DonorsChoose

DonorsChoose.org receives hundreds of thousands of project proposals each year for classroom projects in need of funding. Right now, a large number of volunteers is needed to manually screen each submission before it's approved to be posted on the DonorsChoose.org website.

Next year, DonorsChoose.org expects to receive close to 500,000 project proposals. As a result, there are three main problems they need to solve:

- 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
- How to increase the consistency of project vetting across different volunteers to improve the experience for teachers
- How to focus volunteer time on the applications that need the most assistance

The goal of the competition is to predict whether or not a DonorsChoose.org project proposal submitted by a teacher will be approved, using the text of project descriptions as well as additional metadata about the project, teacher, and school. DonorsChoose.org can then use this information to identify projects most likely to need further review before approval.

About the DonorsChoose Data Set

The `train.csv` data set provided by DonorsChoose contains the following features:

Feature	Description
`project_id`	A unique identifier for the proposed project. **Example:** `p036502`
`project_title`	Title of the project. **Examples:** • `Art Will Make You Happy!` • `First Grade Fun`
`project_grade_category`	Grade level of students for which the project is targeted. One of the following enumerated values: • `Grades PreK-2` • `Grades 3-5` • `Grades 6-8` • `Grades 9-12`
`project_subject_categories`	One or more (comma-separated) subject categories for the project from the following enumerated list of values: • `Applied Learning` • `Care & Hunger` • `Health & Sports` • `History & Civics` • `Literacy & Language` • `Math & Science` • `Music & The Arts` • `Special Needs` • `Warmth` **Examples:** • `Music & The Arts` • `Literacy & Language, Math & Science`
`school_state`	State where school is located (Two-letter U.S. postal code). **Example:** `WY`
`project_subject_subcategories`	One or more (comma-separated) subject subcategories for the project. **Examples:** • `Literacy` • `Literature & Writing, Social Sciences`
`project_resource_summary`	An explanation of the resources needed for the project. **Example:** • `My students need hands on literacy materials to manage sensory needs!`
`project_essay_1`	First application essay*
`project_essay_2`	Second application essay*
`project_essay_3`	Third application essay*

Feature	Description
`project_essay_4`	Fourth application essay
`project_submitted_datetime`	Datetime when project application was submitted. **Example:** `2016-04-28 12:43:56.245`
`teacher_id`	A unique identifier for the teacher of the proposed project. **Example:** `bdf8baa8fedef6bfeec7ae4ff1c15c56`
`teacher_prefix`	Teacher's title. One of the following enumerated values: • `nan` • `Dr.` • `Mr.` • `Mrs.` • `Ms.` • `Teacher.`
`teacher_number_of_previously_posted_projects`	Number of project applications previously submitted by the same teacher. **Example:** `2`

* See the section **Notes on the Essay Data** for more details about these features.

Additionally, the `resources.csv` data set provides more data about the resources required for each project. Each line in this file represents a resource required by a project:

Feature	Description
`id`	A `project_id` value from the `train.csv` file. **Example:** `p036502`
`description`	Desciption of the resource. **Example:** `Tenor Saxophone Reeds, Box of 25`
`quantity`	Quantity of the resource required. **Example:** `3`
`price`	Price of the resource required. **Example:** `9.95`

Note: Many projects require multiple resources. The `id` value corresponds to a `project_id` in train.csv, so you use it as a key to retrieve all resources needed for a project:

The data set contains the following label (the value you will attempt to predict):

Label	Description
`project_is_approved`	A binary flag indicating whether DonorsChoose approved the project. A value of `0` indicates the project was not approved, and a value of `1` indicates the project was approved.

Notes on the Essay Data

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.

In [2]:

```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer

import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle

from tqdm import tqdm
import os

from plotly import plotly
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from collections import Counter
```

1.1 Reading Data

In [3]:

```
project_data = pd.read_csv('train_data.csv')
resource_data = pd.read_csv('resources.csv')
```

In [4]:

```
print("Number of data points in train data", project_data.shape)
print('-'*50)
print("The attributes of data :", project_data.columns.values)
```

```
Number of data points in train data (109248, 17)
--------------------------------------------------
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_prefix' 'school_state'
 'project_submitted_datetime' 'project_grade_category'
 'project_subject_categories' 'project_subject_subcategories'
 'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects' 'project_is_approved']
```

In [5]:

```
print("Number of data points in train data", resource_data.shape)
print(resource_data.columns.values)
resource_data.head(2)
```

```
Number of data points in train data (1541272, 4)
['id' 'description' 'quantity' 'price']
```

Out[5]:

	id	description	quantity	price
0	p233245	LC652 - Lakeshore Double-Space Mobile Drying Rack	1	149.00
1	p069063	Bouncy Bands for Desks (Blue support pipes)	3	14.95

In [6]:

```
project_data["project_is_approved"].value_counts()
```

Out[6]:

```
1    92706
0    16542
Name: project_is_approved, dtype: int64
```

1.2 preprocessing of `project_subject_categories`

In [7]:

```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python:
https://stackoverflow.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-python
cat_list = []
for i in catogories:
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Science", "Warmth", "Care & H
unger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "Math & Scienc
e"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to replace it with ''(i
.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"Math &
Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spaces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

project_data['clean_categories'] = cat_list
project_data.drop(['project_subject_categories'], axis=1, inplace=True)

from collections import Counter
my_counter = Counter()
for word in project_data['clean_categories'].values:
    my_counter.update(word.split())

cat_dict = dict(my_counter)
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))
```

1.3 preprocessing of `project_subject_subcategories`

In [8]:

```
sub_catogories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python:
https://stackoverflow.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-python

sub_cat_list = []
for i in sub_catogories:
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Science", "Warmth", "Care & H
unger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "Math & Scienc
e"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to replace it with ''(i
.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"Math &
```

```
Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the trailing spaces
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())

project_data['clean_subcategories'] = sub_cat_list
project_data.drop(['project_subject_subcategories'], axis=1, inplace=True)

# count of all the words in corpus python: https://stackoverflow.com/a/22898595/4084039
my_counter = Counter()
for word in project_data['clean_subcategories'].values:
    my_counter.update(word.split())

sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv: kv[1]))
```

1.2.7 Univariate Analysis: Text features (Project Essay's)

In [9]:

```
# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) +\
                        project_data["project_essay_2"].map(str) + \
                        project_data["project_essay_3"].map(str) + \
                        project_data["project_essay_4"].map(str)
```

1.2.8 Univariate Analysis: Cost per project

In [10]:

```
# we get the cost of the project using resource.csv file
resource_data.head(2)
```

Out[10]:

	id	description	quantity	price
0	p233245	LC652 - Lakeshore Double-Space Mobile Drying Rack	1	149.00
1	p069063	Bouncy Bands for Desks (Blue support pipes)	3	14.95

In [11]:

```
# https://stackoverflow.com/questions/22407798/how-to-reset-a-dataframes-indexes-for-all-groups-in
-one-step
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'}).reset_index()
price_data.head(2)
```

Out[11]:

	id	price	quantity
0	p000001	459.56	7
1	p000002	515.89	21

In [12]:

```
# join two dataframes in python:
project_data = pd.merge(project_data, price_data, on='id', how='left')

print(project_data)
```

```
        Unnamed: 0       id                        teacher_id teacher_prefix  \
0           160221  p253737  c90749f5d961ff158d4b4d1e7dc665fc           Mrs.
1           140945  p258326  897464ce9ddc600bced1151f324dd63a            Mr.
2            21895  p182444  3465aaf82da834c0582ebd0ef8040ca0            Ms.
3               45  p246581  f3cb9bffbba169bef1a77b243e620b60           Mrs.
```

```
4           172407  p104768  be1f7507a41f8479dc06f047086a39ec           Mrs.
5           141660  p154343  a50a390e8327a95b77b9e495b58b9a6e           Mrs.
6            21147  p099819  9b40170bfa65e399981717ee8731efc3           Mrs.
7            94142  p092424  5bfd3d12fae3d2fe88684bbac570c9d2            Ms.
8           112489  p045029  487448f5226005d08d36bdd75f095b31           Mrs.
9           158561  p001713  140eeac1885c820ad5592a409a3a8994            Ms.
10           43184  p040307  363788b51d40d978fe276bcb1f8a2b35           Mrs.
11          127083  p251806  4ba7c721133ef651ca54a03551746708            Ms.
12           19090  p051126  5e52c92b7e3c472aad247a239d345543           Mrs.
13           15126  p003874  178f6ae765cd4e0fb143a77c47fd65e2           Mrs.
14           62232  p233127  424819801de22a60bba7d0f4354d0258            Ms.
15           67303  p132832  bb6d6d054824fa01576ab38dfa2be160            Ms.
16          127215  p174627  4ad7e280fddff889e1355cc9f29c3b89           Mrs.
17          157771  p152491  e39abda057354c979c5b075cffbe5f88            Ms.
18          122186  p196421  fcd9b003fc1891383f340a89da02a1a6           Mrs.
19          146331  p058343  8e07a98deb1bc74c75b97521e05b1691            Ms.
20           75560  p052326  e0c1aad1f71badeff703fadc15f57680           Mrs.
21          132078  p187097  2d4a4d2d774e5c2fdd25b2ba0e7341f8           Mrs.
22           84810  p165540  30f08fbe02eba5453c4ce2e857e88eb4            Ms.
23            8636  p219330  258ef2e6ab5ce007ac6764ce15d261ba            Mr.
24           21478  p126524  74f8690562c44fc88f65f845b9fe61d0           Mrs.
25           20142  p009037  b8bf3507cee960d5fedcb27719df2d59           Mrs.
26           33903  p040091  7a0a5de5ed94e7036946b1ac3eaa99d0            Ms.
27            1156  p161033  efdc3cf14d136473c9f62becc00d4cec        Teacher
28           35430  p085706  22c8184c4660f1c589bea061d14b7f35           Mrs.
29           22088  p032018  45f16a103f1e00b7439861d4e0728a59           Mrs.
...            ...      ...                               ...            ...
109218      127181  p077978  91f5c69bf72c82edb9bc1f55596d8d95           Mrs.
109219       65838  p042022  9a6784108c76576565f46446594f99c4        Teacher
109220       21062  p064087  19c622a38a0cd76c2e9dbcc40541fabd           Mrs.
109221       81490  p117254  031e299278ac511616b2950fc1312a55        Teacher
109222       69138  p152194  6f6e951e435aa9dc966091945414bcc4            Ms.
109223        5110  p041136  6db62616b4ef6efc2310088f7ea0ae14            Ms.
109224      109630  p257774  651866d8215616f65934aafcbee21bf5            Ms.
109225      177841  p079425  c628dff071aa8028b08a5d4972bef2a1           Mrs.
109226       65359  p085810  1d286ff10ee3982b2b47813f1e415ef2            Ms.
109227       55643  p146149  e15cd063caa1ce11a45f2179535105f2           Mrs.
109228      103666  p191845  d0603199630760d8d0eb003108208998           Mrs.
109229      121219  p055363  523f95270c6aec82bee90e3931ceeeca           Mrs.
109230      117282  p235512  ee59900af64d9244487e7ed87d0bc423            Ms.
109231      170085  p248898  9d7a4dae637d1a170778e2db1515e574           Mrs.
109232       36083  p204774  c116af7435274872bea9ff123a69cf6a           Mrs.
109233      155847  p120664  b90258ab009b84e0dc11a7186d597141            Ms.
109234       52918  p057638  dd68d9fbae85933c0173c13f66291cbe            Ms.
109235       69971  p105083  9636fcacbf65eb393133a94c83c4a0d4           Mrs.
109236      120581  p254202  2950019dd34581dbcddcae683e74207a           Mrs.
109237      115336  p056813  07fd2c09f8dfcc74dbb161e1ec3df1fe           Mrs.
109238       32628  p143363  5b42211690ca8418c7c839436d0b7e49           Mrs.
109239      156548  p103958  8b9a9dc5bd4aa0301b0ff416e2ed29f6           Mrs.
109240       93971  p257729  58c112dcb2f1634a4d4236bf0dcdcb31           Mrs.
109241       36517  p180358  3e5c98480f4f39d465837b2955df6ae0           Mrs.
109242       34811  p080323  fe10e79b7aeb570dfac87eeea7e9a8f1           Mrs.
109243       38267  p048540  fadf72d6cd83ce6074f9be78a6fcd374            Mr.
109244      169142  p166281  1984d915cc8b91aa16b4d1e6e39296c6            Ms.
109245      143653  p155633  cdbfd04aa041dc6739e9e576b1fb1478           Mrs.
109246      164599  p206114  6d5675dbfafa1371f0e2f6f1b716fe2d           Mrs.
109247      128381  p191189  ca25d5573f2bd2660f7850a886395927            Ms.

       school_state project_submitted_datetime project_grade_category  \
0                IN        2016-12-05 13:43:57          Grades PreK-2
1                FL        2016-10-25 09:22:10             Grades 6-8
2                AZ        2016-08-31 12:03:56             Grades 6-8
3                KY        2016-10-06 21:16:17          Grades PreK-2
4                TX        2016-07-11 01:10:09          Grades PreK-2
5                FL        2017-04-08 22:40:43             Grades 3-5
6                CT        2017-02-17 19:58:56             Grades 6-8
7                GA        2016-09-01 00:02:15             Grades 3-5
8                SC        2016-09-25 17:00:26          Grades PreK-2
9                NC        2016-11-17 18:18:56          Grades PreK-2
10               CA        2017-01-04 16:40:30             Grades 3-5
11               CA        2016-11-14 22:57:28          Grades PreK-2
12               NY        2016-05-23 15:46:02             Grades 6-8
13               OK        2016-10-17 09:49:27          Grades PreK-2
14               MA        2017-02-14 16:29:10          Grades PreK-2
15               TX        2016-10-05 21:05:38             Grades 3-5
16               FL        2017-01-18 10:59:05          Grades PreK-2
17               NV        2016-11-23 17:14:17             Grades 3-5
```

```
18               GA        2016-08-28 15:04:42          Grades PreK-2
19               OH        2016-08-06 13:05:20             Grades 3-5
20               PA        2016-10-07 18:27:02          Grades PreK-2
21               NC        2016-05-17 19:45:13             Grades 6-8
22               CA        2016-09-01 10:09:15            Grades 9-12
23               AL        2017-01-10 11:41:06             Grades 6-8
24               FL        2017-03-31 12:34:44          Grades PreK-2
25               AL        2017-03-09 15:36:20             Grades 3-5
26               TX        2016-09-18 22:10:40          Grades PreK-2
27               LA        2016-11-06 16:02:31             Grades 3-5
28               GA        2017-01-27 12:34:59            Grades 9-12
29               VA        2016-07-15 12:58:40          Grades PreK-2
...             ...                        ...                    ...
109218           IL        2017-01-10 14:08:28          Grades PreK-2
109219           FL        2016-07-26 22:43:52          Grades PreK-2
109220           WI        2016-09-18 13:15:13             Grades 6-8
109221           NY        2016-07-03 23:09:29          Grades PreK-2
109222           NC        2016-12-01 20:29:04          Grades PreK-2
109223           GA        2017-02-15 14:07:07             Grades 6-8
109224           NY        2016-05-23 20:36:51          Grades PreK-2
109225           NC        2016-11-14 21:04:43          Grades PreK-2
109226           CA        2016-08-12 09:19:22             Grades 3-5
109227           NY        2016-10-19 10:10:04             Grades 3-5
109228           LA        2016-10-14 18:05:17          Grades PreK-2
109229           CO        2016-09-06 23:19:17          Grades PreK-2
109230           NY        2016-08-09 21:06:33            Grades 9-12
109231           AZ        2016-09-17 09:58:59            Grades 9-12
109232           MD        2017-03-14 19:59:52             Grades 3-5
109233           AZ        2016-12-21 16:36:26             Grades 6-8
109234           NY        2017-03-29 20:06:10             Grades 3-5
109235           TX        2017-01-07 14:50:08          Grades PreK-2
109236           OH        2016-08-14 08:27:24             Grades 3-5
109237           IN        2016-05-05 13:03:58             Grades 3-5
109238           WI        2016-08-01 21:17:33          Grades PreK-2
109239           MN        2016-08-15 17:01:00             Grades 6-8
109240           MD        2016-08-25 13:09:19          Grades PreK-2
109241           MD        2016-06-24 11:48:12             Grades 3-5
109242           SC        2017-03-09 20:00:33          Grades PreK-2
109243           MO        2016-06-17 12:02:31          Grades PreK-2
109244           NJ        2017-01-11 12:49:39          Grades PreK-2
109245           NJ        2016-08-25 17:11:32          Grades PreK-2
109246           NY        2016-07-29 17:53:15             Grades 3-5
109247           VA        2016-06-29 09:17:01             Grades 6-8

                                            project_title  \
0         Educational Support for English Learners at Home   
1                    Wanted: Projector for Hungry Learners   
2        Soccer Equipment for AWESOME Middle School Stu...   
3                                   Techie Kindergarteners   
4                                   Interactive Math Tools   
5        Flexible Seating for Mrs. Jarvis' Terrific Thi...   
6        Chromebooks for Special Education Reading Program   
7                                    It's the 21st Century   
8                          Targeting More Success in Class   
9          Just For the Love of Reading--\r\nPure Pleasure   
10                                   Reading Changes Lives   
11       Elevating Academics and Parent Rapports Throug...   
12                       Building Life Science Experiences   
13                          Everyone deserves to be heard!   
14                           TABLETS CAN SHOW US THE WORLD   
15                                    Making Recess Active   
16                      Making Great LEAP's With Leapfrog!   
17             Technology Teaches Tomorrow's Talents Today   
18                                               Test Time   
19                             Wiggling Our Way to Success   
20                        Magic Carpet Ride in Our Library   
21               From Sitting to Standing in the Classroom   
22                         Books for Budding Intellectuals   
23                   Instrumental Power: Conquering STEAM!   
24       S.T.E.A.M. Challenges(Science Technology Engin...   
25                                           Math Masters!   
26                                          Techy Teaching   
27                  4th Grade French Immersion Class Ipads   
28                          Hands-On Language and Literacy   
29                         Basic Classroom Supplies Needed   
...                                                    ...   
109218                 ***Multi-Sensory Classroom Wish!***   
```

```
109218                     Multi Sensory Classroom Wish!
109219                   Make Learning Fun in Grade One!
109220         Hooking Young Readers with Engaging Books
109221                               Dual language Class
109222  Replenishing Our Supplies to Extend Our Learni...
109223                       Hunger Busters for Students
109224                                STEM for 2nd Grade
109225                                 Together We Learn
109226                            Stand Up for Learning!
109227         Grab a Stool...the Fun is About to Start!
109228         Technology For Flooded Kindergarten Class
109229     Criss Cross Applesauce, we are ready to roll!
109230  Ipad Minis for Special Needs High School Students
109231            Keeping Students Informed and Inspired
109232                Everyone Needs to have an Opinion!
109233                        Engagement through Tablets
109234    Developing A Growth Mindset for School Success
109235                     Let's focus through movement!
109236                                Portable Projector
109237                 Choose Kindness Book Club: Wonder
109238  We Like to Move It, Move It! Flexible Seating ...
109239                              Integrating the Arts
109240                     Spread the Love of Literature
109241                              Read Your Heart Out!
109242                   STEM LEARNERS NEED AN IPAD MINI
109243  Privacy Shields Help Promote Independent Thinking
109244                       Technology in Our Classroom
109245            2016/2017 Beginning of the Year Basics
109246           Flexible Seating in Inclusive Classroom
109247    Classroom Tech to Develop 21st Century Leaders

                                          project_essay_1  \
0       My students are English learners that are work...
1       Our students arrive to our school eager to lea...
2       \r\n\"True champions aren't always the ones th...
3       I work at a unique school filled with both ESL...
4       Our second grade classroom next year will be m...
5       I will be moving from 2nd grade to 3rd grade a...
6       My students are a dynamic and very energetic g...
7       Not only do our students struggle with poverty...
8       My students are enthusiastic and inquisitive l...
9       Over 95% of my students are on free or reduced...
10      \"There are many little ways to enlarge your w...
11      All of our students receive free breakfast, lu...
12      My students are always working on new projects...
13      I teach in a small school district in central ...
14      My students are my babies...I want the world f...
15      Located in West Dallas, my students face sever...
16      My Preschool children, ages 3-5 years old with...
17      My students are special because they come from...
18      I teach at a Title I school in a low-income ar...
19      We are apart of an urban district and many of ...
20      The students in our school come from diverse b...
21      My students walk into school every day full of...
22      Every day in my English classroom, we work to ...
23      100% of our musical students eat free breakfas...
24      This year, I am teaching in an EFL (Extended F...
25      My students are highly motivated to succeed. U...
26      I teach 22 bright 5 and 6 year olds. My studen...
27      My students spend most of their day learning f...
28      My students all have a primary diagnosis of au...
29      I have an awesome group of 24 students any tea...
...                                                   ...
109218  My students are an amazing group of kind heart...
109219  Creating an Interactive Learning Environment! ...
109220  Do you remember middle school?  I am sure tons...
109221  Most of the students are ENL students.  This i...
109222  For most of my students, this is the first tim...
109223  The students that I serve are in a low-income ...
109224  My students are an amazing group of kids. Thes...
109225  My students come in ready to rock and roll eve...
109226  My students are inquisitive, engaging 4th grad...
109227  My students attend a Title 1 school in Upstate...
109228  Every day, my kindergarten class comes in exci...
109229  Each morning 21 loving and smiling faces walk ...
109230  can only be described as a diverse group of ch...
109231  There is nothing better than seeing a student ...
109232  Our class is home to a diverse class of studen...
```

```
109232  Our class is home to a diverse class of studen...
109233  My students are hard workers who strive to be ...
109234  are kind, respectful, and eager learners. They...
109235  As a Kindergarten teacher in a low-income/high...
109236  Many students have a hard time making a connec...
109237  \"You can find magic wherever you look. Sit ba...
109238  We are the Bucket Fillers! This means we fill ...
109239  My students are amazing and motivated. We are ...
109240  Leaving your family to come to school for the ...
109241  I had a wonderful group of inquisitive and ent...
109242  My students come from a rural community in Sou...
109243  Welcome to Mr. Ramos's 2nd grade classroom! We...
109244  Every morning, we start our day with our core ...
109245  This is a great group of sharing and caring st...
109246  Our students live in a small rural community. ...
109247  When was the last time that you used math? Pro...

                                             project_essay_2  \
0       \"The limits of your language are the limits o...
1       The projector we need for our school is very c...
2       The students on the campus come to school know...
3       My students live in high poverty conditions wi...
4       For many students, math is a subject that does...
5       These flexible seating options will allow my s...
6       My students are an engaging and active group o...
7       My students need 4 iPads, the latest technolog...
8       My second graders need extra activity time dur...
9       Reading is Fundamental! My students will read ...
10      I've had 8 sets of students enjoy the books in...
11      With three chromebooks, I can teach the Common...
12      My Spanish Dual Language students are always r...
13      My students are smart, creative, and also have...
14      Having this computer in the classroom would pr...
15      Due to the size of our school, and the tiny na...
16      Having a set of Leapfrog iPads and educational...
17      Classroom ChromebookCar\r\n\r\nMy name is Shan...
18      My 2nd grade students will benefit from having...
19      Many of my students struggle to sit still for ...
20      Each week our students love visiting the schoo...
21      I want to purchase desks in my classroom that ...
22      My students need books that interest them so t...
23      We need classroom instruments for our band pro...
24      I will use these items to create S.T.E.A.M. bi...
25      These math games will help reinforce the skill...
26      The iPads will be effectively used to improve ...
27      The iPads will also be used to enhance the stu...
28      Children with autism struggle in core deficit ...
29      My students need basic school supplies such as...
...                                                  ...
109218  Our Kindergarten classroom currently lacks any...
109219  \r\n\r\nThese materials will provide flexible ...
109220  These books will go into the hands of young re...
109221  These school supplies will help my students to...
109222  For most of my students, this is their first y...
109223  Since the students in the schools I serve are ...
109224  STEM is the use of science, technology, engine...
109225  Having two classes, that are bilingual, is ama...
109226  My students love to learn and move. Sitting in...
109227  My students are very active.  Many students at...
109228  Before the flooding, my classroom had 4 deskto...
109229  My classroom is in need of a large learning ru...
109230  All of my students demonstrate a desire to lea...
109231  My PE classes are very fitness based and my st...
109232  Our students struggle when it comes to writing...
109233  Having a tablet in the classroom will help my ...
109234  These materials will make a difference in my c...
109235  After teaching kindergarten for 9 years, I hav...
109236  I would love to have a projector to help stude...
109237  We are the most diverse school district in Ind...
109238  We like to move it, move it!  My students do N...
109239  Our school is a creative arts integrated schoo...
109240  There is no tool as powerful as a good book. I...
109241  Children need to be exposed to all different t...
109242  Students will use the iPad mini for hands on S...
109243  I would like to start preparing my second grad...
109244  In this technological age, we need to give our...
109245  My students learn about special events, holida...
109246  Flexible classroom seating has been researched...
```

```
109246  Flexible classroom seating has been researched...
109247  According to Forbes Magazine (2014), companies...

                                          project_essay_3  \
0                                                     NaN   
1                                                     NaN   
2                                                     NaN   
3                                                     NaN   
4                                                     NaN   
5                                                     NaN   
6                                                     NaN   
7                                                     NaN   
8                                                     NaN   
9                                                     NaN   
10                                                    NaN   
11                                                    NaN   
12                                                    NaN   
13                                                    NaN   
14                                                    NaN   
15                                                    NaN   
16                                                    NaN   
17                                                    NaN   
18                                                    NaN   
19                                                    NaN   
20                                                    NaN   
21                                                    NaN   
22                                                    NaN   
23                                                    NaN   
24                                                    NaN   
25                                                    NaN   
26                                                    NaN   
27                                                    NaN   
28                                                    NaN   
29                                                    NaN   
...                                                   ...   
109218                                                NaN   
109219                                                NaN   
109220                                                NaN   
109221                                                NaN   
109222                                                NaN   
109223                                                NaN   
109224                                                NaN   
109225                                                NaN   
109226                                                NaN   
109227                                                NaN   
109228                                                NaN   
109229                                                NaN   
109230                                                NaN   
109231                                                NaN   
109232                                                NaN   
109233                                                NaN   
109234                                                NaN   
109235                                                NaN   
109236                                                NaN   
109237  \"Some things you just can't explain. You don'...   
109238                                                NaN   
109239                                                NaN   
109240                                                NaN   
109241                                                NaN   
109242                                                NaN   
109243                                                NaN   
109244                                                NaN   
109245                                                NaN   
109246                                                NaN   
109247                                                NaN   

                                          project_essay_4  \
0                                                     NaN   
1                                                     NaN   
2                                                     NaN   
3                                                     NaN   
4                                                     NaN   
5                                                     NaN   
6                                                     NaN   
7                                                     NaN   
8                                                     NaN   
9                                                     NaN   
10                                                    NaN   
```

```
10                                                      NaN
11                                                      NaN
12                                                      NaN
13                                                      NaN
14                                                      NaN
15                                                      NaN
16                                                      NaN
17                                                      NaN
18                                                      NaN
19                                                      NaN
20                                                      NaN
21                                                      NaN
22                                                      NaN
23                                                      NaN
24                                                      NaN
25                                                      NaN
26                                                      NaN
27                                                      NaN
28                                                      NaN
29                                                      NaN
...                                                     ...
109218                                                  NaN
109219                                                  NaN
109220                                                  NaN
109221                                                  NaN
109222                                                  NaN
109223                                                  NaN
109224                                                  NaN
109225                                                  NaN
109226                                                  NaN
109227                                                  NaN
109228                                                  NaN
109229                                                  NaN
109230                                                  NaN
109231                                                  NaN
109232                                                  NaN
109233                                                  NaN
109234                                                  NaN
109235                                                  NaN
109236                                                  NaN
109237  I am asking for a class set of books. This wil...
109238                                                  NaN
109239                                                  NaN
109240                                                  NaN
109241                                                  NaN
109242                                                  NaN
109243                                                  NaN
109244                                                  NaN
109245                                                  NaN
109246                                                  NaN
109247                                                  NaN

                                  project_resource_summary  \
0       My students need opportunities to practice beg...
1       My students need a projector to help with view...
2       My students need shine guards, athletic socks,...
3       My students need to engage in Reading and Math...
4       My students need hands on practice in mathemat...
5       My students need movement to be successful. Be...
6       My students need some dependable laptops for d...
7       My students need ipads to help them access a w...
8       My students need three devices and three manag...
9       My students need great books to use during Ind...
10      My students need books by their favorite autho...
11      My students need paper, three chromebooks, and...
12      My students need 3D and 4D life science activi...
13      My students need access to technology that wil...
14      My students need 5 tablets for our classroom t...
15      My students need activities to play during rec...
16      My students need 2 LeapPad that will engage th...
17      My students need Chromebooks to publish writte...
18      My students need privacy partitions to use whi...
19      My students need 7 Hokki stools to encourage a...
20      My students need carpet in our library to brig...
21      My students need desks to stand at and be able...
22      My students need books so that they can become...
23      My students need these instruments to give the...
```

```
24      My students need building materials, such as g...
25      My students need the learning centers and mult...
26      My students need 2 ipad minis to enhance learn...
27      My students need Ipads to work in smaller grou...
28      My students need to increase language and lite...
29      My students need basic school supplies such as...
...                                                   ...
109218  My students need to have a multi sensory learn...
109219  My students need access to a fun, learning and...
109220  My students need engaging books with topics th...
109221  My students need seat sacks to have their book...
109222  My students need classroom supplies to repleni...
109223  My students need snacks for during the day whe...
109224  My students need STEM activities to make their...
109225  My students need a whole group environment to ...
109226  My students need standing desks and wobble cha...
109227  My students need ergonomic seats that are made...
109228  My students need chromebooks to replace comput...
109229  My students need a rug for whole group learnin...
109230  My students need two Ipad minis with cases to ...
109231  My students need access to printed materials f...
109232  My students need a set of 5 opinion picture bo...
109233  My students need a tablet to increase motivati...
109234  My students need story books, team building ac...
109235  My students need Bouncy Bands to help students...
109236  My students need a portable projector. This wi...
109237  My students need a class set of Wonder books, ...
109238  My students need flexible seating options like...
109239  My students need a green screen kit and iPad s...
109240  My students need books to supplement the thema...
109241  My students need a nonfiction leveled library ...
109242  My students need an iPad mini anda Life Proof ...
109243  My students need these privacy partitions to h...
109244  My students need two iPad's and protective cas...
109245  My students need giant comfy pillows in order ...
109246  My students need flexible seating options: bea...
109247  My students need opportunities to work with te...

        teacher_number_of_previously_posted_projects  project_is_approved  \
0                                                  0                    0
1                                                  7                    1
2                                                  1                    0
3                                                  4                    1
4                                                  1                    1
5                                                  1                    1
6                                                  1                    1
7                                                  7                    1
8                                                 28                    1
9                                                 36                    1
10                                                37                    1
11                                                32                    1
12                                                 5                    0
13                                                30                    1
14                                                15                    0
15                                                 3                    1
16                                                 1                    1
17                                                 0                    1
18                                                 0                    1
19                                                 9                    1
20                                                23                    1
21                                                 0                    1
22                                                 0                    0
23                                                 2                    1
24                                                 0                    1
25                                                11                    0
26                                                 2                    1
27                                                 2                    1
28                                                 5                    0
29                                                 0                    1
...                                              ...                  ...
109218                                             4                    0
109219                                             0                    0
109220                                             3                    1
109221                                             1                    1
109222                                            34                    1
109223                                            12                    1
109224                                             7                    1
```

```
109225                                          1                    0
109226                                         47                    1
109227                                          0                    1
109228                                          8                    1
109229                                          0                    1
109230                                          0                    1
109231                                          7                    1
109232                                          0                    1
109233                                          1                    1
109234                                          9                    1
109235                                          1                    1
109236                                          6                    1
109237                                          4                    1
109238                                         41                    1
109239                                          6                    1
109240                                         10                    1
109241                                          0                    1
109242                                         26                    1
109243                                          0                    1
109244                                          0                    1
109245                                          3                    1
109246                                          0                    1
109247                                          0                    1

                         clean_categories  \
0                       Literacy_Language   
1             History_Civics Health_Sports   
2                           Health_Sports   
3           Literacy_Language Math_Science   
4                            Math_Science   
5           Literacy_Language SpecialNeeds   
6           Literacy_Language SpecialNeeds   
7                            Math_Science   
8                           Health_Sports   
9                       Literacy_Language   
10                      Literacy_Language   
11       Literacy_Language AppliedLearning   
12                           Math_Science   
13                           SpecialNeeds   
14                      Literacy_Language   
15                          Health_Sports   
16          Literacy_Language SpecialNeeds   
17          Math_Science Literacy_Language   
18                        AppliedLearning   
19                          Health_Sports   
20                      Literacy_Language   
21               Math_Science SpecialNeeds   
22                      Literacy_Language   
23                             Music_Arts   
24                           Math_Science   
25                           Math_Science   
26          Literacy_Language Math_Science   
27          Literacy_Language Math_Science   
28          Literacy_Language SpecialNeeds   
29       Literacy_Language AppliedLearning   
...                                   ...   
109218                  Literacy_Language   
109219   Literacy_Language History_Civics   
109220                  Literacy_Language   
109221                  Literacy_Language   
109222      Literacy_Language Math_Science   
109223                 Warmth Care_Hunger   
109224      Math_Science Literacy_Language   
109225   AppliedLearning Literacy_Language   
109226                      Health_Sports   
109227                  Literacy_Language   
109228      Literacy_Language Math_Science   
109229      Literacy_Language Math_Science   
109230                       SpecialNeeds   
109231                      Health_Sports   
109232                  Literacy_Language   
109233         Math_Science History_Civics   
109234                    AppliedLearning   
109235   AppliedLearning Literacy_Language   
109236                       SpecialNeeds   
109237                  Literacy_Language   
109238                      Health_Sports   
```

```
109239    Literacy_Language Math_Science
109240                 Literacy_Language
109241  AppliedLearning Literacy_Language
109242    Math_Science Literacy_Language
109243    Literacy_Language Math_Science
109244    Literacy_Language Math_Science
109245    Literacy_Language Math_Science
109246        Health_Sports SpecialNeeds
109247      AppliedLearning Math_Science

                         clean_subcategories  \
0                                ESL Literacy
1                 Civics_Government TeamSports
2                   Health_Wellness TeamSports
3                         Literacy Mathematics
4                                  Mathematics
5              Literature_Writing SpecialNeeds
6                        Literacy SpecialNeeds
7                                  Mathematics
8                              Health_Wellness
9                  Literacy Literature_Writing
10                                    Literacy
11                   Literacy ParentInvolvement
12      EnvironmentalScience Health_LifeScience
13                                SpecialNeeds
14                                    Literacy
15                             Health_Wellness
16                       Literacy SpecialNeeds
17          AppliedSciences Literature_Writing
18                            EarlyDevelopment
19                             Health_Wellness
20                                    Literacy
21             Health_LifeScience SpecialNeeds
22                                    Literacy
23                                       Music
24                 AppliedSciences Mathematics
25                                 Mathematics
26                        Literacy Mathematics
27                ForeignLanguages Mathematics
28                       Literacy SpecialNeeds
29                              Literacy Other
...                                        ...
109218                      Literature_Writing
109219       Literature_Writing SocialSciences
109220                                Literacy
109221                        ForeignLanguages
109222                    Literacy Mathematics
109223                      Warmth Care_Hunger
109224                AppliedSciences Literacy
109225   CharacterEducation Literature_Writing
109226             Gym_Fitness Health_Wellness
109227                                Literacy
109228                    Literacy Mathematics
109229          Literature_Writing Mathematics
109230                            SpecialNeeds
109231                             Gym_Fitness
109232             Literacy Literature_Writing
109233       AppliedSciences FinancialLiteracy
109234                      CharacterEducation
109235     EarlyDevelopment Literature_Writing
109236                            SpecialNeeds
109237               ForeignLanguages Literacy
109238                         Health_Wellness
109239                    Literacy Mathematics
109240                                Literacy
109241               EarlyDevelopment Literacy
109242                AppliedSciences Literacy
109243          Literature_Writing Mathematics
109244                    Literacy Mathematics
109245                    Literacy Mathematics
109246             Health_Wellness SpecialNeeds
109247          College_CareerPrep Mathematics

                                                    essay   price  quantity
0       My students are English learners that are work...  154.60        23
1       Our students arrive to our school eager to lea...  299.00         1
2       \r\n\"True champions aren't always the ones th...  516.85        22
```

```
3       I work at a unique school filled with both ESL...  232.90         4
4       Our second grade classroom next year will be m...   67.98         4
5       I will be moving from 2nd grade to 3rd grade a...  113.22        11
6       My students are a dynamic and very energetic g...  159.99         3
7       Not only do our students struggle with poverty...  229.00         4
8       My students are enthusiastic and inquisitive l...  241.98         6
9       Over 95% of my students are on free or reduced...  125.36        14
10      \"There are many little ways to enlarge your w...  100.21        10
11      All of our students receive free breakfast, lu...  431.77         8
12      My students are always working on new projects...  219.46        22
13      I teach in a small school district in central ...  399.99         1
14      My students are my babies...I want the world f...   91.94        10
15      Located in West Dallas, my students face sever...  435.84        24
16      My Preschool children, ages 3-5 years old with...  298.43         7
17      My students are special because they come from...  158.63        12
18      I teach at a Title I school in a low-income ar...   59.98         4
19      We are apart of an urban district and many of ...  749.42         7
20      The students in our school come from diverse b...  213.85         1
21      My students walk into school every day full of...  250.91         4
22      Every day in my English classroom, we work to ...  278.09        21
23      100% of our musical students eat free breakfas...  299.98         2
24      This year, I am teaching in an EFL (Extended F...  250.00         6
25      My students are highly motivated to succeed. U...  268.99         2
26      I teach 22 bright 5 and 6 year olds. My studen...  280.83         4
27      My students spend most of their day learning f...  660.84         7
28      My students all have a primary diagnosis of au...  129.98         3
29      I have an awesome group of 24 students any tea...   86.74        53
...                                                   ...     ...       ...
109218  My students are an amazing group of kind heart...  747.00         3
109219  Creating an Interactive Learning Environment! ...  300.18        14
109220  Do you remember middle school?  I am sure tons...  121.59        14
109221  Most of the students are ENL students.  This i...  289.52        32
109222  For most of my students, this is the first tim...  241.08        40
109223  The students that I serve are in a low-income ...  692.17        46
109224  My students are an amazing group of kids. Thes...  915.27         7
109225  My students come in ready to rock and roll eve...  737.95         2
109226  My students are inquisitive, engaging 4th grad...  379.96         9
109227  My students attend a Title 1 school in Upstate...  428.24         5
109228  Every day, my kindergarten class comes in exci...  159.43         4
109229  Each morning 21 loving and smiling faces walk ...  688.00         2
109230  can only be described as a diverse group of ch...  309.60         4
109231  There is nothing better than seeing a student ...  155.70         5
109232  Our class is home to a diverse class of studen...   43.20        20
109233  My students are hard workers who strive to be ...  490.05         1
109234  are kind, respectful, and eager learners. They...  273.72         6
109235  As a Kindergarten teacher in a low-income/high...   11.86        24
109236  Many students have a hard time making a connec...  269.00         1
109237  \"You can find magic wherever you look. Sit ba...   30.76        30
109238  We are the Bucket Fillers! This means we fill ...  267.50        12
109239  My students are amazing and motivated. We are ...  178.98         2
109240  Leaving your family to come to school for the ...  225.10        30
109241  I had a wonderful group of inquisitive and ent...  659.00         1
109242  My students come from a rural community in Sou...  592.16         2
109243  Welcome to Mr. Ramos's 2nd grade classroom! We...   59.98         8
109244  Every morning, we start our day with our core ...  846.32         4
109245  This is a great group of sharing and caring st...  239.96         4
109246  Our students live in a small rural community. ...   73.05        16
109247  When was the last time that you used math? Pro...  109.90         5

[109248 rows x 20 columns]
```

In [13]:

```
approved_price = project_data[project_data['project_is_approved']==1]['price'].values
print
rejected_price = project_data[project_data['project_is_approved']==0]['price'].values
```

1.3 Text preprocessing

1.3.1 Essay Text

In [14]:

```
project_data.head(2)
```

Out[14]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_grade_cate
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	Grades P
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Grade

In [15]:

```
# printing some random essays.
print(project_data['essay'].values[0])
print("="*50)
print(project_data['essay'].values[150])
print("="*50)
```

```
My students are English learners that are working on English as their second or third languages. W
e are a melting pot of refugees, immigrants, and native-born Americans bringing the gift of langua
ge to our school. \r\n\r\n We have over 24 languages represented in our English Learner program wi
th students at every level of mastery.  We also have over 40 countries represented with the
families within our school.  Each student brings a wealth of knowledge and experiences to us that
open our eyes to new cultures, beliefs, and respect.\"The limits of your language are the limits o
f your world.\"-Ludwig Wittgenstein  Our English learner's have a strong support system at home th
at begs for more resources.  Many times our parents are learning to read and speak English along s
ide of their children.  Sometimes this creates barriers for parents to be able to help their child
learn phonetics, letter recognition, and other reading skills.\r\n\r\nBy providing these dvd's and
players, students are able to continue their mastery of the English language even if no one at hom
e is able to assist.  All families with students within the Level 1 proficiency status, will be a
offered to be a part of this program.  These educational videos will be specially chosen by the En
glish Learner Teacher and will be sent home regularly to watch.  The videos are to help the child
develop early reading skills.\r\n\r\nParents that do not have access to a dvd player will have the
opportunity to check out a dvd player to use for the year.  The plan is to use these videos and ed
ucational dvd's for the years to come for other EL students.\r\nnannan
==================================================
The 51 fifth grade students that will cycle through my classroom this year all love learning, at l
east most of the time. At our school, 97.3% of the students receive free or reduced price lunch. O
f the 560 students, 97.3% are minority students. \r\nThe school has a vibrant community that loves
to get together and celebrate. Around Halloween there is a whole school parade to show off the bea
utiful costumes that students wear. On Cinco de Mayo we put on a big festival with crafts made by
the students, dances, and games. At the end of the year the school hosts a carnival to celebrate t
he hard work put in during the school year, with a dunk tank being the most popular activity.My st
udents will use these five brightly colored Hokki stools in place of regular, stationary, 4-legged
chairs. As I will only have a total of ten in the classroom and not enough for each student to hav
e an individual one, they will be used in a variety of ways. During independent reading time they
will be used as special chairs students will each use on occasion. I will utilize them in place of
chairs at my small group tables during math and reading times. The rest of the day they will be us
ed by the students who need the highest amount of movement in their life in order to stay focused
on school.\r\n\r\nWhenever asked what the classroom is missing, my students always say more Hokki
Stools. They can't get their fill of the 5 stools we already have. When the students are sitting i
n group with me on the Hokki Stools, they are always moving, but at the same time doing their
work. Anytime the students get to pick where they can sit, the Hokki Stools are the first to be ta
ken. There are always students who head over to the kidney table to get one of the stools who are
disappointed as there are not enough of them. \r\n\r\nWe ask a lot of students to sit for 7 hours
a day. The Hokki stools will be a compromise that allow my students to do desk work and move at th
e same time. These stools will help students to meet their 60 minutes a day of movement by
allowing them to activate their core muscles for balance while they sit. For many of my students,
these chairs will take away the barrier that exists in schools for a child who can't sit
still.nannan
==================================================
```

In [16]:

```
# https://stackoverflow.com/a/47091490/4084039
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)

    # general
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase
```

In [17]:

```
sent = decontracted(project_data['essay'].values[4000])
print(sent)
print("="*50)
```

```
I teach language arts and social studies to about 50 students each day.  I teach two groups of ama
zing kids each day!\r\n\r\nThe students in my classroom range from advanced or gifted learners to
students with various learning disabilities. My school is located in an urban environment in
Maryland. The school is a Title I (low-income) school, and 99% of the students in the school recei
ve free and reduced price lunch. All students at my school receive free breakfast which is the mos
t important meal of the day!High interest reading supports comprehension and learning. I want to e
ncourage a love of reading by choosing books that interest my third grade students. Many of my
students are classified as \"struggling readers\". There is extensive research to support the
premise that the best way to become a better reader is to read more. In order for my students to b
ecome better or more fluent readers I need to increase both the quantity and quality of their read
ing.  They need reading materials that they can read and will want to read. \r\n\r\nI want to send
my students into summer vacation with a high interest book. If they find success and interest with
one book, research shows that learning will generate more learning! The book I have chosen is read
able, has a convincing plot, and has realistic characters.nannan
==================================================
```

In [18]:

```
# \r \n \t remove from string python: http://texthandler.com/info/remove-line-breaks-python/
sent = sent.replace('\\r', ' ')
sent = sent.replace('\\"', ' ')
sent = sent.replace('\\n', ' ')
print(sent)
```

```
I teach language arts and social studies to about 50 students each day.  I teach two groups of ama
zing kids each day!    The students in my classroom range from advanced or gifted learners to stud
ents with various learning disabilities. My school is located in an urban environment in Maryland.
The school is a Title I (low-income) school, and 99% of the students in the school receive free an
d reduced price lunch. All students at my school receive free breakfast which is the most
important meal of the day!High interest reading supports comprehension and learning. I want to enc
ourage a love of reading by choosing books that interest my third grade students. Many of my
students are classified as  struggling readers . There is extensive research to support the
premise that the best way to become a better reader is to read more. In order for my students to b
ecome better or more fluent readers I need to increase both the quantity and quality of their read
ing.  They need reading materials that they can read and will want to read.     I want to send my
students into summer vacation with a high interest book. If they find success and interest with on
e book, research shows that learning will generate more learning! The book I have chosen is readab
le, has a convincing plot, and has realistic characters.nannan
```

In [19]:

```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
print(sent)
```

```
I teach language arts and social studies to about 50 students each day I teach two groups of amazi
ng kids each day The students in my classroom range from advanced or gifted learners to students w
ith various learning disabilities My school is located in an urban environment in Maryland The sch
ool is a Title I low income school and 99 of the students in the school receive free and reduced p
rice lunch All students at my school receive free breakfast which is the most important meal of th
e day High interest reading supports comprehension and learning I want to encourage a love of read
ing by choosing books that interest my third grade students Many of my students are classified as
struggling readers There is extensive research to support the premise that the best way to become
a better reader is to read more In order for my students to become better or more fluent readers I
need to increase both the quantity and quality of their reading They need reading materials that t
hey can read and will want to read I want to send my students into summer vacation with a high int
erest book If they find success and interest with one book research shows that learning will gener
ate more learning The book I have chosen is readable has a convincing plot and has realistic chara
cters nannan
```

In [20]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've",
\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his',
'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them',
'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll",
'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having',
'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', '
while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',
'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under'
, 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'e
ach', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll'
, 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "do
esn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn',
"mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn',
"wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [21]:

```
# Combining all the above statemennts
from tqdm import tqdm
preprocessed_essays = []
# tqdm is for printing the status bar
for sentance in tqdm(project_data['essay'].values):
    sent = decontracted(sentance)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    # https://gist.github.com/sebleier/554280
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    preprocessed_essays.append(sent.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████████████| 109248/109248
[02:12<00:00, 823.27it/s]
```

In [22]:

```
# after preprocesing
preprocessed_essays[2000]

project_data['essay']=pd.DataFrame(preprocessed_essays)
```

1.3.2 Project title Text

In [23]:

```
# similarly you can preprocess the titles also

# Combining all the above statemennts
from tqdm import tqdm
preprocessed_titles = []
# tqdm is for printing the status bar
for sentance in tqdm(project_data['project_title'].values):
    sent = decontracted(sentance)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    # https://gist.github.com/sebleier/554280
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    preprocessed_titles.append(sent.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████████████| 109248/109248
[00:06<00:00, 16172.93it/s]
```

In [24]:

```
preprocessed_titles[2000]

project_data['project_title']=pd.DataFrame(preprocessed_titles)
```

1. 4 Preparing data for models

In [25]:

```
project_data.columns
```

Out[25]:

```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', 'project_grade_category', 'project_title',
       'project_essay_1', 'project_essay_2', 'project_essay_3',
       'project_essay_4', 'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay', 'price',
       'quantity'],
      dtype='object')
```

we are going to consider

```
- school_state : categorical data
- clean_categories : categorical data
- clean_subcategories : categorical data
- project_grade_category : categorical data
- teacher_prefix : categorical data

- project_title : text data
- text : text data
- project_resource_summary: text data

- quantity : numerical
- teacher_number_of_previously_posted_projects : numerical
- price : numerical
```

In [26]:

```
grades = list(project_data['project_grade_category'].values)
```

```
# remove special characters from list of strings python:
https://stackoverflow.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-python
grades_list = []
for i in grades:
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Science", "Warmth", "Care & H
unger"]
        j = j.replace(' ','_') # we are placeing all the ' '(space) with ''(empty) ex:"Math & Scien
ce"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spaces
        temp = temp.replace('-','_') # we are replacing the & value into
        temp = temp.replace('Grades','grades') # we are replacing the & value into
        temp = temp.replace('PreK','prek') # we are replacing the & value into
    grades_list.append(temp.strip())

project_data['project_grade_category'] = grades_list
```

In [27]:

```
y = project_data['project_is_approved'].values
project_data.drop(['project_is_approved'], axis=1, inplace=True)
project_data.head(1)
```

Out[27]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_grade_categ
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	grades_pre

Assignment 4: Naive Bayes

1. **Apply Multinomial NaiveBayes on these feature sets**
    - Set 1: categorical, numerical features + project_title(BOW) + preprocessed_eassay (BOW)
    - Set 2: categorical, numerical features + project_title(TFIDF)+ preprocessed_eassay (TFIDF)
2. **The hyper paramter tuning(find best Alpha)**
    - Find the best hyper parameter which will give the maximum AUC value
    - Consider a wide range of alpha values for hyperparameter tuning, start as low as 0.00001
    - Find the best hyper paramter using k-fold cross validation or simple cross validation data
    - Use gridsearch cv or randomsearch cv or you can also write your own for loops to do this task of hyperparameter tuning
3. **Feature importance**
    - Find the top 10 features of positive class and top 10 features of negative class for both feature sets Set 1 and Set 2 using values of `feature_log_prob_` parameter of MultinomialNB and print their corresponding feature names
4. **Representation of results**
    - You need to plot the performance of model both on train data and cross validation data for each hyper parameter, like shown in the figure. Here on X-axis you will have alpha values, since they have a wide range, just to represent those alpha values on the graph, apply log function on those alpha values.
    - Once after you found the best hyper parameter, you need to train your model with it, and find the AUC on test data and plot the ROC curve on both train and test.
    - Along with plotting ROC curve, you need to print the confusion matrix with predicted and original labels of test data points. Please visualize your confusion matrices using seaborn heatmaps.

5. Conclusion

- You need to summarize the results at the end of the notebook, summarize it in the table format. To print out a table please refer to this prettytable library link

2. Naive Bayes

2.1 Splitting data into Train and cross validation(or test): Stratified Sampling

In [28]:

```
X=project_data
```

In [29]:

```
#train test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, stratify=y)
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.33, stratify=y_train)
```

In [30]:

```
print(X_train.shape, y_train.shape)
print(X_cv.shape, y_cv.shape)
print(X_test.shape, y_test.shape)
print("="*100)
```

```
(49041, 19) (49041,)
(24155, 19) (24155,)
(36052, 19) (36052,)
====================================================================================================
```

2.2 Make Data Model Ready: encoding numerical, categorical features

Normalizing the numerical features: Price

In [31]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X_train['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96 96.01 ... 368.98 80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1) if it contains a single sample.
normalizer.fit(X_train['price'].values.reshape(-1,1))
X_train_price_norm = normalizer.transform(X_train['price'].values.reshape(-1,1))
X_cv_price_norm = normalizer.transform(X_cv['price'].values.reshape(-1,1))
X_test_price_norm = normalizer.transform(X_test['price'].values.reshape(-1,1))
print("After vectorizations")
print(X_train_price_norm.shape, y_train.shape)
print(X_cv_price_norm.shape, y_cv.shape)
print(X_test_price_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(49041, 1) (49041,)
(24155, 1) (24155,)
(36052, 1) (36052,)
====================================================================================================
```

Normalizing the numerical features: Previously posted projects

In [32]:

```python
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X_train['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96 96.01 ... 368.98 80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1) if it contains a single sample.
normalizer.fit(X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))
X_train_ppp_norm = normalizer.transform(X_train['teacher_number_of_previously_posted_projects'].va
lues.reshape(-1,1))
X_cv_ppp_norm = normalizer.transform(X_cv['teacher_number_of_previously_posted_projects'].values.r
eshape(-1,1))
X_test_ppp_norm =
normalizer.transform(X_test['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))
print("After vectorizations")
print(X_train_ppp_norm.shape, y_train.shape)
print(X_cv_ppp_norm.shape, y_cv.shape)
print(X_test_ppp_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(49041, 1) (49041,)
(24155, 1) (24155,)
(36052, 1) (36052,)
====================================================================================================
```

Normalizing the numerical features : Quantity

In [33]:

```python
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X_train['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96 96.01 ... 368.98 80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1) if it contains a single sample.
normalizer.fit(X_train['quantity'].values.reshape(-1,1))
X_train_qty_norm = normalizer.transform(X_train['quantity'].values.reshape(-1,1))
X_cv_qty_norm = normalizer.transform(X_cv['quantity'].values.reshape(-1,1))
X_test_qty_norm = normalizer.transform(X_test['quantity'].values.reshape(-1,1))
print("After vectorizations")

print(X_train_qty_norm.shape, y_train.shape)
print(X_cv_qty_norm.shape, y_cv.shape)
print(X_test_qty_norm.shape, y_test.shape)

print("="*100)
```

```
After vectorizations
(49041, 1) (49041,)
(24155, 1) (24155,)
(36052, 1) (36052,)
====================================================================================================
```

One hot encoding the catogorical features: State

In [34]:

```python
vectorizer = CountVectorizer()
```

```
vectorizer.fit(X_train['school_state'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_state_ohe = vectorizer.transform(X_train['school_state'].values)
X_cv_state_ohe = vectorizer.transform(X_cv['school_state'].values)
X_test_state_ohe = vectorizer.transform(X_test['school_state'].values)
print("After vectorizations")
print(X_train_state_ohe.shape, y_train.shape)
print(X_cv_state_ohe.shape, y_cv.shape)
print(X_test_state_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)

stateVec=vectorizer.get_feature_names()
type(stateVec)
```

```
After vectorizations
(49041, 51) (49041,)
(24155, 51) (24155,)
(36052, 51) (36052,)
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga', 'hi', 'ia', 'id', 'il', 'in', 'k
s', 'ky', 'la', 'ma', 'md', 'me', 'mi', 'mn', 'mo', 'ms', 'mt', 'nc', 'nd', 'ne', 'nh', 'nj', 'nm',
'nv', 'ny', 'oh', 'ok', 'or', 'pa', 'ri', 'sc', 'sd', 'tn', 'tx', 'ut', 'va', 'vt', 'wa', 'wi', 'wv
', 'wy']
====================================================================================================
```

Out[34]:

```
list
```

One hot encoding the catogorical features: Project Grade

In [35]:

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['project_grade_category'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_grade_ohe = vectorizer.transform(X_train['project_grade_category'].values)
X_cv_grade_ohe = vectorizer.transform(X_cv['project_grade_category'].values)
X_test_grade_ohe = vectorizer.transform(X_test['project_grade_category'].values)
print("After vectorizations")
print(X_train_grade_ohe.shape, y_train.shape)
print(X_cv_grade_ohe.shape, y_cv.shape)
print(X_test_grade_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)

projGradeVec=vectorizer.get_feature_names()
```

```
After vectorizations
(49041, 4) (49041,)
(24155, 4) (24155,)
(36052, 4) (36052,)
['grades_3_5', 'grades_6_8', 'grades_9_12', 'grades_prek_2']
====================================================================================================
```

One hot encoding the catogorical features: Teacher Prefix

In [36]:

```
#replacing nan with empty string
X_train.teacher_prefix=X_train.teacher_prefix.fillna('')
X_cv.teacher_prefix=X_cv.teacher_prefix.fillna('')
X_test.teacher_prefix=X_test.teacher_prefix.fillna('')
uniqueData=X_train['teacher_prefix'].unique()
print(uniqueData)

vectorizer = CountVectorizer(lowercase=False, binary=True)
vectorizer.fit(X_train['teacher_prefix'].values) # fit has to happen only on train data
```

```
# we use the fitted CountVectorizer to convert the text to vector
X_train_teacher_ohe = vectorizer.transform(X_train['teacher_prefix'].values)
X_cv_teacher_ohe = vectorizer.transform(X_cv['teacher_prefix'].values)
X_test_teacher_ohe = vectorizer.transform(X_test['teacher_prefix'].values)
print("After vectorizations")
print(X_train_teacher_ohe.shape, y_train.shape)
print(X_cv_teacher_ohe.shape, y_cv.shape)
print(X_test_teacher_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)

prefixteacherVec=vectorizer.get_feature_names()
```

```
['Ms.' 'Mrs.' 'Mr.' 'Teacher' 'Dr.' '']
After vectorizations
(49041, 5) (49041,)
(24155, 5) (24155,)
(36052, 5) (36052,)
['Dr', 'Mr', 'Mrs', 'Ms', 'Teacher']
====================================================================================================
```

One hot encoding the catogorical features: Clean categories

In [37]:

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_categories'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_ccat_ohe = vectorizer.transform(X_train['clean_categories'].values)
X_cv_ccat_ohe = vectorizer.transform(X_cv['clean_categories'].values)
X_test_ccat_ohe = vectorizer.transform(X_test['clean_categories'].values)
print("After vectorizations")
print(X_train_ccat_ohe.shape, y_train.shape)
print(X_cv_ccat_ohe.shape, y_cv.shape)
print(X_test_ccat_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)

cleanCatVec=vectorizer.get_feature_names()
```

```
After vectorizations
(49041, 9) (49041,)
(24155, 9) (24155,)
(36052, 9) (36052,)
['appliedlearning', 'care_hunger', 'health_sports', 'history_civics', 'literacy_language',
'math_science', 'music_arts', 'specialneeds', 'warmth']
====================================================================================================
```

One hot encoding the catogorical features: Cleab subcategories

In [38]:

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_subcategories'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_csub_ohe = vectorizer.transform(X_train['clean_subcategories'].values)
X_cv_csub_ohe = vectorizer.transform(X_cv['clean_subcategories'].values)
X_test_csub_ohe = vectorizer.transform(X_test['clean_subcategories'].values)
print("After vectorizations")
print(X_train_csub_ohe.shape, y_train.shape)
print(X_cv_csub_ohe.shape, y_cv.shape)
print(X_test_csub_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)

cleansubCatVec=vectorizer.get_feature_names()
```

```
After vectorizations
(49041, 30) (49041,)
(24155, 30) (24155,)
(36052, 30) (36052,)
['appliedsciences', 'care_hunger', 'charactereducation', 'civics_government',
'college_careerprep', 'communityservice', 'earlydevelopment', 'economics', 'environmentalscience',
'esl', 'extracurricular', 'financialliteracy', 'foreignlanguages', 'gym_fitness',
'health_lifescience', 'health_wellness', 'history_geography', 'literacy', 'literature_writing', 'm
athematics', 'music', 'nutritioneducation', 'other', 'parentinvolvement', 'performingarts', 'socia
lsciences', 'specialneeds', 'teamsports', 'visualarts', 'warmth']
====================================================================================================
```

2.3 Make Data Model Ready: encoding essay, and project_title

Bag of Words

In [39]:

```python
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(min_df=10,ngram_range=(1,4))
vectorizer.fit(X_train['project_title'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_title_bow = vectorizer.transform(X_train['project_title'].values)
X_cv_title_bow = vectorizer.transform(X_cv['project_title'].values)
X_test_title_bow = vectorizer.transform(X_test['project_title'].values)
print("After vectorizations")
print(X_train_title_bow.shape, y_train.shape)
print(X_cv_title_bow.shape, y_cv.shape)
print(X_test_title_bow.shape, y_test.shape)
print("="*100)

projTitleBowVec=vectorizer.get_feature_names()
```

```
After vectorizations
(49041, 4097) (49041,)
(24155, 4097) (24155,)
(36052, 4097) (36052,)
====================================================================================================
```

In [40]:

```python
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(min_df=10,ngram_range=(1,4))
vectorizer.fit(X_train['essay'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_essay_bow = vectorizer.transform(X_train['essay'].values)
X_cv_essay_bow = vectorizer.transform(X_cv['essay'].values)
X_test_essay_bow = vectorizer.transform(X_test['essay'].values)
print("After vectorizations")
print(X_train_essay_bow.shape, y_train.shape)
print(X_cv_essay_bow.shape, y_cv.shape)
print(X_test_essay_bow.shape, y_test.shape)
print("="*100)


projEssayBowVec=vectorizer.get_feature_names()
```

```
After vectorizations
(49041, 167185) (49041,)
(24155, 167185) (24155,)
(36052, 167185) (36052,)
====================================================================================================
```

TFIDF vectorizer

In [41]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=10)
vectorizer.fit(X_train['project_title'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_title_tfidf = vectorizer.transform(X_train['project_title'].values)
X_cv_title_tfidf = vectorizer.transform(X_cv['project_title'].values)
X_test_title_tfidf = vectorizer.transform(X_test['project_title'].values)
print("After vectorizations")
print(X_train_title_tfidf.shape, y_train.shape)
print(X_cv_title_tfidf.shape, y_cv.shape)
print(X_test_title_tfidf.shape, y_test.shape)
print("="*100)

projTitleTfidfVec=vectorizer.get_feature_names()
```

```
After vectorizations
(49041, 2074) (49041,)
(24155, 2074) (24155,)
(36052, 2074) (36052,)
====================================================================================================
```

In [42]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=10)
vectorizer.fit(X_train['essay'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_essay_tfidf = vectorizer.transform(X_train['essay'].values)
X_cv_essay_tfidf = vectorizer.transform(X_cv['essay'].values)
X_test_essay_tfidf = vectorizer.transform(X_test['essay'].values)
print("After vectorizations")
print(X_train_essay_tfidf.shape, y_train.shape)
print(X_cv_essay_tfidf.shape, y_cv.shape)
print(X_test_essay_tfidf.shape, y_test.shape)
print("="*100)


#print(vectorizer.get_feature_names())

projEssayTfidfVec=vectorizer.get_feature_names()
```

```
After vectorizations
(49041, 12169) (49041,)
(24155, 12169) (24155,)
(36052, 12169) (36052,)
====================================================================================================
```

2.4.1 Applying Naive Bayes on BOW, SET 1

In [43]:

```
#merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr_bow = hstack((X_train_qty_norm, X_train_ppp_norm, X_train_price_norm, X_train_state_ohe, X_tra
in_grade_ohe,X_train_teacher_ohe,
X_train_ccat_ohe,X_train_csub_ohe,X_train_title_bow,X_train_essay_bow)).tocsr()

X_cr_bow = hstack((X_cv_qty_norm, X_cv_ppp_norm, X_cv_price_norm, X_cv_state_ohe, X_cv_grade_ohe,X_
cv_teacher_ohe,
X_cv_ccat_ohe,X_cv_csub_ohe,X_cv_title_bow,X_cv_essay_bow)).tocsr()

X_te_bow = hstack((X_test_qty_norm, X_test_ppp_norm, X_test_price_norm, X_test_state_ohe,
X_test_grade_ohe,X_test_teacher_ohe,
X_test_ccat_ohe,X_test_csub_ohe,X_test_title_bow,X_test_essay_bow)).tocsr()
print("Final Data matrix")
print(X_tr_bow.shape, y_train.shape)
print(X_cr_bow.shape, y_cv.shape)
print(X_te_bow.shape, y_test.shape)
print("="*100)
```

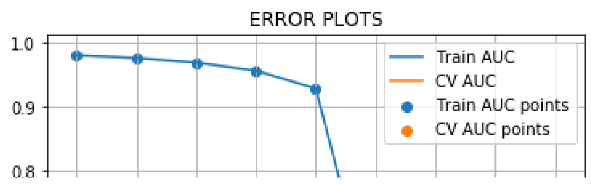
```
Final Data matrix
(49041, 171384) (49041,)
(24155, 171384) (24155,)
(36052, 171384) (36052,)
====================================================================================================
```

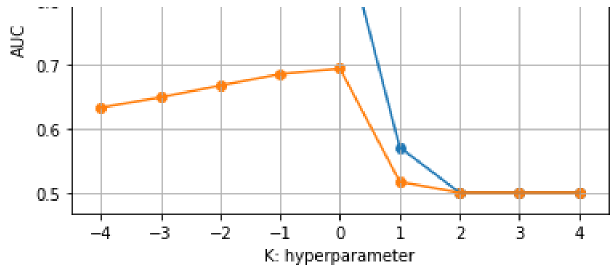
In [44]:

```
def batch_predict(clf, data):
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the posi
tive class
    # not the predicted outputs
    y_data_pred = []
    tr_loop = data.shape[0] - data.shape[0]%1000
    # consider you X_tr shape is 49041, then your cr_loop will be 49041 - 49041%1000 = 49000
    # in this for loop we will iterate unti the last 1000 multiplier
    for i in range(0, tr_loop, 1000):
        y_data_pred.extend(clf.predict_proba(data[i:i+1000])[:,1])
    # we will be predicting for the last data points
    y_data_pred.extend(clf.predict_proba(data[tr_loop:])[:,1])
    return y_data_pred
```

In [45]:

```
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_auc_score
from sklearn.naive_bayes import MultinomialNB
import numpy as np
"""
y_true : array, shape = [n_samples] or [n_samples, n_classes]
True binary labels or binary label indicators.
y_score : array, shape = [n_samples] or [n_samples, n_classes]
Target scores, can either be probability estimates of the positive class, confidence values, or no
n-thresholded measure of
decisions (as returned by "decision_function" on some classifiers).
For binary y_true, y_score is supposed to be the score of the class with greater label.
"""
train_auc = []
cv_auc = []
K = [10**-4,10**-3,10**-2,10**-1,10**0,10**1,10**2,10**3,10**4]
for i in K:
    neigh = MultinomialNB(alpha=i,class_prior = [0.5, 0.5])
    neigh.fit(X_tr_bow, y_train)
    y_train_pred = neigh.predict_proba( X_tr_bow)[:,1]
    y_cv_pred = neigh.predict_proba( X_cr_bow)[:,1]
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the posi
tive class
    # not the predicted outputs
    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))
plt.plot(np.log10(K), train_auc, label='Train AUC')
plt.plot(np.log10(K), cv_auc, label='CV AUC')
plt.scatter(np.log10(K), train_auc, label='Train AUC points')
plt.scatter(np.log10(K), cv_auc, label='CV AUC points')
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



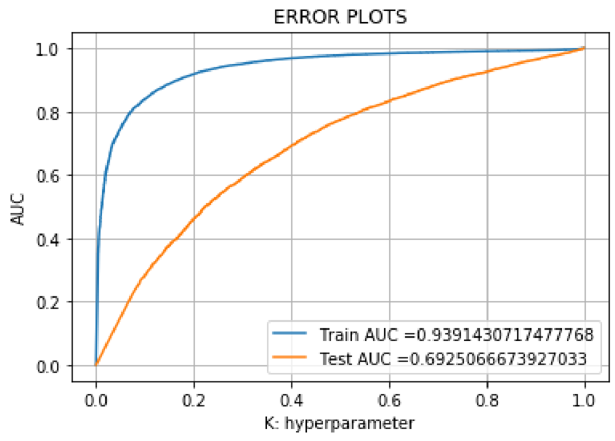


In [124]:

```
best_k=0.5
```

In [125]:

```
#https://scikitlearn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.ro
rve
from sklearn.metrics import roc_curve, auc
neigh = MultinomialNB(alpha=best_k,class_prior = [0.5, 0.5] )
neigh.fit(X_tr_bow, y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the positive
class
# not the predicted outputs
y_train_pred = neigh.predict_proba(X_tr_bow)[:,1]
y_test_pred = neigh.predict_proba( X_te_bow)[:,1]
train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
plt.plot(train_fpr, train_tpr, label="Train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="Test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



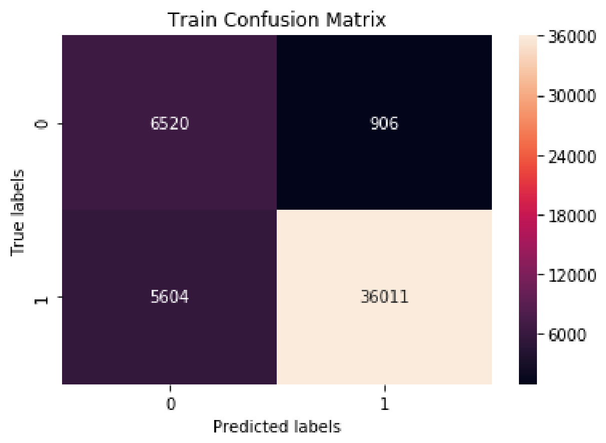
In [126]:

```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def predict(proba, threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very high
   # print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshold", np.round(t,3))
    predictions = []
    for i in proba:
        if i>=t:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
```

In [127]:

```
import seaborn as sn
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix


print("Train confusion matrix")
a=confusion_matrix(y_train, predict(y_train_pred, tr_thresholds, train_fpr, train_tpr))
ax= plt.subplot()
sns.heatmap(a, annot=True, ax = ax,fmt='g'); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Train Confusion Matrix');
```

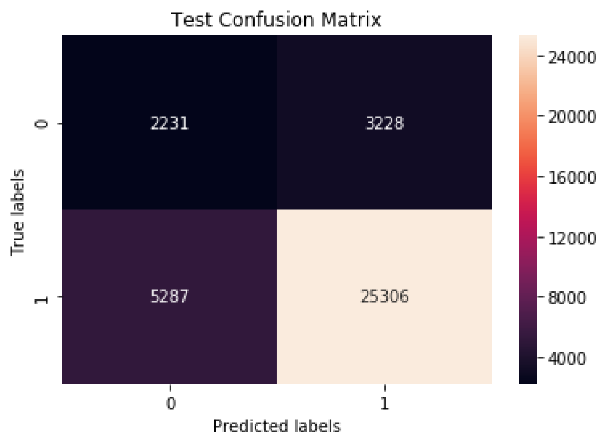
```
Train confusion matrix
```



In [128]:

```
import seaborn as sn
import matplotlib.pyplot as plt

print("Test confusion matrix")
b=confusion_matrix(y_test, predict(y_test_pred, tr_thresholds, test_fpr, test_tpr))
ax1= plt.subplot()
sns.heatmap(b, annot=True, ax = ax1,fmt='g'); #annot=True to annotate cells

# labels, title and ticks
ax1.set_xlabel('Predicted labels');
ax1.set_ylabel('True labels');
ax1.set_title('Test Confusion Matrix');
```

```
Test confusion matrix
```



In [97]:

```
from sklearn.naive_bayes import MultinomialNB
```

```
feature_names=[]
feature_names.extend(['quantity'])
feature_names.extend(['teacher_number_of_previously_posted_projects'])
feature_names.extend(['price'])
feature_names.extend(stateVec)
feature_names.extend(projGradeVec)
feature_names.extend(prefixteacherVec)
feature_names.extend(cleanCatVec)
feature_names.extend(cleansubCatVec)
feature_names.extend(projTitleBowVec)
feature_names.extend(projEssayBowVec)

len(feature_names)
```

Out[97]:

```
171384
```

2.4.1.1 Top 10 important features of positive class from SET 1

In [98]:

```
max_ind_pos=np.argsort((neigh.feature_log_prob_)[1])[::-1][0:10]
top_pos=np.take(feature_names,max_ind_pos)
print(top_pos)
```

```
['students' 'school' 'my' 'learning' 'classroom' 'the' 'they' 'not'
 'my students' 'learn']
```

2.4.1.2 Top 10 important features of negative class from SET 1

In [99]:

```
max_ind_neg=np.argsort((neigh.feature_log_prob_)[0])[::-1][0:10]
top_neg=np.take(feature_names,max_ind_neg)
print(top_neg)
```

```
['students' 'school' 'learning' 'my' 'classroom' 'not' 'learn' 'they'
 'the' 'help']
```

2.4.2 Applying KNN brute force on TFIDF, SET 2

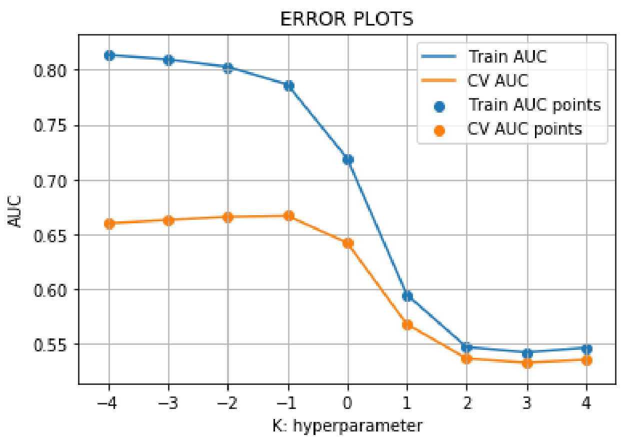
In [100]:

```
#merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr_tfidf = hstack((X_train_qty_norm, X_train_ppp_norm, X_train_price_norm, X_train_state_ohe, X_t
rain_grade_ohe,X_train_teacher_ohe,
X_train_ccat_ohe,X_train_csub_ohe,X_train_title_tfidf,X_train_essay_tfidf)).tocsr()

X_cr_tfidf = hstack((X_cv_qty_norm, X_cv_ppp_norm, X_cv_price_norm, X_cv_state_ohe, X_cv_grade_ohe,
X_cv_teacher_ohe,
X_cv_ccat_ohe,X_cv_csub_ohe,X_cv_title_tfidf,X_cv_essay_tfidf)).tocsr()

X_te_tfidf = hstack((X_test_qty_norm, X_test_ppp_norm, X_test_price_norm, X_test_state_ohe,
X_test_grade_ohe,X_test_teacher_ohe,
X_test_ccat_ohe,X_test_csub_ohe,X_test_title_tfidf,X_test_essay_tfidf)).tocsr()
print("Final Data matrix")
print(X_tr_tfidf.shape, y_train.shape)
print(X_cr_tfidf.shape, y_cv.shape)
print(X_te_tfidf.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix
(49041, 14345) (49041,)
(24155, 14345) (24155,)
(36052, 14345) (36052,)
====================================================================================================
```

In [101]:

```
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_auc_score
from sklearn.naive_bayes import MultinomialNB
import numpy as np
"""
y_true : array, shape = [n_samples] or [n_samples, n_classes]
True binary labels or binary label indicators.
y_score : array, shape = [n_samples] or [n_samples, n_classes]
Target scores, can either be probability estimates of the positive class, confidence values, or no
n-thresholded measure of
decisions (as returned by “decision_function” on some classifiers).
For binary y_true, y_score is supposed to be the score of the class with greater label.
"""
train_auc = []
cv_auc = []
K = [10**-4,10**-3,10**-2,10**-1,10**0,10**1,10**2,10**3,10**4]
for i in K:
    neigh = MultinomialNB(alpha=i)
    neigh.fit(X_tr_tfidf, y_train)
    y_train_pred = neigh.predict_proba( X_tr_tfidf)[:,1]
    y_cv_pred = neigh.predict_proba( X_cr_tfidf)[:,1]
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the posi
tive class
    # not the predicted outputs
    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))
plt.plot(np.log10(K), train_auc, label='Train AUC')
plt.plot(np.log10(K), cv_auc, label='CV AUC')
plt.scatter(np.log10(K), train_auc, label='Train AUC points')
plt.scatter(np.log10(K), cv_auc, label='CV AUC points')
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



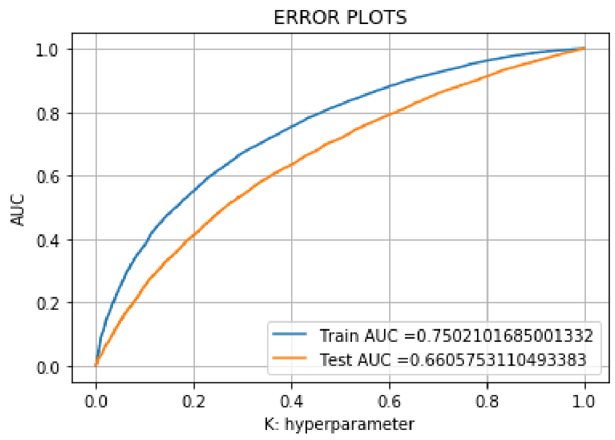
In [117]:

```
best_k=0.5
```

In [118]:

```
#https://scikitlearn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.rc
rve
from sklearn.metrics import roc_curve, auc
neigh = MultinomialNB(alpha=best_k)
neigh.fit(X_tr_tfidf, y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the positive
class
# not the predicted outputs
```

```
y_train_pred = neigh.predict_proba( X_tr_tfidf)[:,1]
y_test_pred = neigh.predict_proba( X_te_tfidf)[:,1]
train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
plt.plot(train_fpr, train_tpr, label="Train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="Test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



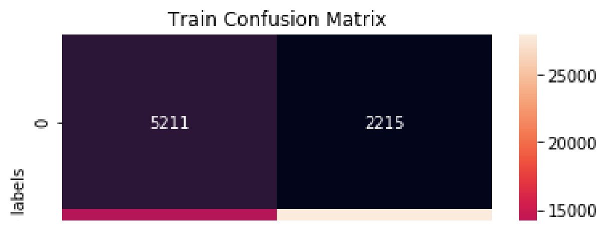
In [104]:

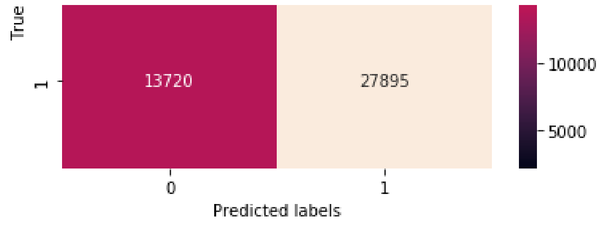
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def predict(proba, threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshold", np.round(t,3))
    predictions = []
    for i in proba:
        if i>=t:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
```

In [105]:

```
print("="*100)
from sklearn.metrics import confusion_matrix
print("Train confusion matrix")
c=confusion_matrix(y_train, predict(y_train_pred, tr_thresholds, train_fpr, train_tpr))

ax= plt.subplot()
sns.heatmap(c, annot=True, ax = ax,fmt='g'); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Train Confusion Matrix');
```

```
====================================================================================================

Train confusion matrix
the maximum value of tpr*(1-fpr) 0.4703732277903405 for threshold 0.866
```

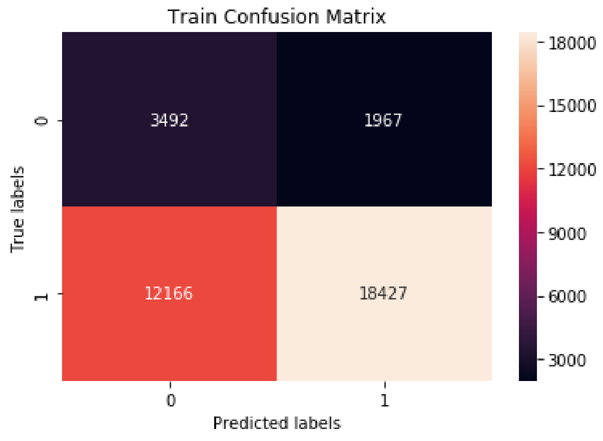




In [106]:

```
print("Test confusion matrix")
d=confusion_matrix(y_test, predict(y_test_pred, tr_thresholds, test_fpr, test_tpr))


ax= plt.subplot()
sns.heatmap(d, annot=True, ax = ax,fmt='g'); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Train Confusion Matrix');
```

```
Test confusion matrix
the maximum value of tpr*(1-fpr) 0.38579055882187874 for threshold 0.882
```



In [107]:

```
from sklearn.naive_bayes import MultinomialNB

feature_names1=[]
feature_names1.extend(['quantity'])
feature_names1.extend(['teacher_number_of_previously_posted_projects'])
feature_names1.extend(['price'])
feature_names1.extend(stateVec)
feature_names1.extend(projGradeVec)
feature_names1.extend(prefixteacherVec)
feature_names1.extend(cleanCatVec)
feature_names1.extend(cleansubCatVec)
feature_names1.extend(projTitleTfidfVec)
feature_names1.extend(projEssayTfidfVec)

len(feature_names1)
```

Out[107]:

```
14345
```

2.4.2.1 Top 10 important features of positive class from SET 2

In [108]:

```
max_ind_pos=np.argsort((neigh.feature_log_prob_)[1])[::-1][0:10]
top_pos=np.take(feature_names,max_ind_pos)
print(top_pos)
```

```
['quantity' 'price' 'teacher_number_of_previously_posted_projects' 'Mrs'
 'literacy_language' 'grades_prek_2' 'math_science' 'Ms' 'grades_3_5'
 'literacy']
```

2.4.2.2 Top 10 important features of negative class from SET 2

In [109]:

```
max_ind_neg=np.argsort((neigh.feature_log_prob_)[0])[::-1][0:10]
top_neg=np.take(feature_names,max_ind_neg)
print(top_neg)
```

```
['quantity' 'price' 'teacher_number_of_previously_posted_projects' 'Mrs'
 'literacy_language' 'grades_prek_2' 'math_science' 'Ms' 'grades_3_5'
 'literacy']
```

Summary

In [129]:

```
from prettytable import PrettyTable

x = PrettyTable()

x.field_names = ["Vectorizer", "Model", "Hyperparamter", "AUC"]

x.add_row(["BOW", "Naive Bayes", 0.5, 0.692])
x.add_row(["TFIDF", "Naive Bayes", 0.7, 0.6605])

print(x)
```

```
+------------+-------------+---------------+--------+
| Vectorizer |    Model    | Hyperparamter |  AUC   |
+------------+-------------+---------------+--------+
|    BOW     | Naive Bayes |      0.5      | 0.692  |
|   TFIDF    | Naive Bayes |      0.7      | 0.6605 |
+------------+-------------+---------------+--------+
```

In [ ]: