
5d499cba7badd9eb0dce8b00b6a4370b	0000c0bdc0f15bd239cfffa884791a10	70
72b04f710931d0699e0975d8d40a103e	0000c0bdc0f15bd239cfffa884791a10	175
99488c9de95c1f7ae3ca4748bf211744	0000c0bdc0f15bd239cfffa884791a10	175
9f845a4a86f2982c3f5f402de41377f5	0000c0bdc0f15bd239cfffa884791a10	140
a3cfc28ca16f2927d02203c142279a16	0000c0bdc0f15bd239cfffa884791a10	158.7600021
a7fa881a86cb4c7e70815b38168d1702	0000c0bdc0f15bd239cfffa884791a10	21
e1a9ef3d18fad06f5ab2f7c1aab38e2e	0000c0bdc0f15bd239cfffa884791a10	350
ef5f051d5008f36e81f14f0d38b0e69a	0000c0bdc0f15bd239cfffa884791a10	140
fa843f037182711c3ea4483983c8af2b	0000c0bdc0f15bd239cfffa884791a10	350
fcec0988e54a7607f172e13079839b4e	0000c0bdc0f15bd239cfffa884791a10	175

Once this table was made, I created a new column in the table that is the log of each donation amount, as it is a convenient way to express large numbers:

donor_id	project_id	donation_sum	donation_log
17f3f75fe1d4c9058f9a36720b312fbc	0000c0bdc0f15bd239cfffa884791a10	175	5.164785974
e1a9ef3d18fad06f5ab2f7c1aab38e2e	0000c0bdc0f15bd239cfffa884791a10	350	5.857933154
ef5f051d5008f36e81f14f0d38b0e69a	0000c0bdc0f15bd239cfffa884791a10	140	4.941642423
fa843f037182711c3ea4483983c8af2b	0000c0bdc0f15bd239cfffa884791a10	350	5.857933154
99488c9de95c1f7ae3ca4748bf211744	0000c0bdc0f15bd239cfffa884791a10	175	5.164785974
a3cfc28ca16f2927d02203c142279a16	0000c0bdc0f15bd239cfffa884791a10	158.7600021	5.067393641
5d499cba7badd9eb0dce8b00b6a4370b	0000c0bdc0f15bd239cfffa884791a10	70	4.248495242
9f845a4a86f2982c3f5f402de41377f5	0000c0bdc0f15bd239cfffa884791a10	140	4.941642423
a7fa881a86cb4c7e70815b38168d1702	0000c0bdc0f15bd239cfffa884791a10	21	3.044522438
fcec0988e54a7607f172e13079839b4e	0000c0bdc0f15bd239cfffa884791a10	175	5.164785974
72b04f710931d0699e0975d8d40a103e	0000c0bdc0f15bd239cfffa884791a10	175	5.164785974
ab00b62ea86ec20c2eb4a69b40c6fa80	0000c0bdc0f15bd239cfffa884791a10	350	5.857933154

In order to finish prepping the data for the ALS model, the donor_id column and project_id column were selected into their own data frames by distinct value. Once in these dataframes, integer IDs were assigned to each before being re-joined with the original table. After renaming and dropping old columns, the table ready for modeling was completed:

donor_int_ID	project_int_id	donation_log
0	0	5.164785974
1	0	5.857933154
2	0	4.941642423
3	0	5.857933154
331792	0	5.164785974
331793	0	5.067393641
663060	0	4.248495242
663061	0	4.941642423
994514	0	3.044522438
994515	0	5.164785974

Lastly, all columns were formatted as integers, as that is the only format that ALS will accept. With the data frame cleaned up, we can begin creating our model!

Model Preparation: In order to prepare the data for the model, I split it into a training and testing set, using a .6/.4 split. Once the data was split, the training data was fit to the ALS model with default parameters. The testing data was then used to make predictions.

Alternating Least Squares: The ALS model with default parameters was run on the training and test data, and then evaluated using Root Mean Square Error (RMSE). The training data was first fit to the ALS model, and then the testing data was used to create predictions. When the ALS model was run on the testing data, using default parameters, the resulting RMSE was 1.9427716349485609.

Hyperparameter Tuning: In order to determine what the best parameters would be for this model, a parameter grid was created using a 3 fold cross validation. By fitting the training data to the cross validator, we were able to determine the best parameters for rank, maxiter and regparams. With these parameters added to the ALS model, the new RMSE is 1.8975567342104507.

The RMSE determines that the predicted recommendations would fall within the range of 1.90, below or above. The total range of the donation sum, converted to logs is -4 and 15. WIth that in mind, a 1.90 RMSE seems to be a fairly decent indicator of recommendation accuracy.

ALS Model #2 - Number of Donations per Project Category

Pre-Process Data: Preparing the data frame for ALS Model #2 was similar to ALS Model #1. Because our dataset does not have any sort of explicit ranking system in which the donors can rate the projects they're donating to, the second ALS model was created using the amount of times a donor made a donation to a particular category. Within the Donors Choose dataset, there were several categories listed per project. For the model, only the main category #1 was used. I began by creating a new table that consisted of the donor_id, main_cat1 and the donation_id. These identifiers were then transformed into integer ID's becoming userID and catID as well as their count:

userID	catID	count
1712923	7	10187
981150	7	5547
180721	6	5303
1231374	6	4584
1712923	6	2876
1712923	3	2165
981150	6	3054
981150	3	2156
981150	2	1774

Model Preparation: In order to prepare the data for the model, I split it into a training and testing set, using a .6/.4 split. Once the data was split, the training data was fit to the ALS model with default parameters. The testing data was then used to make predictions.

Alternating Least Squares: The ALS model with default parameters was run on the training and test data, and then evaluated using Rank Ordering Error Metric (ROEM). The ROEM function used was adapted from here, because PySpark and MLib do not currently have ROEM validation functionality. ROEM was used as an evaluator for this model because it checks to see if categories with a higher number of donations have higher predictions. The training data was first fit to the ALS model, and then the testing data was used to create predictions. When the ALS model was run on the testing data, using default parameters, the resulting ROEM was 0.18879302687557417. ROEM values close to .5 indicate that the predictions are not any better than random, so to begin with this ROEM already has fairly accurate predictions.

Hyperparameter Tuning: In order to determine what the best parameters would be for this model, a hyperparameter cross validator function was created using a 3 fold cross validation. The ROEM_cv function loops through all possible ALS models using the parameter grid specified, and determines the best parameters of: rank, maxiter, regparams and alphas. By fitting the training data to the cross validator, we were able to determine the best parameters for rank, maxiter, regparams and alphas. With these parameters added to the ALS model, the optimized ROEM is: TBD

Conclusion & Next Steps

Donors Choose has multiple options to move forward with in determining how to retain donors. Donors appear to be drawn to multiple features of projects such as the school that the project is coming from, the type of project based on the category, and the type of resources needed by the school. This gives

Donors Choose the unique opportunity to determine which features mean the most to a donor in determining whether or not they will make a future donation. Donors Choose currently has about 26% of donors that make repeat donations. With even a small increase of recurring donors, Donors Choose will see a huge impact in their overall donations.

In order to attract new recurring donors and to retain existing recurring donors, I would recommend a campaign such as email marketing that encourages donors to look at projects similar to those they have donated to in the past. With additional time and resources, Donors Choose should be able to follow a similar framework to ALS Model #2, and create a recommendation model based off of the features of projects that appeal the most to donors.