Machine Learning Engineer Nanodegree

Capstone Project

Anirudh Varanasi
August 1, 2018

Title

Application Screening for DonorChoose.org

I. Definition

Project Overview

Donorchoose.org is a crowdfunding platform, which connects public school teachers and donors. A high school teacher from the Bronx founded this organization in 2000. This organization empowers public school teachers from across the country to request much-needed materials and experiences for their students. Teachers submit project proposals to Donorchoose.org then manually screens and approve the proposals before posting on the website. Right now, a large number of volunteers are needed to manually screen each submission before it is approved by DonorChoose.org. Next year, DonorChoose.org expects to receive close to 500,000 project proposals. The goal is to predict whether or not a DonorChoose.org project proposal submitted by a teacher will be approved.

Problem Statement

Next year, DonorChoose.org expects to receive close to 500,000 project proposals. As a result, manual screening of application would be a tedious task. Using machine learning, through this project an attempt to discover a classification model that predicts which application to accept or reject. There are three main problems they need to attention:

1. How to scale current manual processes and resources to screen 500,000 projects so that they can be posted as quickly and as efficiently as possible.
2. How to increase the consistency of project vetting across different volunteers to improve the experience for teachers.
3. How to focus volunteer time on the applications that need the most assistance.

Building a model to predict a proposal's approval accomplishes all three of these goals in a quantifiable, measurable and replicable way.

Metrics

Metrics used to evaluate is AOC (area under the ROC curve between the predicted probability and the observed target). ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection. The false positive rate is also known as the fall-out or probability of false alarm.

II. Analysis

Data Exploration

The dataset can be downloaded from kaggle.com. The dataset contains information from teacher's project applications to DonorsChoose.org including teacher attributes, school attributes, and the project proposals including application essays.

- train.csv - The training set contains the label indicating whether the proposal was approved
- test.csv - The test set contains information without a label indicating the proposal was approved.
- resources.csv – The resources set contains the description, quantity, and price of resources requested as part of the dataset.
- sample_submission.csv – a sample submission file in the correct format

Data fields

test.csv and train.csv:

- id - unique id of the project application -
- teacher_id - id of the teacher submitting the application
- teacher_prefix - the title of the teacher's name (Ms., Mr., etc.)
- school_state - US state of the teacher's school
- project_submitted_datetime - application submission timestamp
- project_grade_category - school grade levels (PreK-2, 3-5, 6-8, and 9-12)
- project_subject_categories - category of the project (e.g., "Music & The Arts")
- project_subject_subcategories - sub-category of the project (e.g., "Visual Arts")
- project_title - the title of the project
- project_essay_1 - first essay*
- project_essay_2 - second essay*
- project_essay_3 - third essay*
- project_essay_4 - fourth essay*
- project_resource_summary - summary of the resources needed for the project
- teacher_number_of_previously_posted_projects - the number of previously posted applications by the submitting teacher

- project_is_approved - whether DonorsChoose proposal was accepted (0="rejected", 1="accepted"); train.csv only

Note: Prior to May 17, 2016, the prompts for the essays were as follows:

- project_essay_1: "Introduce us to your classroom"
- project_essay_2: "Tell us more about your students"
- project_essay_3: "Describe how your students will use the materials you're requesting"
- project_essay_4: "Close by sharing why your project will make a difference"

Starting on May 17, 2016, the number of essays was reduced from 4 to 2, and the prompts for the first 2 essays were changed to the following:

- project_essay_1: "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
- project_essay_2: "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

For all projects with project_submitted_datetime of 2016-05-17 and later, the values of project_essay_3 and project_essay_4 will be NaN.

The training dataset could be split into 80 train and 20 validation sets.

The data fields that might be useful in predicting the project_is_approved (label) are:

- teacher_prefix,
- school_state,
- project_submitted_datetime,
- project_subject_categories,
- project_grade_categories,
- project_subject_subcategories,
- project_title,
- teacher_number_of_previously_posted_project

resources.csv:

Proposals also include resources requested. Each project may include multiple requested resources. Each row in resources.csv corresponds to a resource, so multiple rows may tie to the same project by id.

- id - unique id of the project application; joins with test.csv. and train.csv on id
- description - description of the resource requested
- quantity - quantity of resource requested
- price - the price of resource requested

The first part of data exploration is to understand data. In this, we use statistical methods to summarize the dataset and relationship between the variable in the dataset.

Columns	Size	Type	Feature Type
id	182080	object	categorical
teacher_id	182080	object	categorical
teacher_prefix	182076	object	categorical
school_state	182080	object	categorical
project_submitted_datetime	182080	object	categorical(date time)
project_grade_category	182080	object	categorical
project_subject_categories	182080	object	categorical
project_subject_subcategories	182080	object	categorical
project_title	182080	object	categorical(text)
project_essay_1	182080	object	categorical(text)
project_essay_2	182080	object	categorical(text)
project_essay_3	6374	object	categorical(text)
project_essay_4	6374	object	categorical(text)
project_resource_summary	182080	object	categorical(text)
teacher_number_of_previously_posted_projects	182080	Int64	numerical

Columns	Size	Type	Feature Type
id	1541272	object	categorical
description	1540980	object	categorical(text)
quantity	1541272	int64	numerical
price	1541272	float64	numerical

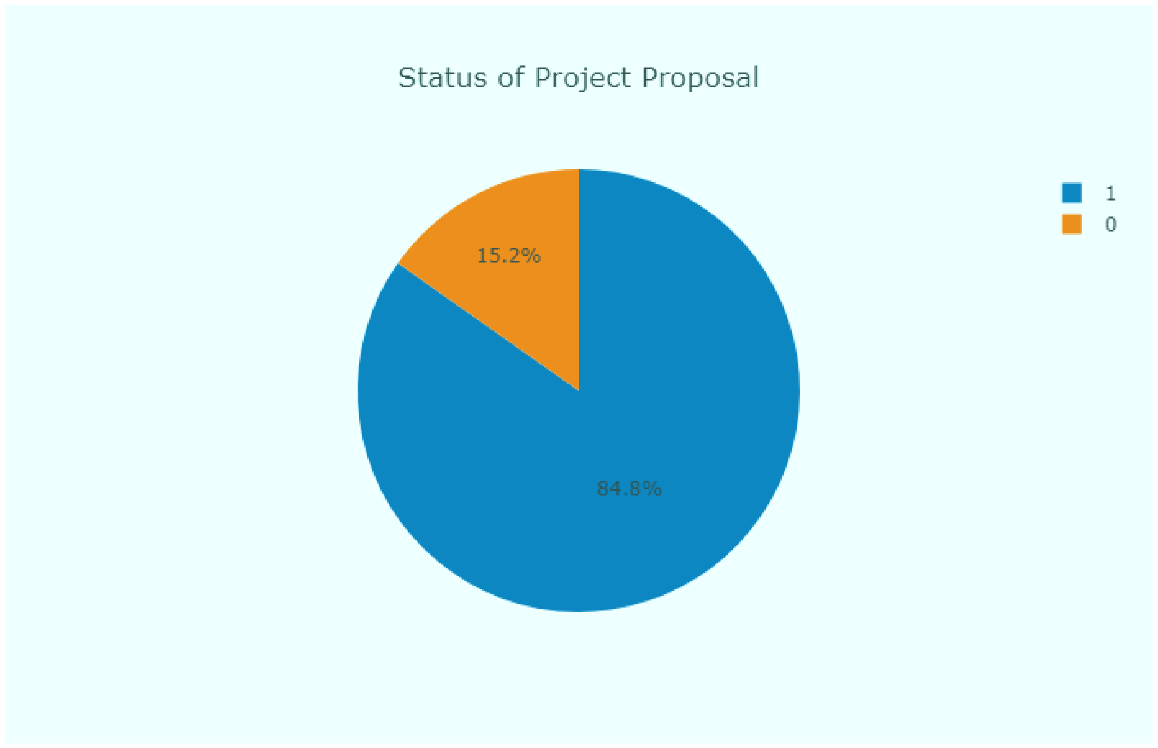
- Size of all the three tables.
  - Training data has 182,080 rows and 15 columns
  - Test data has 78,035 rows and 15 columns
  - Resource data has 1,541,278 rows and 4 columns
- Type of all the three tables
  - Type of training data
    - All are object except for teacher_number_of_previously_posted_projects is int64
  - Type of test data – object
  - Resource data
    - Object – id, description

        - Int64 – quantity
        - Float64 – price
- As ‘id’ column of resource data is unique and matches with that of train and test data set, the resource data set is merged with train and test dataset.
- Further analyzing the features
    - **Numerical features:** teacher_number_of_previously_posted_projects
    - **Categorical features**: id, teacher_id, teacher_prefix, school_state, project_grade_categories, project_subject_categories,
    - **Text features**: project title, project_essay_1, project_essay_2, project_essay_3, project_essay_4, project_resource_summary
    - **Date Time features:** project_submitted_datetime
- The target variable (‘project_is_approved’) for this case is a binary variable, which indicates if the project is approved to be hosted or not.

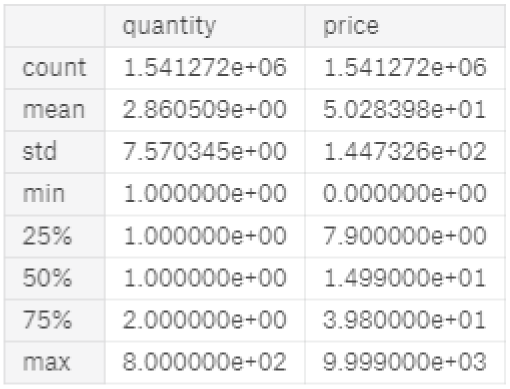
Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature of the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

1. In the training period feature ‘project_is_approved’, it is observed that there is 84% approval rate.



2. Based on the numerical data, on an average 'teacher_number_of_previously_posted_projects' the average number of projects submitted are 12 and based on the above graph the percentage of projects approved is large.



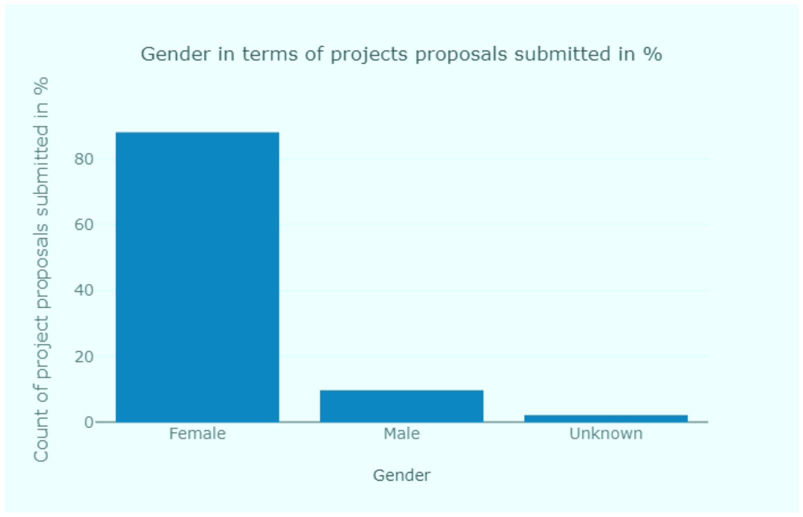
	quantity	price
count	1.541272e+06	1.541272e+06
mean	2.860509e+00	5.028398e+01
std	7.570345e+00	1.447326e+02
min	1.000000e+00	0.000000e+00
25%	1.000000e+00	7.900000e+00
50%	1.000000e+00	1.499000e+01
75%	2.000000e+00	3.980000e+01
max	8.000000e+02	9.999000e+03

Further evaluation of feature 'teacher_number_of_previously_posted_projects' to see if experience has any effect on the approval of submitted projects. The statistical calculation based on the data are

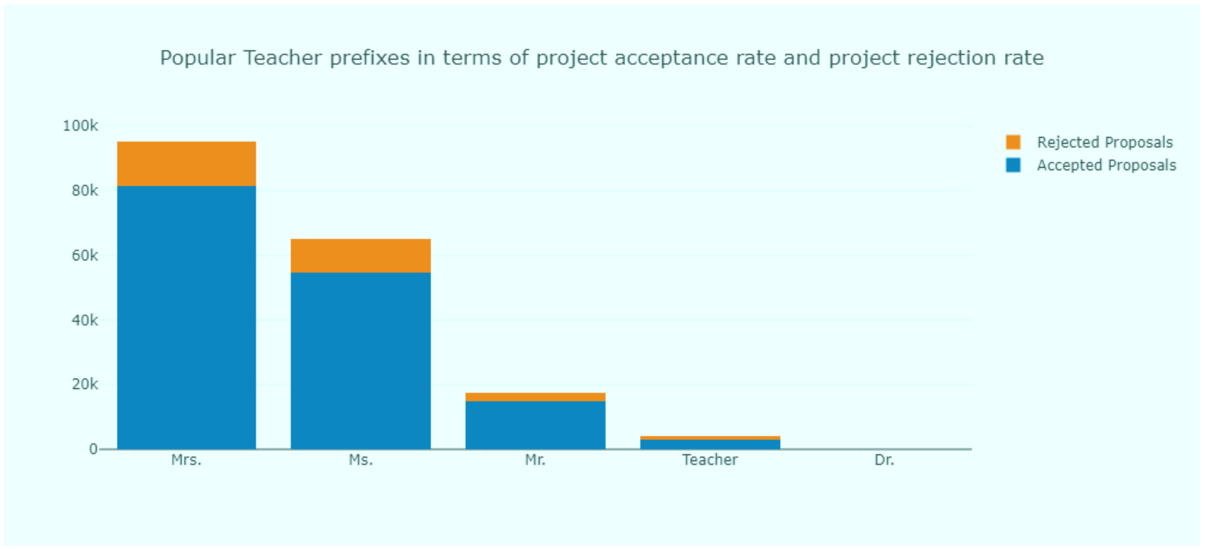
- Percentage of teachers with no projects: 27.497
- Percentage of teachers with one project: 14.68
- Percentage of teachers with more than one project: 57.821

Based on the data, it is unclear that the feature 'teacher_number_of_previously_posted_projects' might not be the only factor influencing the success rate of the project. As the dataset contains 84% percentage of approved projects, there might be a chance of dataset missing some of the projects. The projects submitted by the first and fourth teacher are 26 and 16 respectively while the projects present in the data-set are 16 and 10 respectively.

3. The feature 'teacher_prefix' could be analyzed to see if there is any effect of gender on the approval of the project. The statistical data of the gender are
    - Mrs + Ms – 95405 + 65066 = 130471
    - Mr – 17667
    - Teacher + Dr + NaN – 3912 + 26 + 4 = 3938



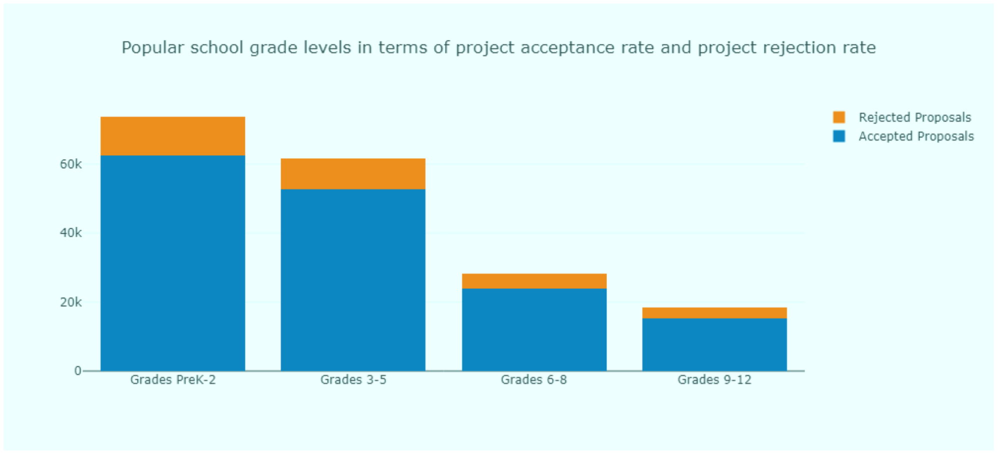
This shows there number of females submitted projects are the highest. Plotting popular teacher prefixes in terms of project acceptance rate and project rejection rate.



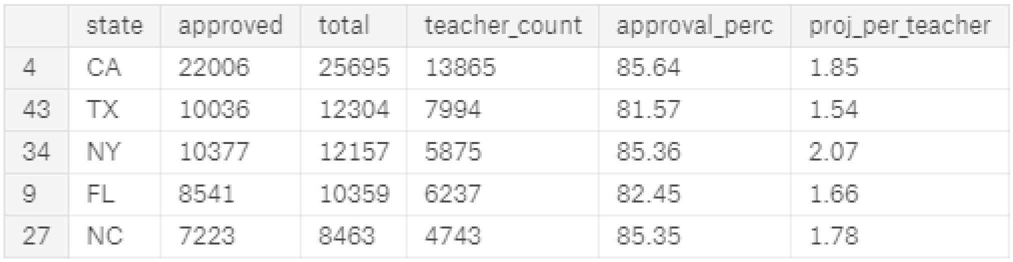
4. The feature 'project_grade_category' could be analyzed to see if there is any relation between the age limit of the kids to that of approval of the project. There are totally four grades

Grade	Counts
PreK-2	73890
3 - 5	61682
6 - 8	28197
9 - 12	18311

The approval of the projects based on grade category is 3-5, 6-8, 9-12, PreK-2.

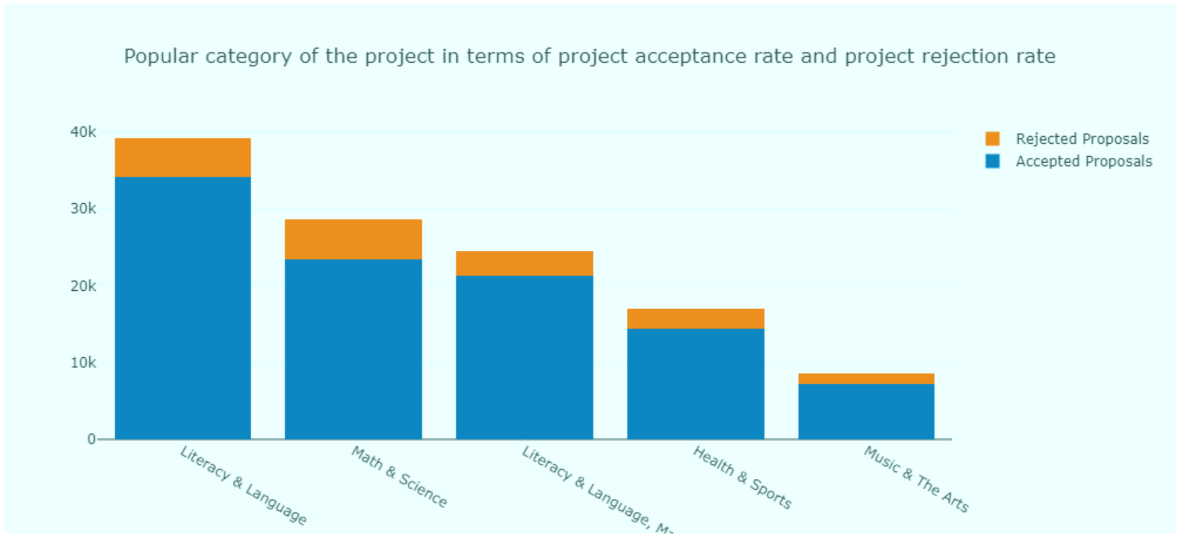


5. The feature ‘school_state’ could be analyzed to see if there is any relation between the approval of the project.

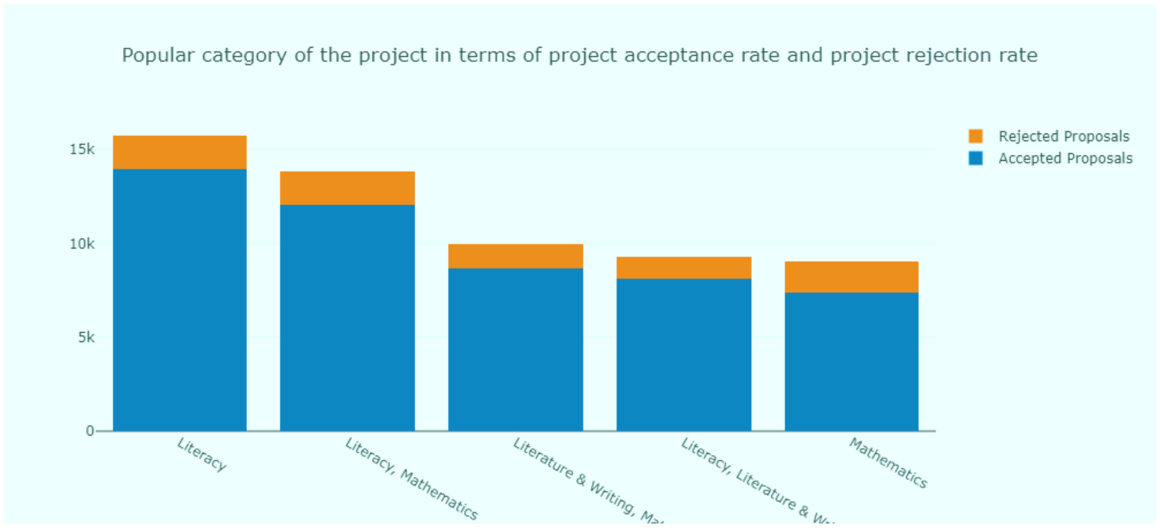


	state	approved	total	teacher_count	approval_perc	proj_per_teacher
4	CA	22006	25695	13865	85.64	1.85
43	TX	10036	12304	7994	81.57	1.54
34	NY	10377	12157	5875	85.36	2.07
9	FL	8541	10359	6237	82.45	1.66
27	NC	7223	8463	4743	85.35	1.78

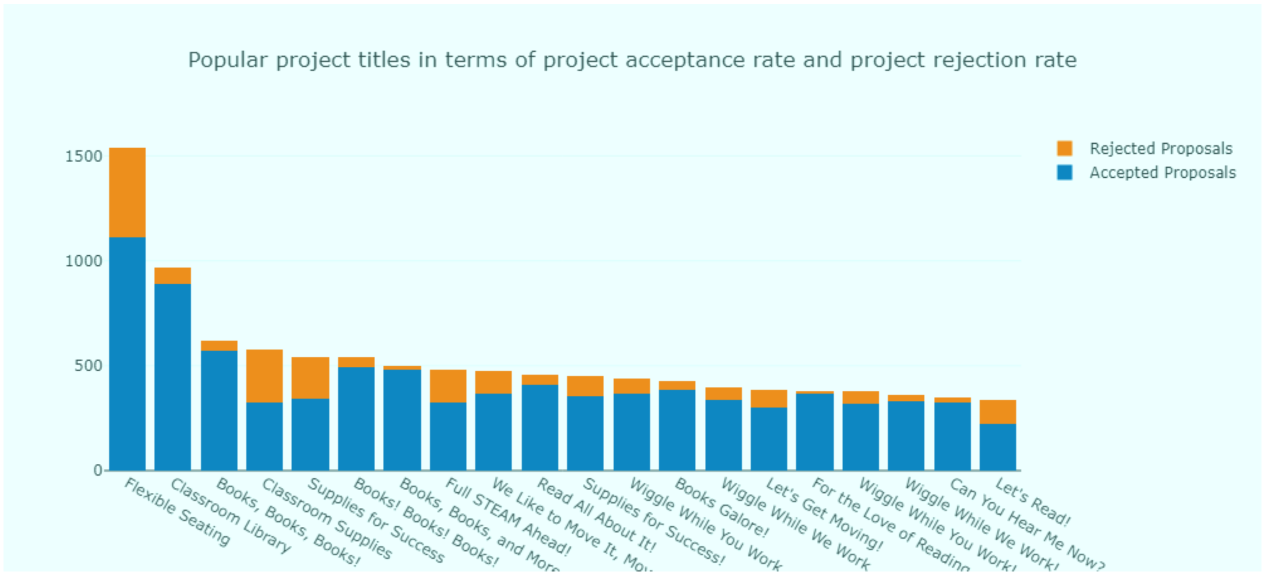
6. The feature ‘project_subject_categories’ could be analyzed to see if there is any relation between the approval of the project. There are 51 unique subject categories. ‘Literacy & Language’ is the top subject category.

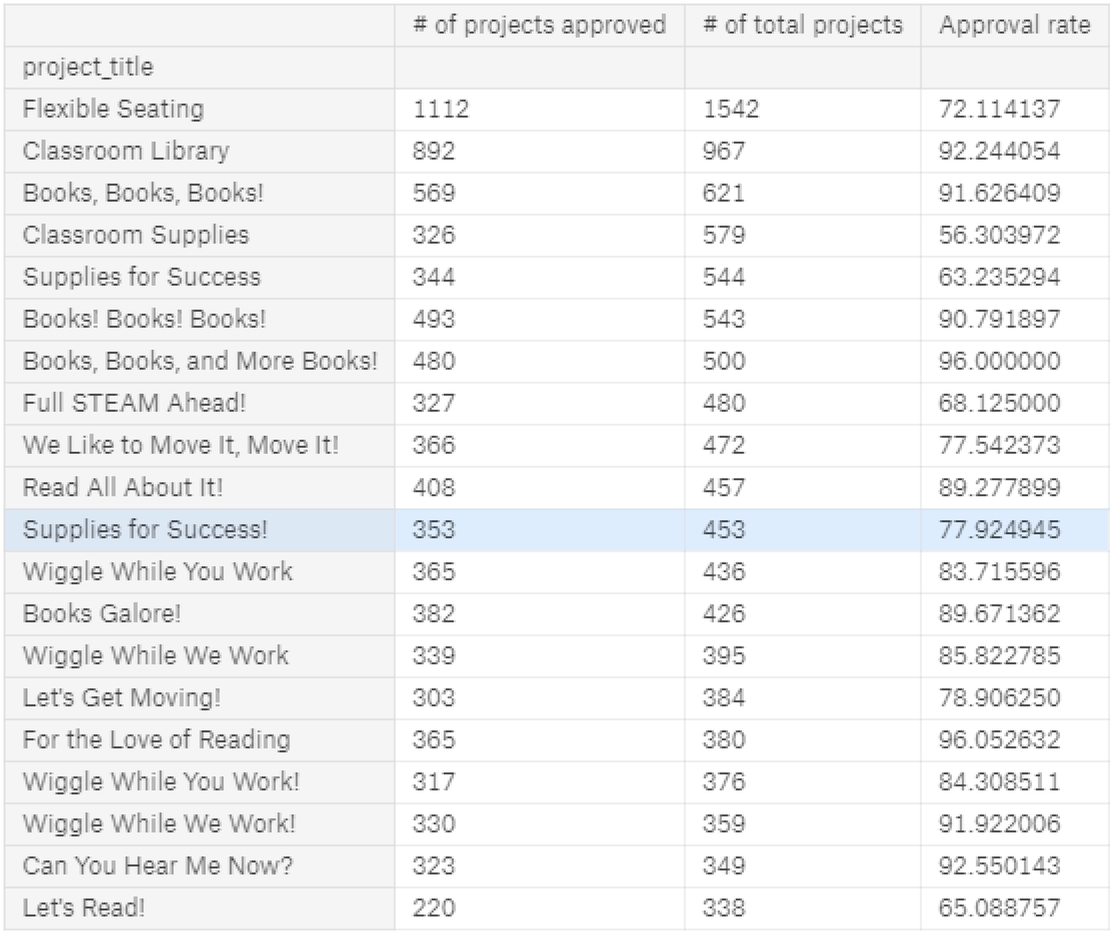


7. The feature ‘project_subject_subcategories’ could be analyzed to see if there is any relation between the approval of the project. There are 407 unique subject subcategories. ‘Literacy’ is the top in subject subcategories.



8. The feature ‘project_title’ could be analyzed to see if there is any relation between approvals of the project. Flexible Seating is top, but when the top 20 records are analyzed Wiggle While You Work is the most commonly used phrase with slight changes. There is also 90% approved with projects with that title.

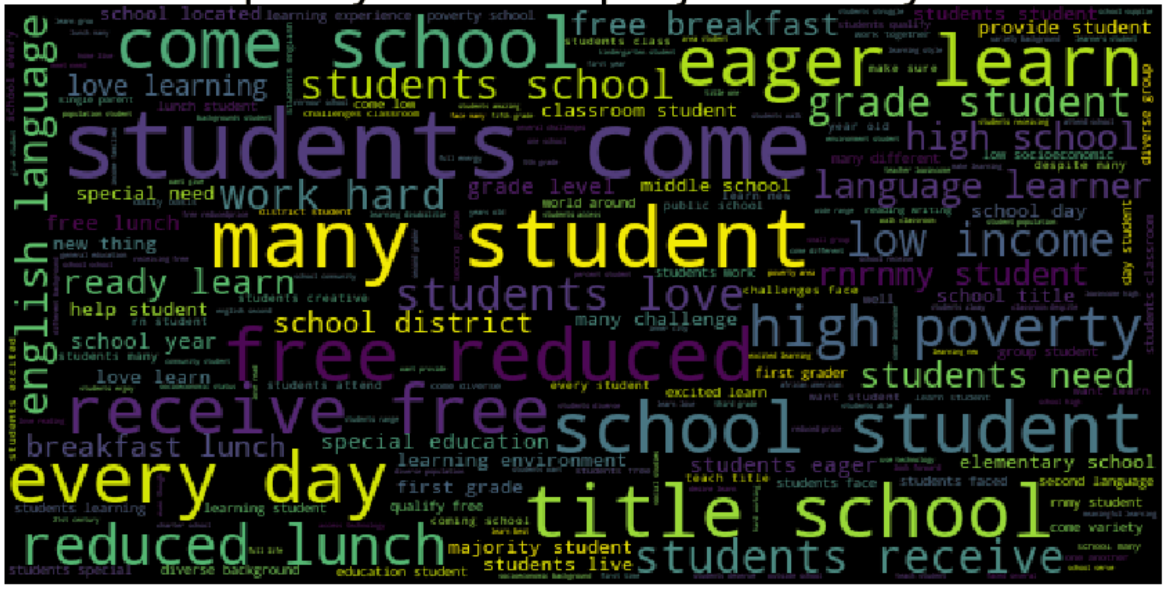




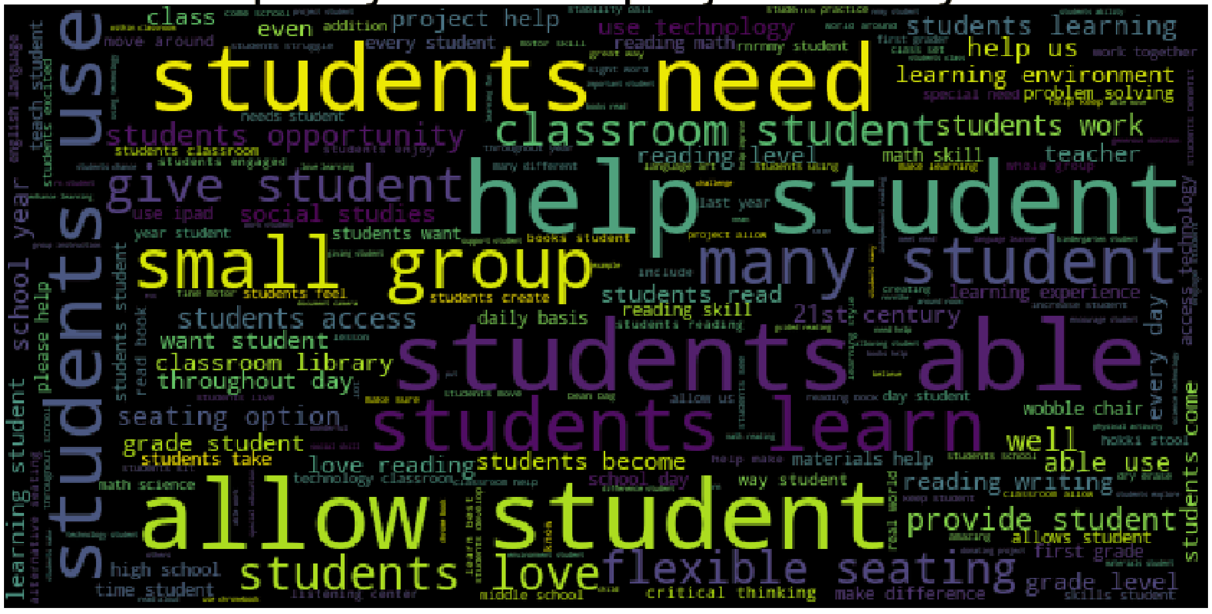
	# of projects approved	# of total projects	Approval rate
project_title			
Flexible Seating	1112	1542	72.114137
Classroom Library	892	967	92.244054
Books, Books, Books!	569	621	91.626409
Classroom Supplies	326	579	56.303972
Supplies for Success	344	544	63.235294
Books! Books! Books!	493	543	90.791897
Books, Books, and More Books!	480	500	96.000000
Full STEAM Ahead!	327	480	68.125000
We Like to Move It, Move It!	366	472	77.542373
Read All About It!	408	457	89.277899
Supplies for Success!	353	453	77.924945
Wiggle While You Work	365	436	83.715596
Books Galore!	382	426	89.671362
Wiggle While We Work	339	395	85.822785
Let's Get Moving!	303	384	78.906250
For the Love of Reading	365	380	96.052632
Wiggle While You Work!	317	376	84.308511
Wiggle While We Work!	330	359	91.922006
Can You Hear Me Now?	323	349	92.550143
Let's Read!	220	338	65.088757

9. The features ‘project_essay_1’, ‘project_essay_2’, ‘project_essay_3’, ‘project_essay_4’ is a text feature. Only 6374 columns of ‘project_essay_3’ and project_essay_4’ are full. The words which have been extensively used in ‘project_essay_1’ and project_essay_2’ are

Top Keywords in project_essay_1

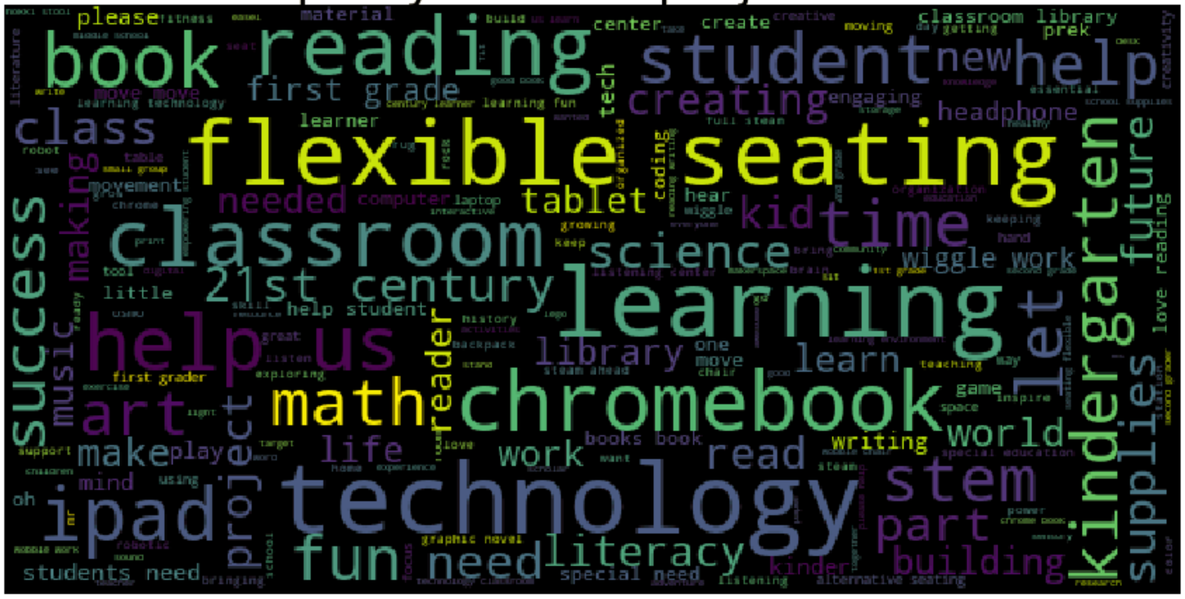


Top Keywords in project_essay_2

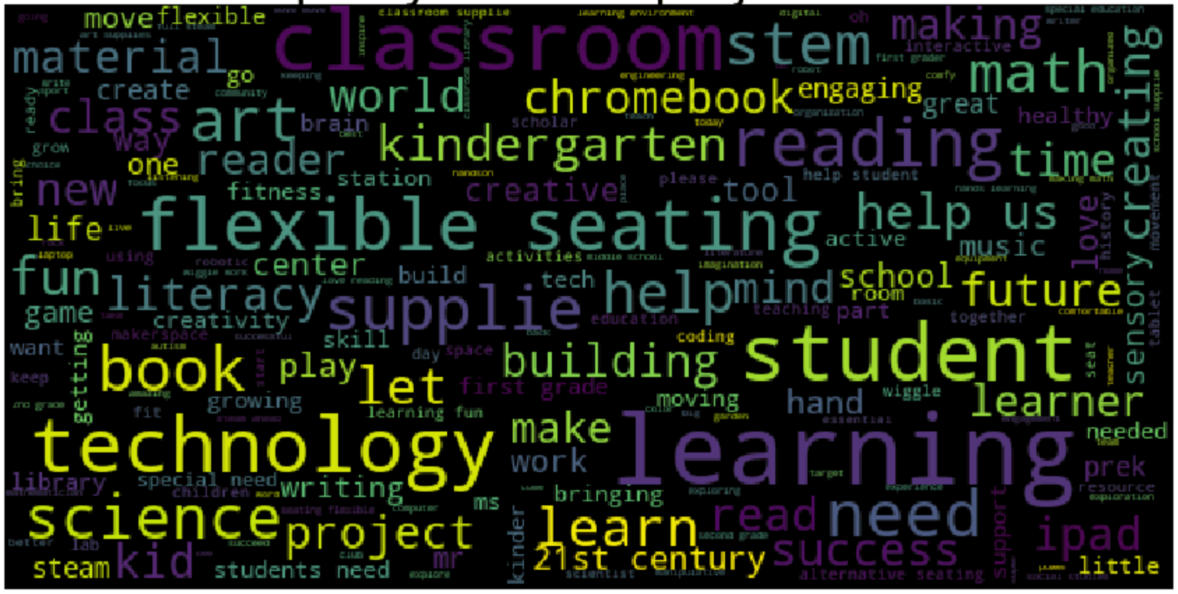


10. The feature ‘project_title’ is a text feature and the approved and rejected projects top keywords could be analyzed.

Top Keywords in project title

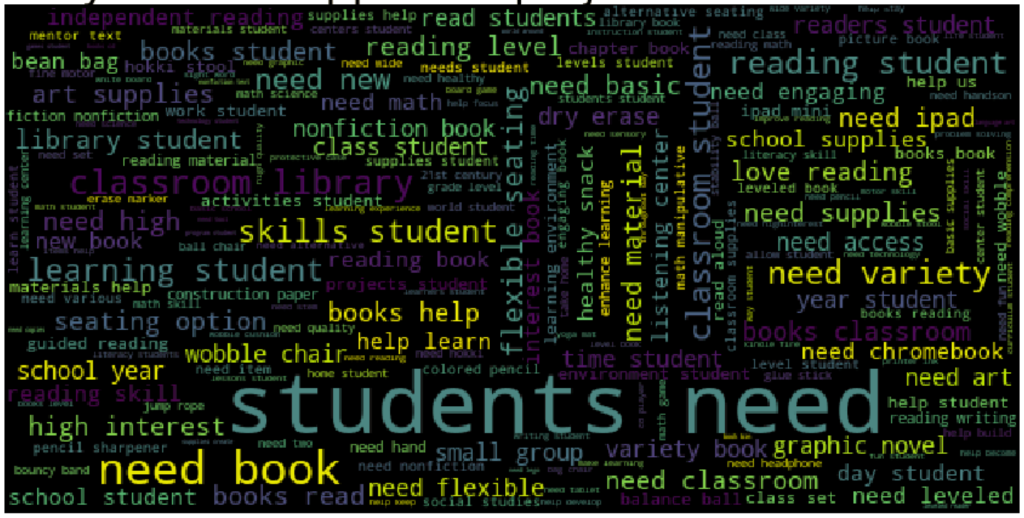


Top Keywords in project title

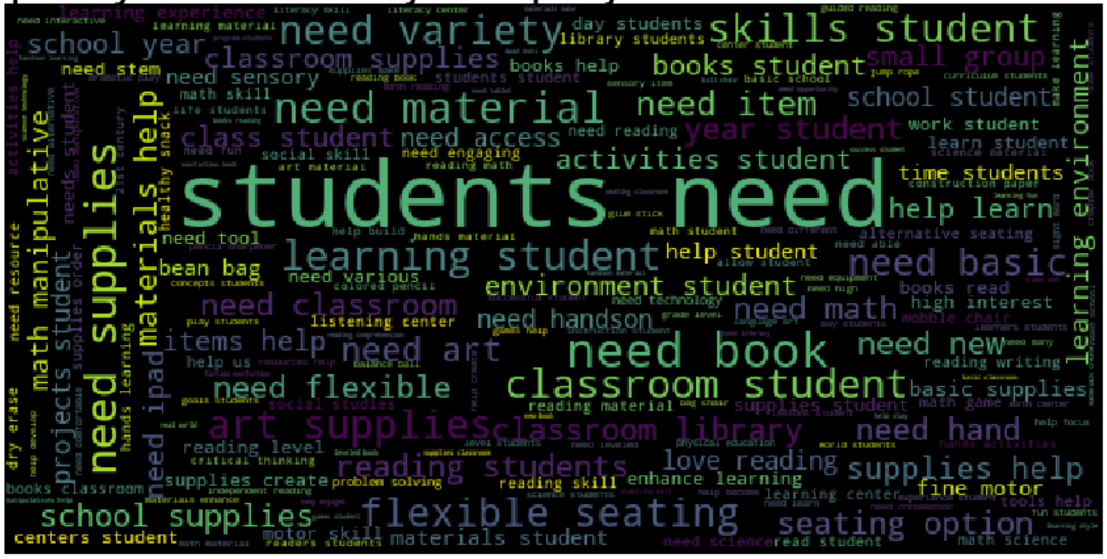


11. The feature ‘project_resource_summary’ is a text feature and the top keywords could be analyzed.

Top Keywords in approved project resource summary

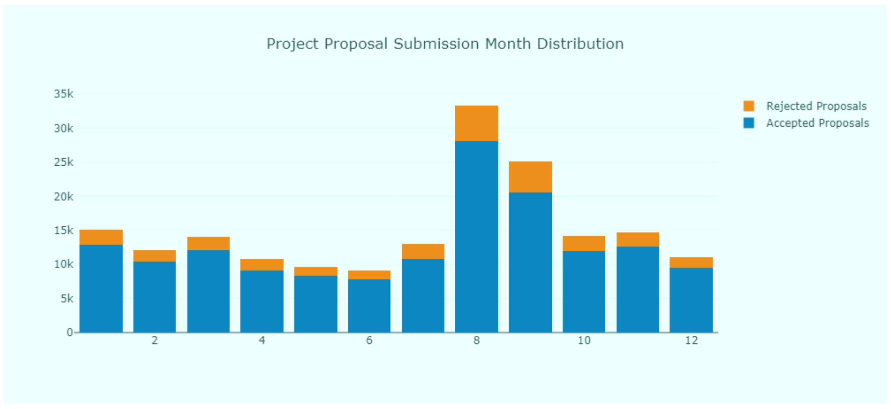


Top Keywords in Rejects project resource summary

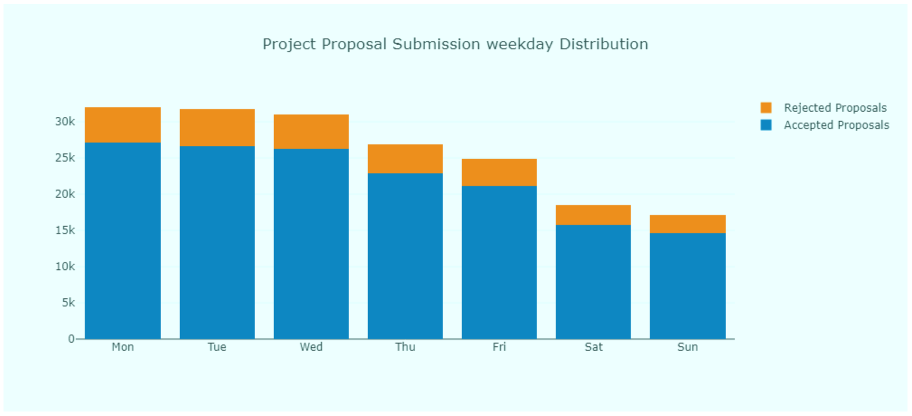


12. The feature ‘project_submitted_datetime’ is date time features and this feature could be broken down to analyze

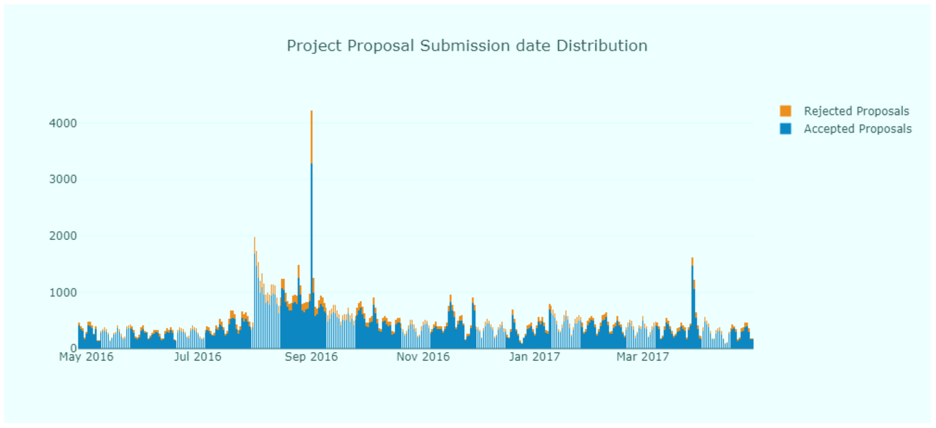
    - Projects submitted month wise



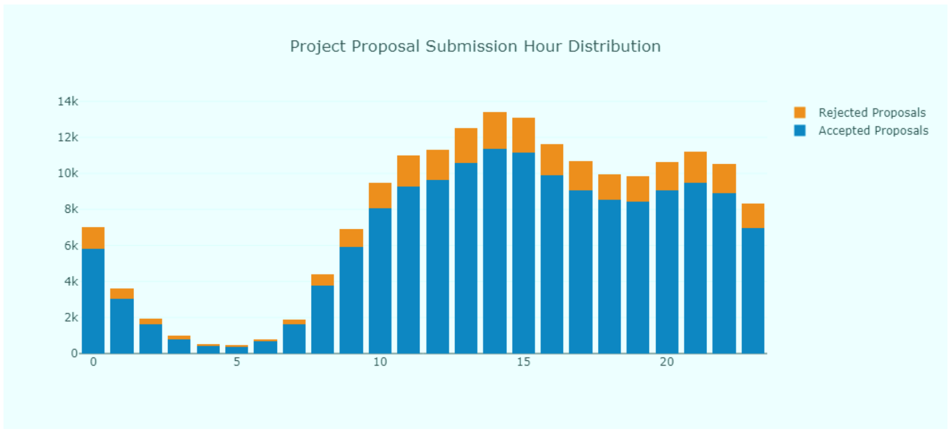
- Projects submitted week day wise



- Projects submitted date wise



- Projects submitted hour wise



Algorithms and Techniques

The most common solution to such problems is the method of classification. Some of the classification methods are:

a) Linear Classifiers: Logistic Regression, Naïve Bayes Classifier

b) Support Vector Machines

c) Decision Trees

d) Boosted Trees

e) Random Forest

f) Neural Networks

g) Nearest Neighbor

Considering all of these the best method chosen for predicting the status of application is Decision Trees. Decision trees builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree is called as the root node, which is the best predictor. Decision Trees is a type of supervised learning algorithm, which works for both categorical and continuous input and output variables. They can handle numerical and categorical data, missing data. Although the decision trees work, ensemble methods are methods to boost tree models. Ensemble methods involve a group of predictive models to achieve a better accuracy and model stability. The models used were Light GBM, XGBoost are seen to be extremely effective in such areas.

Light GBM grows trees vertically. The leaf with max delta loss is chosen to grow. Light GBM can handle the large size of data and takes lower memory to run, focuses on accuracy of results, supports GPU learning. The complicated thing is parameter tuning.

XGBoost is extreme gradient boosting. XGBoost is similar to gradient boosting implementation. It supports various objective functions, including regression, classification, and ranking. Since it is very high in predictive power but relatively slow with implementation. XGBoost only works with numeric vectors. Therefore, you need to convert all other forms of data into numeric data. As the dataset contains several columns with texts. We will be using TF-IDF Vectorization is based on the frequency method but it is different to the count vectorization in the sense that it takes into account not just the occurrence of a word in a single document but in the entire collection.

Benchmark

At the present DonorChoose.org manually screens for applications and also the model mentioned in the tutorial "Getting Started with the DonorChoose.org dataset" by Sanders Kleinfeld uses a linear classification model to train. The training and validation log losses are the results which are the outputs. Later, AUC (area under the curve), which is the metric for assessing the accuracy of prediction calculated. The model achieves the AUC score of 0.56

A classification model, which will screen for application quickly than the manual process would solve the problems of DonorChoose.org

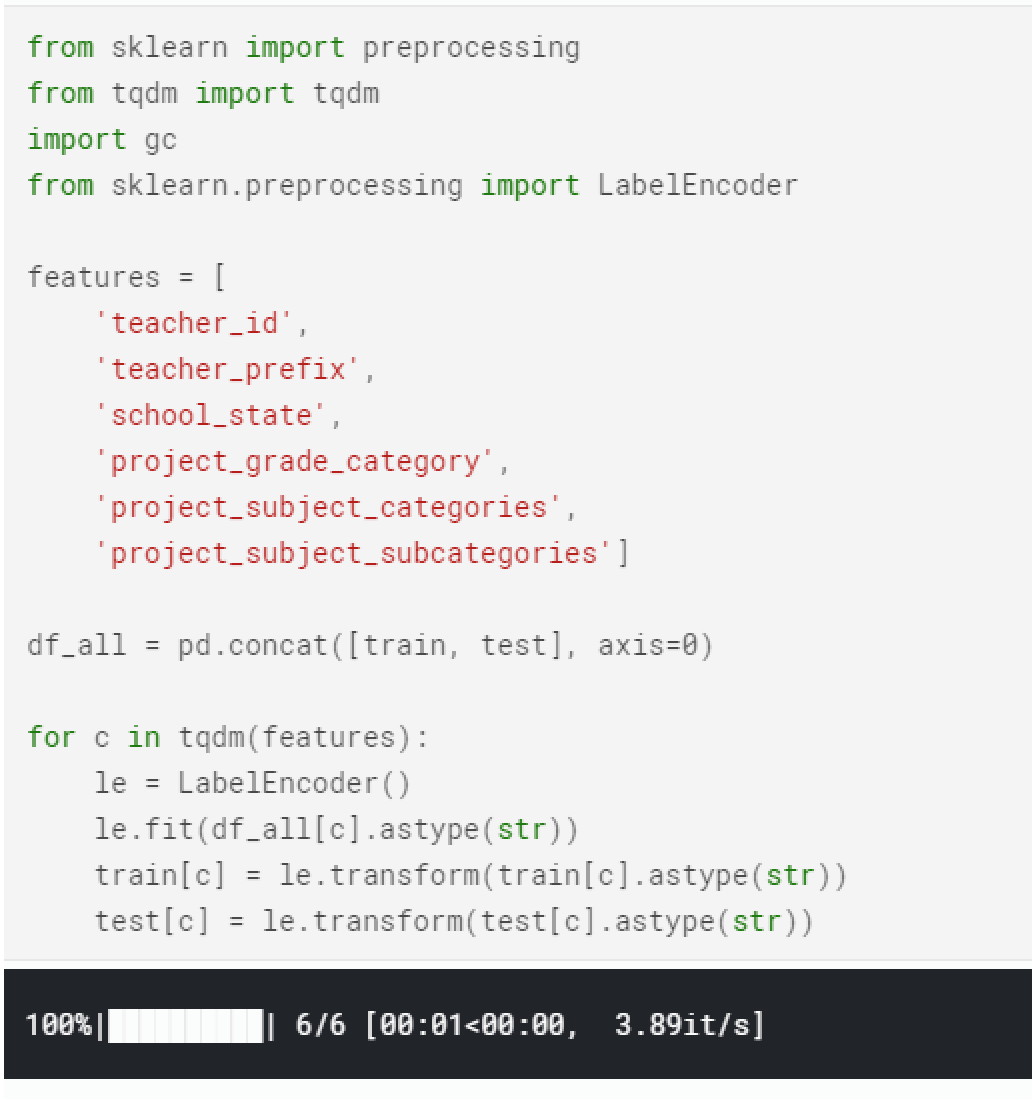
III. Methodology

Data Preprocessing

Three data files, train.csv, test.csv and resources.csv are given and can be downloaded from the competition page. The data has a size of 195 MB and has information from teacher's project applications submitted to DonorChoose.org. This included teacher attributes, school attributes, and the project proposals including application essays. The objective is to predict whether a DonorChoose.org project proposal submitted by a teacher will be approved.

The dataset have wide range of null values, the null values are filled with 'unk' to clean up the data.

As discussed above, feature engineering is necessary and the first step in feature engineering is to turn the labels or categorical values, into integers. Generally, XGBoost does better with label encoding that one hot encoding. In classification problems, we have many labels. These labels are words, number. The machine learning function expect number and for this, we need to convert labels to numbers. A label encoding refers to the process of transforming the word labels into numerical form.



```
from sklearn import preprocessing
from tqdm import tqdm
import gc
from sklearn.preprocessing import LabelEncoder

features = [
    'teacher_id',
    'teacher_prefix',
    'school_state',
    'project_grade_category',
    'project_subject_categories',
    'project_subject_subcategories']

df_all = pd.concat([train, test], axis=0)

for c in tqdm(features):
    le = LabelEncoder()
    le.fit(df_all[c].astype(str))
    train[c] = le.transform(train[c].astype(str))
    test[c] = le.transform(test[c].astype(str))
```

```
100%|██████████| 6/6 [00:01<00:00,  3.89it/s]
```

Feature Engineering from train.csv and test.csv:

- **‘project_submitted_datetime’**: We will be creating features like year, month, hour, day, day of the week from the ‘project_submitted_datetime’ feature.
- **‘project_title’** : We will be creating features like project title length
- **‘project_essay_1’ ‘project_essay_2’:** We will be creating features like project essay length.
- **‘project_resource_summary’:** We will be creating features like project resource summary length.

- ‘project_title’, ‘project_essay_1’, ‘project_essay_2’, ‘project_essay_3’, ‘project_essay_4’, ‘project_resource_summary’ are joined together.

Feature Engineering from resource.csv:

- **‘quantity’ and ‘price’:** A product of both quantity and price can be calculated for each id.
- **‘resource_total’:** The sum and mean are calculated for resources_total.
- **‘quantity’:** The sum and count are calculated for quantity.

Implementation

The implementation is done in two steps

- Learn about the model
- Train the model

List of algorithms tested

- Light GBM
- XGBoost

Refinement

TFID Vectorizer: TFID Vectorizer is used to transform the concatenated text into number. Rf-idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), the second terms is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

- TF(t) = (Number of times term t appears in a document)/ (Total number of terms in the document)
- IDF(t) = log_e(Total number of documents / Number of documents with terms t in it)

First, all the text columns are transformed into numbers. Then it is cleaned up a bit and lemmatized. Lemmatization is a process of grouping together the infected forms of a words so they can be analyzed as a single item.

Then text vectorization or TFIDF is performed. We fit on train and test individually.

Light GBM: A gradient boosting framework that uses tree based learning algorithm. This is a two steps process:

1. Learning about the model
2. Training the model

For this, we use cross-validation to avoid overfitting. For this, a process called Repeated K-Fold which repeats K-Fold n times. This is generally used when one requires to run KFold n times, and this produces different splits in each repetition. The parameters for this are

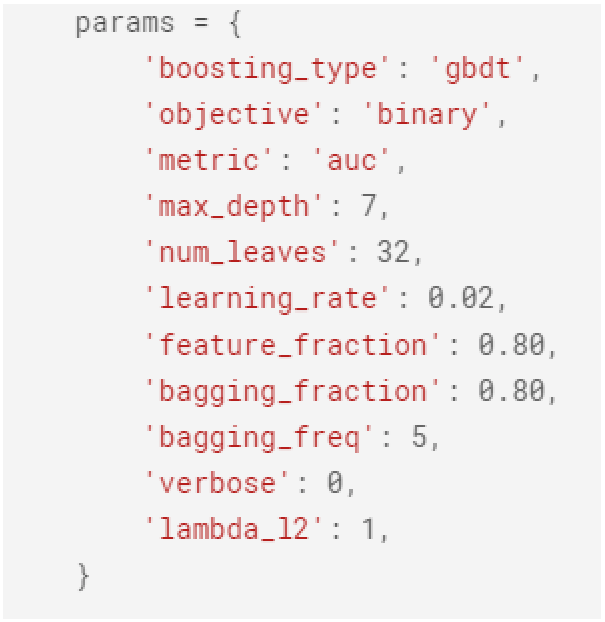
- n_splits = 5, n_repeats = 1, random_state = 28 i.e., seed = 28

We then split into train, test and validation with parameters

- test_size = 0.20, random_state = random.seed(28)

The model was built using Light GBM using the parameters

- boosting_type: ‘gbdt’, objective: ‘binary’, metric: ‘auc’, max_depth: 7, num_leaves: 32, learning_rate: 0.82, feature_fraction: 0.80, bagging_fraction: 0.80, bagging_freq: 5, verbose: 0, lambda_12: 1



```
params = {
    'boosting_type': 'gbdt',
    'objective': 'binary',
    'metric': 'auc',
    'max_depth': 7,
    'num_leaves': 32,
    'learning_rate': 0.02,
    'feature_fraction': 0.80,
    'bagging_fraction': 0.80,
    'bagging_freq': 5,
    'verbose': 0,
    'lambda_l2': 1,
}
```

This Light GBM model got the scores:

- AUC: 0.776310 +/ 0.003589

XGBoost: A gradient boosted decision trees designed for speed and performance. The implementation is used in the similar way as that of Light GBM.

We use KFold instead of Repeated K-Fold, then split the dataset into train, test and validation set and then perform XGBoost on it. We try to use the similar parameters for this

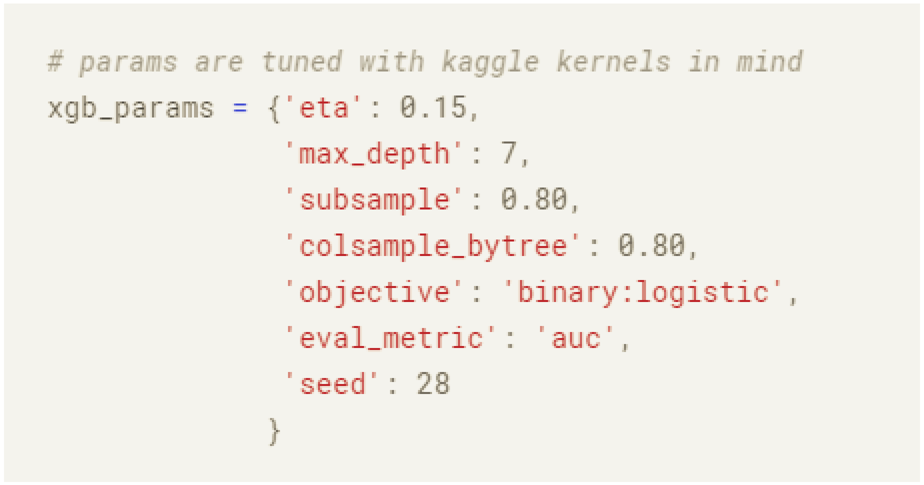
- n_splits = 5, random_state = 28 i.e., seed = 28

For splitting, the parameters used are

- test_size = 0.20, random_state = random.seed(28)

The model was built using XGBoost using the parameters

- eta: 0.15, max_depth: 7, subsample: 0.80, colsample_bytree: 0.80, objective: binary:logistic, eval_metric: auc, seed: 28



```
# params are tuned with kaggle kernels in mind
xgb_params = {'eta': 0.15,
              'max_depth': 7,
              'subsample': 0.80,
              'colsample_bytree': 0.80,
              'objective': 'binary:logistic',
              'eval_metric': 'auc',
              'seed': 28
             }
```

The XGBoost model got scores:

- AUC: 0.764542

Both the models perform well than the benchmark model.

IV. Results

This Light GBM model got the scores:

- AUC: 0.776310 +/ 0.003589

The XGBoost model got scores:

- AUC: 0.764542

Both the models perform well than the benchmark model.

Model Evaluation and Validation

Metrics used to evaluate is AOC (area under the ROC curve between the predicted probability and the observed target). ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also knows as sensitivity, recall or probability of detection. The false positive rate is also known as the fall-out or probability of false alarm.

The model was built using XGBoost using the parameters

- eta: 0.15, max_depth: 7, subsample: 0.80, colsample_bytree: 0.80, objective: binary:logistic, eval_metric: auc, seed: 28

The model was built using Light GBM using the parameters

- boosting_type: ‘gbdt’, objective: ‘binary’, metric: ‘auc’, max_depth: 7, num_leaves: 32, learning_rate: 0.82, feature_fraction: 0.80, bagging_fraction: 0.80, bagging_freq: 5, verbose: 0, lambda_12: 1

Justification

At the present DonorChoose.org manually screens for applications and also the model mentioned in the tutorial “Getting Started with the DonorsChoose.com Data set” by Sanders Kleinfeld uses linear classification model to train. The training and validation log losses are the results which are the outputs. Later, AUC (area under the curve), which is the metric for assessing the accuracy of prediction calculated. The model achieves AUC score of 0.56.

A classification model, which will screen for applications quickly than the manual process would solve the problems of DonorChoose.org

This Light GBM model got the scores:

- AUC: 0.776310 +/ 0.003589

The XGBoost model got scores:

- AUC: 0.764542

V. Conclusion

Free-Form Visualization

- Out of the 50 states, California has the highest number of project proposals submitted.
- Out of the four school grades, Grades Prek-2 is approximately 41%.
- Out of the 51 project categories, Literacy and Language is 27%.

- Out of 1,82,020 project subcategories, Literacy is 16%.
- Out of 1,82,020 project titles, “wiggle while you work” is the most seen.
- Females have more count which is approx.. 88% than male.
- September month has the second highest number of proposals. The number of proposals decreases as we move towards the end of the week.

Reflection

The whole project was well-defined classification problem. Exploratory data analysis was the most time consuming part. It was fun to explore all the features and there dependency with the approval of the project. There were four different types of features; date time, text, numerical, categorical. Analyzing is the text features was one of the interest part in the project. Some of the features have null values and these null values are filled with the string ‘unk’. The date time and text features are engineered like date time feature is broken down into hour, day, month, year, day of week and text features are broken down into length of the text. There is also another feature engineered to understand the data set more like ‘total’ and based on the total, sum, mean, count are created.

The next biggest challenge was figuring out what algorithm to use. The algorithms i.e., Decision trees, Naïve Bayes, Neural Networks, and Logistic Regression all seemed to work. However, the AUC score varied.

The parameters were not modified as the initial run got a better score than the benchmark mode.

Improvement

- Tune XGB, LGBM better
- Explore other models
- Use Eli5
- Explore state and time variables
- Explore topic modeling and text clustering
- Explore vector based features.