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	De
project_id	A unique identifier for the proposed project. **Example:**
project_title	Title of the project. **E** • Art Will Make You • First Gr
project_grade_category	Grade level of students for which the project is targeted. One of the enumerate • Grades • Gra • Gra • Grad
project_subject_categories	One or more (comma-separated) subject categories for the projec following enumerated list • Applied L • Care & • Health & • History & • Literacy & L • Math & • Music & T • Specia • **E** • Music & T • Literacy & Language, Math &
school_state	State where school is located (Two-letter U.S. p (https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations#Posta **Exar**
project_subject_subcategories	One or more (comma-separated) subject subcategories for th **E** • L • Literature & Writing, Social S
project_resource_summary	An explanation of the resources needed for the project. **E** • My students need hands on literacy materials to sensory
project_essay_1	First applicat
project_essay_2	Second applicat
project_essay_3	Third applicat
project_essay_4	Fourth applicat
project_submitted_datetime	Datetime when project application was submitted. **Example:** 201 12:43
teacher_id	A unique identifier for the teacher of the proposed project. bdf8baa8fedef6bfeec7ae4ff

Feature	De
	Teacher's title. One of the following enumerate
`teacher_prefix`	• • • • • • T
`teacher_number_of_previously_posted_projects`	Number of project applications previously submitted by the sam **Exa**

* 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

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_3:__ "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 [1]:

```
%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
```

```
C:\Users\RASHU TYAGI\Anaconda3\lib\site-packages\smart_open\ssh.py:34: Use
rWarning: paramiko missing, opening SSH/SCP/SFTP paths will be disabled.
`pip install paramiko` to suppress
  warnings.warn('paramiko missing, opening SSH/SCP/SFTP paths will be disa
bled.  `pip install paramiko` to suppress')
```

1.1 Reading Data

In [2]:

```
project_data = pd.read_csv(r'D:\Rashu Studies\AppliedAICourse\Assignments\Mandatory Ass
ignments\Mandatory Assignment 3 Donors Choose KNN\train_data.csv')
resource_data = pd.read_csv(r'D:\Rashu Studies\AppliedAICourse\Assignments\Mandatory As
signments\Mandatory Assignment 3 Donors Choose KNN\resources.csv')
```

In [3]:

```
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 [4]:

```
# how to replace elements in list python: https://stackoverflow.com/a/2582163/4084039
cols = ['Date' if x=='project_submitted_datetime' else x for x in list(project_data.col
umns)]


#sort dataframe based on time pandas python: https://stackoverflow.com/a/49702492/40840
39
project_data['Date'] = pd.to_datetime(project_data['project_submitted_datetime'])
project_data.drop('project_submitted_datetime', axis=1, inplace=True)
project_data.sort_values(by=['Date'], inplace=True)


# how to reorder columns pandas python: https://stackoverflow.com/a/13148611/4084039
project_data = project_data[cols]


project_data.head(2)
```

Out[4]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state
55660	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs.	CA
76127	37728	p043609	3f60494c61921b3b43ab61bdde2904df	Ms.	UT

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

NOW THE MOST IMPORTANT THING HERE IS THAT YOU SHOULD SPLIT OUR DATA INTO TRAIN AND TEST BEFORE APPLYING ANY FIT TECHNIQUE LIKE BOW OR TFIDF BECAUSE OTHEREWISE THERE WILL BE DATA LEAKAGE PROBLEM.ALSO FOR PREPROCESSING LIKE STANDARDIZATION AND NORMALIZATION ALSO WE SHOULD KEEP IN MIND THAT TRAIN TEST SPLIT SHOULD BE DONE BEFORE APPLYING THOSE PREPROCESSING TECHNIQUES

In [6]:

```
# REFER THIS SOUNDCLOUD LINK : https://soundcloud.com/applied-ai-course/leakage-bow-and
-tfidf
```

Train_Test Split

In [7]:

```
# train test split
# note that here This stratify parameter makes a split so that the proportion of values
in the sample produced will be the same as the proportion of values provided to paramet
er stratify.
#For example, if variable y is a binary categorical variable with values 0 and 1 and th
ere are 25% of zeros and 75% of ones, stratify=y will make sure that your random split
 has 25% of 0's and 75% of 1's.

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(project_data, project_data['project
_is_approved'], test_size=0.33, stratify = project_data['project_is_approved'])

# now getting the crossvalidation data from our train data
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.33, strat
ify=y_train)
```

In [8]:

```
# Now we will be removing the column "project_is_approved" because that is the only one
which our model needs to predict

X_train.drop(['project_is_approved'], axis=1, inplace=True)
X_test.drop(['project_is_approved'], axis=1, inplace=True)
X_cv.drop(['project_is_approved'], axis=1, inplace=True)
```

Now we will do all kind of preprocessing required for the train data ,test data,crossvalidation data separately

FOR TRAIN DATA

Preprocessing of `project_subject_categories'

In [9]:

```
catogories = list(X_train['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
ng
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth
on
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", "Warmt
h", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "M
ath & Science"=> "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:"M
ath & Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spa
ces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

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

from collections import Counter
my_counter = Counter()
for word in X_train['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]))
```

Preprocessing of project_subject_subcategories

In [10]:

```
sub_catogories = list(X_train['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

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

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", "Warmt
h", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "M
ath & Science"=> "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:"M
ath & Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the trailing spa
ces
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())

X_train['clean_subcategories'] = sub_cat_list
X_train.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 X_train['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]))
```

Text preprocessing

In [11]:

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

In [12]:

```
X_train.head(2)
```

Out[12]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state
9102	7778	p078715	1a735d2b91bf93c7e14d18fea8b08c84	Mr.	MI
79817	66165	p087329	06afc39f867e1a106904405b4401ca0f	Ms.	MT

In [13]:

```
# printing some random reviews
print(X_train['essay'].values[0])
print("="*50)
print(X_train['essay'].values[150])
print("="*50)
print(X_train['essay'].values[1000])
print("="*50)
print(X_train['essay'].values[20000])
print("="*50)
print(X_train['essay'].values[9999])
print("="*50)
```

My step team is composed of a bunch of carefree and eager to learn young p
eople. They range from grades 9-12th. These students are at risk students
that deal with homelessness and/or lack of food after school. These studen
ts love to volunteer and help others. They participate in community servic
e projects. \r\n\r\nOur school is a unique comprehensive high school which
offers three examination programs within the school: Mathematics, Science
and Technology (MSAT), Center for International Studies and Commerce (CIS
C) and College Preparatory Liberal Arts (CPLA). Our school is also the fir
st high school designed and certified as a green energy building. The inte
rnationally recognized Marching Band, Dance, and Vocal music departments,
along with our championship Chess, Robotics teams, the Student Council, De
bate Team and medical club are just a few of the numerous extracurricular
offerings.My students are in need of running shoes for Spring and Summer s
easons. Their families can not afford to buy them these basic items. My
students usually wear boots all year around because they do not have prope
r shoes. I want to ensure that each one of my students will have a good p
air of running shoes. \r\nEvery student should have a pair of shoes that
fit and are comfortable. Some of these students are homeless and do not h
ave anyone to depend on. These students often go from house to house to s
leep on a couch or a floor. I want my students self esteem and confidence
to be boosted so they can excel in school.nannan
==
Our school district is a low-income district, so we have quite a few stude
nts who do not have basic needs at home. Low income students at our school
are fed through our free and reduced lunch program, but we hope to provide
for them on days that we do not come to school. For some of them, the onl
y nutrition they get is at school. Many times they also do not have basic
hygiene products or school supplies at home.Every Friday we pack backpacks
full of food, toiletries and school supplies. We include several differen
t food items: breakfast items, easy dinner items, juice, snacks and fresh
fruit. We also include tooth brushes, toothpaste, hair brushes, shampoo,
conditioner, wipes, bar soap, deodorant, notebook paper, pens and pencils.
On Fridays the backpacks go home with students who do not have basic neces
sities at home. Some of our students have siblings so we double pack some
of the backpacks to provide nutrition for them as well. Each week they re
turn the bags so that we can refill them for the next week. We also keep c
lothing items on hand for when they need that.nannan
==
As a Special Education teacher in a low-income/high poverty school distric
t, my students are faced with several challenges both in and out of the cl
assroom. Despite the many challenges they face at home, I am looking to cr
eate a safe, nurturing and comforting environment in our classroom. \r\n
\r\nA learning environment that is calm, inviting, and safe is imperative
to the educational growth of my students. \r\n\r\nMy classroom is comprise
d of students with a wide range disabilities, many of which result in high
sensory needs. Many of my students require sensory breaks in order to pro
cess information and learn through out the day. Simple things such as spac
ial barriers, textures, and colors help tremendously in calming students a
nd making learning much more accessible.My project will help foster learni
ng in the classroom by providing my students with a comfortable and cozy l
earning environment. \r\n\r\nWe spend a large majority of our day sitting
on the floor. \r\nThis carpet would allow for my students to have plenty o
f space available to them during classroom discussions, small group work,
collaborative partner work, read aloud, and whole class activities. We ar
e an ICT classroom, therefore we need to have two separate meeting areas f
or our students. \r\n\r\nRight now my classroom is colorful and vibrant,
but we are lacking the comfy learning spaces and organization my students
need. For many of my students, school is the safe, loving place they need
to grow as learners. This project will allow my students a variety of opti
ons for seating while learning.\r\n\r\n\r\nThis project will help my stude
nts be comfortable and focused on learning. Sitting together in this space

will encourage cooperative learning in a flexible way. Students can turn t
o each other to have discussions, or they can move around easily.\r\n\r\n
\r\nThe students will be able to enjoy the daily activities and routines w
hile simultaneously feeling free to wiggle and move as they please. Being
able to wiggle and move as needed will decrease the need for individual mo
vement breaks which not only draw attention to a student's differences, bu
t also removes the student from valuable instructional time. By providing
the students with the movement opportunities that they deserve, instructio
nal time and engagement will be maximized. Thank you so much for helping f
und our project!\r\n\r\nnannan

==========

Throughout the year, I introduce various artists, art periods, and cultura
l art to inspire my students. I also expose them to as many different art
mediums, as possible.\r\n\r\nMy students are a diverse group of elementary
students, who love learning about new ways they can use art to express the
mselves.\r\n\r\nThey are a great group, who I want to give a variety of op
portunities to. \r\n\r\nMy students are 70% economically disadvantaged, me
aning their families can't afford extra supplies and technology for them,
even if they would like to. We are really a school full of love and charis
ma, but are not rich, financially.The clay, glazes, and texture tools woul
d be used by all of my classes, throughout the year. Due to limited budge
t, it is hard to continue to afford clay. My budget was cut, and it is har
d to afford expensive materials like clay and glaze. I would love for them
to experience different materials, but my ability is hindered by budget. T
he clay lesson could easily be tied to a science lesson on how clay comes
from the ground and even the state that clay and glaze go through. The art
terminology discussed would be: clay, texture, slip, glaze, slab, coil, pi
nch pot. We would also use peer critiques, as a written portion of the pro
ject.\r\n\r\nThe donations will expand the art opportunities for my studen
ts, by allowing them to use different materials.\r\nIt will also give my s
tudents the opportunity to learn about another culture and how they utiliz
ed clay since prehistoric times.nannan

==========

The students at my school come from various backgrounds. They primarily co
me from homes that have an extreme interest in education. Even though not
all of the families are wealthy in financial terms, the parents of this co
mmunity are very supportive in sharing their wealth of love and concern fo
r all of the children at our school. I believe that this is the reason my
students are so motivated and have such a desire to learn and do well.\r\n
\r\nThese students are imaginative and not afraid to try new things!\r\n\r
\n\"Recess is boring.\" \"There's never anything to do outside.\" \"Can't
I just stay inside.\" Yes, these have all been said by fourth graders. Why
are children trying to avoid recess? Their comments included lack of playg
round space, friends, and sports equipment.\r\n\r\nThis project should be
called operation recess because it tackles two of these common recess prob
lems, friends and sports equipment.\r\n\r\nLast year, some amazing members
of our community donated buddy benches. These are benches designed to help
students who don't have a friend be noticed and fellowshipped on the playg
round. \r\n\r\nMy students, last May, had the idea that the buddy bench sh
ould have its own playground equipment. The equipment would help students
have something to do with the buddies who spend recesses time on the benc
h.\r\n\r\nTheir project will take a few stages before it becomes completel
y student driven. It will include setting up the rules of borrowing the sp
orts balls and playground equiptment, making sure equiptment is used with
the intended buddies from the bench, and creating a routine clean up proce
dure so all of the balls, jump ropes, and scoops can be used over and over
again.\r\n\r\nBy donating to this project you are helping a large portion
of our school come together, learn important social skills, and help each
other to exercise.nannan

==========

In [14]:

```
# creating a function named as decontracted which does the job of decontraction

# 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 [15]:

```
sent = decontracted(X_train['essay'].values[20000])
print(sent)
print("="*50)
```

```
Throughout the year, I introduce various artists, art periods, and cultura
l art to inspire my students. I also expose them to as many different art
mediums, as possible.\r\n\r\nMy students are a diverse group of elementary
students, who love learning about new ways they can use art to express the
mselves.\r\n\r\nThey are a great group, who I want to give a variety of op
portunities to. \r\n\r\nMy students are 70% economically disadvantaged, me
aning their families can not afford extra supplies and technology for the
m, even if they would like to. We are really a school full of love and cha
risma, but are not rich, financially.The clay, glazes, and texture tools w
ould be used by all of my classes, throughout the year. Due to limited bud
get, it is hard to continue to afford clay. My budget was cut, and it is h
ard to afford expensive materials like clay and glaze. I would love for th
em to experience different materials, but my ability is hindered by budge
t. The clay lesson could easily be tied to a science lesson on how clay co
mes from the ground and even the state that clay and glaze go through. The
art terminology discussed would be: clay, texture, slip, glaze, slab, coi
l, pinch pot. We would also use peer critiques, as a written portion of th
e project.\r\n\r\nThe donations will expand the art opportunities for my s
tudents, by allowing them to use different materials.\r\nIt will also give
my students the opportunity to learn about another culture and how they ut
ilized clay since prehistoric times.nannan
==================================================
```

In [16]:

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

```
Throughout the year, I introduce various artists, art periods, and cultura
l art to inspire my students. I also expose them to as many different art
mediums, as possible.    My students are a diverse group of elementary stu
dents, who love learning about new ways they can use art to express themse
lves.    They are a great group, who I want to give a variety of opportuni
ties to.     My students are 70% economically disadvantaged, meaning their
families can not afford extra supplies and technology for them, even if th
ey would like to. We are really a school full of love and charisma, but ar
e not rich, financially.The clay, glazes, and texture tools would be used
by all of my classes, throughout the year. Due to limited budget, it is ha
rd to continue to afford clay. My budget was cut, and it is hard to afford
expensive materials like clay and glaze. I would love for them to experien
ce different materials, but my ability is hindered by budget. The clay les
son could easily be tied to a science lesson on how clay comes from the gr
ound and even the state that clay and glaze go through. The art terminolog
y discussed would be: clay, texture, slip, glaze, slab, coil, pinch pot. W
e would also use peer critiques, as a written portion of the project.    T
he donations will expand the art opportunities for my students, by allowin
g them to use different materials.  It will also give my students the oppo
rtunity to learn about another culture and how they utilized clay since pr
ehistoric times.nannan
```

In [17]:

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

```
Throughout the year I introduce various artists art periods and cultural a
rt to inspire my students I also expose them to as many different art medi
ums as possible My students are a diverse group of elementary students who
love learning about new ways they can use art to express themselves They a
re a great group who I want to give a variety of opportunities to My stude
nts are 70 economically disadvantaged meaning their families can not affor
d extra supplies and technology for them even if they would like to We are
really a school full of love and charisma but are not rich financially The
clay glazes and texture tools would be used by all of my classes throughou
t the year Due to limited budget it is hard to continue to afford clay My
budget was cut and it is hard to afford expensive materials like clay and
glaze I would love for them to experience different materials but my abili
ty is hindered by budget The clay lesson could easily be tied to a science
lesson on how clay comes from the ground and even the state that clay and
glaze go through The art terminology discussed would be clay texture slip
glaze slab coil pinch pot We would also use peer critiques as a written po
rtion of the project The donations will expand the art opportunities for m
y students by allowing them to use different materials It will also give m
y students the opportunity to learn about another culture and how they uti
lized clay since prehistoric times nannan
```

In [18]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
# because although they are in this list but they matter a lot because
# they change the meaning of the entire sentence.
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', 'each', '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', "doesn'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 [19]:

```
# Combining all the above preprocessing techniques for all the project essays
from tqdm import tqdm
preprocessed_essays_Train = []
# tqdm is for printing the status bar
for sentance in tqdm(X_train['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.lower() not in stopwords)
    preprocessed_essays_Train.append(sent.lower().strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 49041/49041 [00:22<00:00, 2215.75it/s]
```

In [20]:

```
# after preprocesing of project essays
preprocessed_essays_Train[20000]
```

Out[20]:

```
'throughout year introduce various artists art periods cultural art inspir
e students also expose many different art mediums possible students divers
e group elementary students love learning new ways use art express great g
roup want give variety opportunities students 70 economically disadvantage
d meaning families not afford extra supplies technology even would like re
ally school full love charisma not rich financially clay glazes texture to
ols would used classes throughout year due limited budget hard continue af
ford clay budget cut hard afford expensive materials like clay glaze would
love experience different materials ability hindered budget clay lesson co
uld easily tied science lesson clay comes ground even state clay glaze go
art terminology discussed would clay texture slip glaze slab coil pinch po
t would also use peer critiques written portion project donations expand a
rt opportunities students allowing use different materials also give stude
nts opportunity learn another culture utilized clay since prehistoric time
s nannan'
```

Preprocessing of project_title

Now we will simply apply the above preprocessing steps on the project title for the train data as well,as it is also a text feature

In [21]:

```
# printing some random titles.
print(X_train['project_title'].values[0])
print("="*50)
print(X_train['project_title'].values[150])
print("="*50)
print(X_train['project_title'].values[1000])
print("="*50)
print(X_train['project_title'].values[20000])
print("="*50)
print(X_train['project_title'].values[9999])
print("="*50)
```

```
Students In Need Of Shoes
==================================================
NOW (Nutrition on The Weekend) Backpacks
==================================================
We Need A Magic Carpet Ride!
==================================================
Clay projects
==================================================
Getting Off the Buddy Bench
==================================================
```

We have already written the preprocessing codes for different preprocessing approaches now we will simply use those codes on the project titles

In [22]:

```
preprocessed_project_titles_Train = []

for t in tqdm(X_train["project_title"]):
    title = decontracted(t)
    title = title.replace('\\r', ' ')
    title = title.replace('\\"', ' ')
    title = title.replace('\\n', ' ')
    title = re.sub('[^A-Za-z0-9]+', ' ', title)
    title = ' '.join(f for f in title.split() if f not in stopwords)
    preprocessed_project_titles_Train.append(title.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████████
████| 49041/49041 [00:01<00:00, 47871.44it/s]
```

In [23]:

```
# printing some random titles of train dataset after preprocessing

print(preprocessed_project_titles_Train[5000])
print("="*50)
print(preprocessed_project_titles_Train[7000])
print("="*50)
print(preprocessed_project_titles_Train[10000])
print("="*50)
print(preprocessed_project_titles_Train[45000])
print("="*50)
print(preprocessed_project_titles_Train[22000])
print("="*50)
```

```
recess fun
==================================================
coding robots using legos
==================================================
published works
==================================================
lesson lion den stress kit survival
==================================================
comfort functional seating
==================================================
```

Test Data

Preprocessing of project_subject_categories

In [24]:

```
catogories = list(X_test['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
ng
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth
on
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", "Warmt
h", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "M
ath & Science"=> "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:"M
ath & Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spa
ces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

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

from collections import Counter
my_counter = Counter()
for word in X_test['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]))
```

Preprocessing of project_subject_subcategories

In [25]:

```
sub_catogories = list(X_test['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

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

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", "Warmt
h", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "M
ath & Science"=> "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:"M
ath & Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the trailing spa
ces
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())

X_test['clean_subcategories'] = sub_cat_list
X_test.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 X_test['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]))
```

Text Preprocessing

In [26]:

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

In [27]:

```
X_test.head(5)
```

Out[27]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state
50781	150781	p006721	dfc5a6eb5a72671a615c0ff3930e4e2f	Ms.	TX
100297	32904	p112089	92e3092133ef4a261fc527fee7f00523	Mrs.	DE
91134	64175	p053428	bdc47b9c28c780c9631d127409c29b24	Mrs.	TX
20208	62840	p073433	7c4b1cb03fb75a38686bdb939ad0bf8b	Ms.	MD
86363	109390	p046930	873c57314227c8400311a7fcb35db68c	Ms.	CA

In [28]:

```
# printing some random reviews
print(X_test['essay'].values[0])
print("="*50)
print(X_test['essay'].values[150])
print("="*50)
print(X_test['essay'].values[1000])
print("="*50)
print(X_test['essay'].values[20000])
print("="*50)
print(X_test['essay'].values[9999])
print("="*50)
```

There is no bigger honor, or challenge, than welcoming children into schoo l for the very first time. Seeing their eager, optimistic, and smiling fac es everyday brings me overwhelming joy and similar feelings. I am also sli ghtly terrified.\r\n\r\nIn the one environment I can teach them to laugh, love, and learn, but I face external problems that inhibit my ability to d o so. Serving as their kindergarten teacher in a low income school of Plea sant Grove, Dallas my students and I are faced with challenges many of the m shouldn't even know exist yet.\r\n\r\nMy kids are future doctors, lawyer s, and business owners of America. My kids are going to accomplish amazing things this year. My kids will not let their socioeconomic status define t hem, but I need your help to make this a reality for them.In a world that can seem ever constraining and limited, the power of imagination can empow er and yield endless benefits. Donating to this project will allow my stud ents to develop a love and passion for reading by allowing me to purchase classroom library books, a listening center and books for guided reading. \r\n\r\nReading gives us someplace to go when we have to stay where we ar e. This is an opportunity to change kids lives in a way bigger than any pe ncil, packet of problems, or video ever could! \r\n\r\nAs their teacher th ese books allow my amazing kids to engage their brains and passion for lea rning even when I'm not by their side. Please, help me help you by develop ing our future leaders by laying the foundation for a life of learning, im agination, and discovery.nannan

==========

My second grade students are innovative, creative, generous, and very givi ng. They want to learn and do their best. They have high expectations of themselves. I too, have very high expectations for them. I want them to try hard, persevere, challenge themselves, and to be kind to one another. \r\nThe school I teach at is a Title I school in California which is facin g tremendous budget cuts. 85% of our students are enrolled in the free lun ch program, and approximately 60% are English language learners. For many students, their parents are unable to help them with their reading and mat h. They cannot give them the support when needed. Therefore, the responsib ility of giving my students access to the skills needed often falls solely on the teacher and school.The amount of academic rigor my students are giv en on a daily basis to meet our state's challenging standards is difficult for my students. I must find ways for them to find joy in their day. I can easily do this with a quick art lesson. My projects can be something as si mple as a directed draw to a more creative project that teaches students a more specific skill. Currently, we are learning about making our projects more realistic by blending our colors. Colored pencils are a wonderful art medium we can use to practice this skill. Currently my students are scramb ling for colors and paper and have a lot of wait time while they share the colors they need. I would love for each of my students to have one set of colored pencils to share with a partner. With your donations, I will be ab le to provide my students with not only beautiful art projects for them to take home but an enormous smile as well.nannan

==========

The students in my classroom attend what is classified as a Title 1 high s chool (highest poverty), and our school is home to one of the largest stud ent populations in the Columbus Metro area. Many of my students have had I ndividualized Education Programs developed for them, which qualifies them for special education; others are enrolled in as English Language Learner s, which means English is not their most fluent form of communication. At least one quarter of the students who attend the school at which I teach w ill experience one or more music classes with me at some point during thei r high school careers.\r\n\"Music can change the world because it can chan ge people.\" - Bono. This is a quote that I wholeheartedly believe in and teach by every day. Music and music classes are often an escape for studen ts of all ages, especially once they reach their teenage years when life c an become more stressful than it was before. My students not only learn fr om a curriculum, but they also get to experience a class that deals direct

ly with human emotions and helps to get them excited to come to school eve
ry day.The single most important material for choral instruction is the pi
ano. Not only is it a necessary tool to accompany choirs during performanc
es, but it is also absolutely critical to possess an working one for daily
musical instruction. The choirs I teach are in desperate need of a new pia
no in order to be able to learn and perform effectively.\r\nThis piano wil
l be used every single day during each class period for the majority of th
e instructional time. Our current piano does not stay in tune for long, d
oes not have working pedals, and has a broken key cover that promotes furt
her damage to the instrument. A new piano will allow my students to confid
ently give their best performances in front of their friends and family at
the end of each concert cycle.nannan
==
My studio classroom is living entity, just like the artwork its' young art
ists create. They are learning to think and behave just like artists d
o.\r\n\r\nMy students come from a wide variety of backgrounds and home liv
es. They are eager to learn and to create artwork that has meaning or a c
onnection to them. They work hard and create amazing, authentic artwork.
I, as a teacher and an artist, am proud to not only display at school, but
to brag about to others and on social media.Students in my art classes cre
ate artworks that focus around a big idea--ideas that showcase the actions
that artists do, like artists steal, where they learned about copyright an
d how to properly use the work of others in their own work. Students lear
n about the big idea, then following the design process, they figure out h
ow to use what they have learned to bring their own connection into artwor
ks. They choose the subject matter. They choose the material or material
s to work with. They choose the size. They make the creative decisions f
rom beginning to end. This year, many students are very interested in usi
ng paint, both acrylic and oil, to bring their ideas to life. And, having
a variety of canvas sizes help the students to make choices just like the
artists they are learning to behave and think like.\r\n\r\nHaving choice i
n the classroom makes a huge difference in how my students become artists.
Being able to decide if using paint and what size canvas to work on is jus
t one step into thinking like an artist and becoming an artist. It is all
part or the critical thinking process, which is a skill they can transfer
to other aspects of their lives beyond the art studio.nannan
==
It is our school's deep belief that ALL children can achieve academic exce
llence through intellectual, creative, and physical challenges, enabling t
hem to function as productive and successful citizens in a changing societ
y. \r\n\r\nThe strength our students muster to perservere is deserves noti
ce. Despite facing significant challenges, they continue to find ways to o
vercome and thrive. As a Title 1 School, our students battle each day the
effects of poverty and lack of resources. Our students persevere each day
towards excellence.\r\n\r\n\r\n\r\n\r\nThe purpose of this project is to b
ring yoga and mindfulness as part of the everyday happenings in the classr
oom. Yoga and mindfulness techniques will be used as alternatives to deten
tion and punitive punishments. Rather than sit in detention, students can
practice these strategies in order to develop strategies for dealing with
stressors. These strategies can also be used during counseling interaction
s with students.\r\n\r\nMindfulness and yoga can be a tool to help student
s self-regulate, deescalate and destress. What we have noticed is the high
recidivism rate with detention and suspensions. Students who are required
to attend detention more than often find themselves back in detention. Wit
hout a proactive, positive strategy for dealing with stressors, students w
ill cycle in and out of punitive disciplining. \r\n\r\nUsing yoga and mind
fulness techniques within the school day can also prevent over disciplinin
g of students by providing educators strategies for engaging with students
in positive ways. Furthermore, when students are able to apply proactive m
indfulness, they can effectively concentrate and retain focus on academic

```
s, thus increasing the potential for achievement.\r\n\r\nnannan
==================================================
```

In [29]:

```
# creating a function named as decontracted which does the job of decontraction

# 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 [30]:

```
sent = decontracted(X_test['essay'].values[20000])
print(sent)
print("="*50)
```

```
My studio classroom is living entity, just like the artwork its' young art
ists create.  They are learning to think and behave just like artists d
o.\r\n\r\nMy students come from a wide variety of backgrounds and home liv
es.  They are eager to learn and to create artwork that has meaning or a c
onnection to them.  They work hard and create amazing, authentic artwork.
I, as a teacher and an artist, am proud to not only display at school, but
to brag about to others and on social media.Students in my art classes cre
ate artworks that focus around a big idea--ideas that showcase the actions
that artists do, like artists steal, where they learned about copyright an
d how to properly use the work of others in their own work.  Students lear
n about the big idea, then following the design process, they figure out h
ow to use what they have learned to bring their own connection into artwor
ks.  They choose the subject matter.  They choose the material or material
s to work with.  They choose the size.  They make the creative decisions f
rom beginning to end.  This year, many students are very interested in usi
ng paint, both acrylic and oil, to bring their ideas to life.  And, having
a variety of canvas sizes help the students to make choices just like the
artists they are learning to behave and think like.\r\n\r\nHaving choice i
n the classroom makes a huge difference in how my students become artists.
Being able to decide if using paint and what size canvas to work on is jus
t one step into thinking like an artist and becoming an artist.  It is all
part or the critical thinking process, which is a skill  they can transfer
to other aspects of their lives beyond the art studio.nannan
==================================================
```

In [31]:

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

```
My studio classroom is living entity, just like the artwork its' young art
ists create.  They are learning to think and behave just like artists do.
My students come from a wide variety of backgrounds and home lives.  They
are eager to learn and to create artwork that has meaning or a connection
to them.  They work hard and create amazing, authentic artwork.  I, as a t
eacher and an artist, am proud to not only display at school, but to brag
about to others and on social media.Students in my art classes create artw
orks that focus around a big idea--ideas that showcase the actions that ar
tists do, like artists steal, where they learned about copyright and how t
o properly use the work of others in their own work.  Students learn about
the big idea, then following the design process, they figure out how to us
e what they have learned to bring their own connection into artworks.  The
y choose the subject matter.  They choose the material or materials to wor
k with.  They choose the size.  They make the creative decisions from begi
nning to end.  This year, many students are very interested in using pain
t, both acrylic and oil, to bring their ideas to life.  And, having a vari
ety of canvas sizes help the students to make choices just like the artist
s they are learning to behave and think like.    Having choice in the clas
sroom makes a huge difference in how my students become artists.  Being ab
le to decide if using paint and what size canvas to work on is just one st
ep into thinking like an artist and becoming an artist.  It is all part or
the critical thinking process, which is a skill  they can transfer to othe
r aspects of their lives beyond the art studio.nannan
```

In [32]:

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

```
My studio classroom is living entity just like the artwork its young artis
ts create They are learning to think and behave just like artists do My st
udents come from a wide variety of backgrounds and home lives They are eag
er to learn and to create artwork that has meaning or a connection to them
They work hard and create amazing authentic artwork I as a teacher and an
artist am proud to not only display at school but to brag about to others
and on social media Students in my art classes create artworks that focus
around a big idea ideas that showcase the actions that artists do like art
ists steal where they learned about copyright and how to properly use the
work of others in their own work Students learn about the big idea then fo
llowing the design process they figure out how to use what they have learn
ed to bring their own connection into artworks They choose the subject mat
ter They choose the material or materials to work with They choose the siz
e They make the creative decisions from beginning to end This year many st
udents are very interested in using paint both acrylic and oil to bring th
eir ideas to life And having a variety of canvas sizes help the students t
o make choices just like the artists they are learning to behave and think
like Having choice in the classroom makes a huge difference in how my stud
ents become artists Being able to decide if using paint and what size canv
as to work on is just one step into thinking like an artist and becoming a
n artist It is all part or the critical thinking process which is a skill
they can transfer to other aspects of their lives beyond the art studio na
nnan
```

In [33]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
# because although they are in this list but they matter a lot because
# they change the meaning of the entire sentence.
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', 'each', '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', "doesn'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 [34]:

```
# Combining all the above preprocessing techniques for all the project essays
from tqdm import tqdm
preprocessed_essays_Test = []
# tqdm is for printing the status bar
for sentance in tqdm(X_test['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.lower() not in stopwords)
    preprocessed_essays_Test.append(sent.lower().strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████
█████| 36052/36052 [00:16<00:00, 2220.47it/s]
```

In [35]:

```
# after preprocesing of project essays
preprocessed_essays_Test[20000]
```

Out[35]:

```
'studio classroom living entity like artwork young artists create learning
think behave like artists students come wide variety backgrounds home live
s eager learn create artwork meaning connection work hard create amazing a
uthentic artwork teacher artist proud not display school brag others socia
l media students art classes create artworks focus around big idea ideas s
howcase actions artists like artists steal learned copyright properly use
work others work students learn big idea following design process figure u
se learned bring connection artworks choose subject matter choose material
materials work choose size make creative decisions beginning end year many
students interested using paint acrylic oil bring ideas life variety canva
s sizes help students make choices like artists learning behave think like
choice classroom makes huge difference students become artists able decide
using paint size canvas work one step thinking like artist becoming artist
part critical thinking process skill transfer aspects lives beyond art stu
dio nannan'
```

Preprocessing of project_title

Now we will simply apply the above preprocessing steps on the project title for the test data as well,as it is also a text feature

In [36]:

```
# printing some random titles.
print(X_test['project_title'].values[0])
print("="*50)
print(X_test['project_title'].values[150])
print("="*50)
print(X_test['project_title'].values[1000])
print("="*50)
print(X_test['project_title'].values[20000])
print("="*50)
print(X_test['project_title'].values[9999])
print("="*50)
```

```
Ms. Thode's Little Bookworms
==================================================
ART Still Matters!
==================================================
A Piano Is The Key To Choral Success
==================================================
Exploration Comes in Many Sizes
==================================================
Namaste! Yoga and Mindfulness
==================================================
```

We have already written the preprocessing codes for different preprocessing approaches now we will simply use those codes on the project titles

In [37]:

```
preprocessed_project_titles_Test = []

for t in tqdm(X_test["project_title"]):
    title = decontracted(t)
    title = title.replace('\\r', ' ')
    title = title.replace('\\"', ' ')
    title = title.replace('\\n', ' ')
    title = re.sub('[^A-Za-z0-9]+', ' ', title)
    title = ' '.join(f for f in title.split() if f not in stopwords)
    preprocessed_project_titles_Test.append(title.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████████
████| 36052/36052 [00:00<00:00, 46572.55it/s]
```

In [38]:

```
# printing some random titles of train dataset after preprocessing

print(preprocessed_project_titles_Test[5000])
print("="*50)
print(preprocessed_project_titles_Test[7000])
print("="*50)
print(preprocessed_project_titles_Test[10000])
print("="*50)
print(preprocessed_project_titles_Test[4500])
print("="*50)
print(preprocessed_project_titles_Test[22000])
print("="*50)
```

```
let problem solve through stem
==================================================
technically speaking we need technology
==================================================
chromebook green team
==================================================
help us become better readers
==================================================
pitiful classroom library needs modern update
==================================================
```

Cross validation data

Preprocessing of project_subject_categories

In [39]:

```
catogories = list(X_cv['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
ng
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth
on
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", "Warmt
h", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "M
ath & Science"=> "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:"M
ath & Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spa
ces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

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

from collections import Counter
my_counter = Counter()
for word in X_test['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]))
```

Preprocessing of project_subject_subcategories

In [40]:

```
sub_catogories = list(X_cv['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

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

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", "Warmt
h", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "M
ath & Science"=> "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:"M
ath & Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the trailing spa
ces
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())

X_cv['clean_subcategories'] = sub_cat_list
X_cv.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 X_cv['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]))
```

Text Preprocessing

In [41]:

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

In [42]:

```
X_cv.head(5)
```

Out[42]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state
95329	127860	p069640	7d95cacdebaac7306e28ef213fbfc76a	Ms.	NC
35491	85212	p030929	841b362ce40d2da730a4e1e8a0a046af	Mrs.	MI
86193	135072	p163944	b550a84b86ea73d94167bc182ffeccef	Ms.	OK
73726	45473	p219659	08f0f75129924ddd4a803db0a0b0c36d	Mrs.	NC
28146	21001	p247403	b00932680e005d623f9a1af97e17b3f6	Ms.	IL

In [43]:

```
# printing some random reviews
print(X_cv['essay'].values[0])
print("="*50)
print(X_cv['essay'].values[150])
print("="*50)
print(X_cv['essay'].values[1000])
print("="*50)
print(X_cv['essay'].values[20000])
print("="*50)
print(X_cv['essay'].values[9999])
print("="*50)
```

My first grade kids are always very excited to learn many things about cul
tures and life subjects along with reading, writing, and math, using Spani
sh as a second language. I am so proud of them; they really work so hard
the whole year communicating in a foreign language. Although some of them
struggle with learning a new language, they never give up. \r\nMy students
have different backgrounds that make them very respectful, tolerant and co
operative to the diversity of others.To learn sometimes is not easy for ma
ny people. Additionally, to learn a second language could be harder, and t
his is the challenge that my students have to face everyday when they are
in my classroom. They love to discover how magic other cultures can be, bu
t also how difficult it can be to express their ideas in another language.
I have been noticing that sometimes the kids get frustrated with some acti
vities, such as learning to write and describe things, but if they can rel
ease the stress by moving their body while they are working, is easier for
them. \r\n\r\n\"Why can't we have some of the bouncy bands that are in Mrs
Adams' classroom?\" asked one of my kids when he was back from interventio
ns. \"I like to move my foot when I'm writing.\" This question inspired a
nd encouraged me to apply for this project. I want to improve my classroom
in order that my kids feel more comfortable while they are learning.nannan
==========
My class consists of lively Kindergartners who are fascinated with learni
ng! These little sponges soak up all they can in a day! The children know
that reading can take them anywhere and they love to read. I teach in a lo
w income/ high poverty school district and my school isn't immune to that
statistic. Not only are these sweet kids facing poverty struggles but many
are also English language learners. I know that poverty and language does
n't define who they are or what they can do. My job is to help lay a found
ation on which to build a lifelong love of learning and help each child so
ar into a better future.During guided reading I will \"guide\" students th
rough managing, reading, and writing. These books will help my students ga
in confidence while learning how to read books that they enjoy, good quali
ty text that will help build strategic activity.\r\n\r\nIn the words of Ma
rie Clay \"Reading is a message getting, problem solving activity\". In or
der for my students to develop strategic activity and fast processing I ne
ed to expose them to high quality text. The Rigby series of books offers q
uality text that has great kid friendly, relatable story lines. These book
s will hold their interest while helping them learn how to problem solve i
n order to be a great reader.\r\nnannan
==========
My students come from diverse family situations that often interfere with
student learning. My school has a great sense of community, and strives t
o provide quality services to individuals who may or may not be concerned
about their education. Often times, daily obstacles are at the forefront
of their minds. \r\n\r\nMy scholars are phenomenal, young individuals with
incredibly caring hearts. I want each of them to work productively during
their time at school and obtain a great love of learning. \r\nColored pap
er, pencils, and markers will add great excitement to my students' learni
ng experiences.   I was amazed when I gave my students their very own no
tebooks prior to holiday break. Their excitement was shocking. For many
of them, that was their first notebook to call their own. Many of my stud
ents like to create their own books. \r\n\r\nProviding the most basic sch
ool supplies will ignite their imagination and creativity. In turn, their
excitement for school will grow and their academic success will come to fr
uition. Your direct participation in providing my students with these bas
ic supplies will truly inspire them to become strong adults of the future.
nannan
==========
My students need a hydrogarden planter kit, herb seed, greenhouse, cold we
ather thermo cube outlet, a growing tent so we can eat health all year lon
g! My students come from hard working, economically deprived families. Th
is summer I have been able to continue working with them in our community

garden! I have been able to discuss nutrition and the importance of health y with my students and their families! Many of these families depend on th e community gardens for healthy eating over the summer.\r\nMy students nee d a hydrogarden planter kit, herb seed, greenhouse, cold weather thermo cu be outlet, a growing tent so we can eat health all year long! They have a sked how is it possible to eat healthy all year long! I want to continue t eaching them about the benefits of healthy eating and have them bring it b ack to their families and community!\r\nI want my students to bring awaren ess to the community about the importance of eating healthy! They will be able to eat healthy all year long with fruits and vegetables they grow and try!This summer I worked with my students and they expressed an interest i n winter gardening and health eating. They wanted to figure out a way to get fresh veggies all year long. They came to me about indoor gardening a nd I asked them to research what they think they needed. They know they c an not grow enough to feed the entires school but they want to bring aware ness that gardens can grow evening an Urban setting and even during the wi nter! Students will be growing and harvesting and eating the vegetables an d fruits they grow. They want to try different vegetables and learn how t o cook them. They also asked if they could make a cookbook and share it w ith the school about the experience of eating healthy food during winter. Many teens feel healthy eating, fresh and organic food is not easily acces sible for everyone they want to show their peers it is! Many of my student s have said It is often cheaper and easier for them to go to a drive-thru then to have a healthy snack! They want to provide healthy snacks to their classmates!\r\nThe love of gardening is a seed once sown never dies.\r\nG. Jekyll\r\nThe students will be eating more nutrient dense foods as that is what they want to grow. They want to be healthy and they believe this is an excellent way not only to eat healthy but to eat healthy foe a lifetim e!nannan

==========

Despite living in a low-income/high poverty area, my kiddos still deserve the best technology and resources the world has to offer! \r\nI want my st udents to know that they have what they need, when they need it, to help t hem learn and grow!\r\nMy students are the future of the world, and I want to impact their future by getting them on the appropriate grade level. I c an't express to you in a limited number of characters how wonderful and de serving each one of my babies are.These nice, tech-savvy Chromebooks will allow my students, who typically do not have the opportunity, to hold tech nology in their hands on a daily basis. We will be using them to publish o ur writing pieces as well as work to improve our comprehension levels. \r \nMy students deserve to have the same, exact opportunities as children in other neighborhoods. \r\nStudents will be using programs such as Achieve30 00 and i-Ready to help them be the best 5th graders that they can be! At t his time, we have limited access to computers and this proves as a true ed ucational disadvantage to these students.nannan

==========

In [44]:

```
# creating a function named as decontracted which does the job of decontraction

# 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 [45]:

```
sent = decontracted(X_cv['essay'].values[20000])
print(sent)
print("="*50)
```

```
My students need a hydrogarden planter kit, herb seed, greenhouse, cold we
ather thermo cube outlet, a growing tent so we can eat health all year lon
g!  My students come from hard working, economically deprived families. Th
is summer I have been able to continue working with them in our community
garden! I have been able to discuss nutrition and the importance of health
y with my students and their families! Many of these families depend on th
e community gardens for healthy eating over the summer.\r\nMy students nee
d a hydrogarden planter kit, herb seed, greenhouse, cold weather thermo cu
be outlet, a growing tent so we can eat health all year long!  They have a
sked how is it possible to eat healthy all year long! I want to continue t
eaching them about the benefits of healthy eating and have them bring it b
ack to their families and community!\r\nI want my students to bring awaren
ess to the community about the importance of eating healthy! They will be
able to eat healthy all year long with fruits and vegetables they grow and
try!This summer I worked with my students and they expressed an interest i
n winter gardening and health eating.  They wanted to figure out a way to
get fresh veggies all year long.  They came to me about indoor gardening a
nd I asked them to research what they think they needed.  They know they c
an not grow enough to feed the entires school but they want to bring aware
ness that gardens can grow evening an Urban setting and even during the wi
nter! Students will be growing and harvesting and eating the vegetables an
d fruits they grow.  They want to try different vegetables and learn how t
o cook them.  They also asked if they could make a cookbook and share it w
ith the school about the experience of eating healthy food during winter.
Many teens feel healthy eating, fresh and organic food is not easily acces
sible for everyone they want to show their peers it is! Many of my student
s have said It is often cheaper and easier for them to go to a drive-thru
then to have a healthy snack! They want to provide healthy snacks to their
classmates!\r\nThe love of gardening is a seed once sown never dies.\r\nG.
Jekyll\r\nThe students will be eating more nutrient dense foods as that is
what they want to grow.  They want to be healthy and they believe this is
an excellent way not only to eat healthy but to eat healthy foe a lifetim
e!nannan
==================================================
```

In [46]:

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

```
My students need a hydrogarden planter kit, herb seed, greenhouse, cold we
ather thermo cube outlet, a growing tent so we can eat health all year lon
g!  My students come from hard working, economically deprived families. Th
is summer I have been able to continue working with them in our community
garden! I have been able to discuss nutrition and the importance of health
y with my students and their families! Many of these families depend on th
e community gardens for healthy eating over the summer.  My students need
a hydrogarden planter kit, herb seed, greenhouse, cold weather thermo cube
outlet, a growing tent so we can eat health all year long!  They have aske
d how is it possible to eat healthy all year long! I want to continue teac
hing them about the benefits of healthy eating and have them bring it back
to their families and community!  I want my students to bring awareness to
the community about the importance of eating healthy! They will be able to
eat healthy all year long with fruits and vegetables they grow and try!Thi
s summer I worked with my students and they expressed an interest in winte
r gardening and health eating.  They wanted to figure out a way to get fre
sh veggies all year long.  They came to me about indoor gardening and I as
ked them to research what they think they needed.  They know they can not
grow enough to feed the entires school but they want to bring awareness th
at gardens can grow evening an Urban setting and even during the winter! S
tudents will be growing and harvesting and eating the vegetables and fruit
s they grow.  They want to try different vegetables and learn how to cook
them.  They also asked if they could make a cookbook and share it with the
school about the experience of eating healthy food during winter. Many tee
ns feel healthy eating, fresh and organic food is not easily accessible fo
r everyone they want to show their peers it is! Many of my students have s
aid It is often cheaper and easier for them to go to a drive-thru then to
have a healthy snack! They want to provide healthy snacks to their classma
tes!  The love of gardening is a seed once sown never dies.  G.Jekyll  The
students will be eating more nutrient dense foods as that is what they wan
t to grow.  They want to be healthy and they believe this is an excellent
way not only to eat healthy but to eat healthy foe a lifetime!nannan
```

In [47]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
# because although they are in this list but they matter a lot because
# they change the meaning of the entire sentence.
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', 'each', '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', "doesn'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 [48]:

```
# Combining all the above preprocessing techniques for all the project essays
from tqdm import tqdm
preprocessed_essays_Cv = []
# tqdm is for printing the status bar
for sentance in tqdm(X_cv['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.lower() not in stopwords)
    preprocessed_essays_Cv.append(sent.lower().strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████
█████| 24155/24155 [00:11<00:00, 2063.40it/s]
```

In [49]:

```
# after preprocesing of project essays
preprocessed_essays_Test[20000]
```

Out[49]:

```
'studio classroom living entity like artwork young artists create learning
think behave like artists students come wide variety backgrounds home live
s eager learn create artwork meaning connection work hard create amazing a
uthentic artwork teacher artist proud not display school brag others socia
l media students art classes create artworks focus around big idea ideas s
howcase actions artists like artists steal learned copyright properly use
work others work students learn big idea following design process figure u
se learned bring connection artworks choose subject matter choose material
materials work choose size make creative decisions beginning end year many
students interested using paint acrylic oil bring ideas life variety canva
s sizes help students make choices like artists learning behave think like
choice classroom makes huge difference students become artists able decide
using paint size canvas work one step thinking like artist becoming artist
part critical thinking process skill transfer aspects lives beyond art stu
dio nannan'
```

Preprocessing of project_title

Now we will simply apply the above preprocessing steps on the project title for the Cross Validation data as well,as it is also a text feature

In [50]:

```
# printing some random titles.
print(X_cv['project_title'].values[0])
print("="*50)
print(X_cv['project_title'].values[150])
print("="*50)
print(X_cv['project_title'].values[1000])
print("="*50)
print(X_cv['project_title'].values[20000])
print("="*50)
print(X_cv['project_title'].values[9999])
print("="*50)
```

```
Moving While We are Learning
==================================================
Read to be Ready!
==================================================
Back to Basics
==================================================
Urban Winter Gardening!
==================================================
Laptops for Learning
==================================================
```

In [51]:

```
preprocessed_project_titles_Cv = []

for t in tqdm(X_cv["project_title"]):
    title = decontracted(t)
    title = title.replace('\\r', ' ')
    title = title.replace('\\"', ' ')
    title = title.replace('\\n', ' ')
    title = re.sub('[^A-Za-z0-9]+', ' ', title)
    title = ' '.join(f for f in title.split() if f not in stopwords)
    preprocessed_project_titles_Cv.append(title.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████████
████| 24155/24155 [00:00<00:00, 46866.19it/s]
```

In [52]:

```
# printing some random titles of crossvalidation dataset after preprocessing

print(preprocessed_project_titles_Cv[5000])
print("="*50)
print(preprocessed_project_titles_Cv[7000])
print("="*50)
print(preprocessed_project_titles_Cv[10000])
print("="*50)
print(preprocessed_project_titles_Cv[4500])
print("="*50)
print(preprocessed_project_titles_Cv[22000])
print("="*50)
```

```
cheerleading team needs mat
==================================================
ready set graduate x 2
==================================================
fraction frenzy
==================================================
dependable technology dedicated students endless possibilities
==================================================
a new perspective math
==================================================
```

Preparing Data For Models

In [53]:

```
project_data.columns
```

Out[53]:

```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'Date', '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_approve
d'],
      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 (optinal)
- quantity : numerical (optinal)
- teacher_number_of_previously_posted_projects : numerical
- price : numerical

Now firstly we will be vectorizing the categorical data

For vectorizing the categorical data we will be using One Hot Encoding Technique

One Hot Encoding Of Project Clean Categories

In [638]:

```
# we use count vectorizer to convert the values into one hot encoded features


from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()), lowercase=False,
binary=True)

# we will be using the X_train for fitting our model because that is the only data a us
er knows rest all are for testing purposes
vectorizer.fit(X_train['clean_categories'].values)

print(vectorizer.get_feature_names())

categories_one_hot_train = vectorizer.transform(X_train['clean_categories'].values)

categories_one_hot_test = vectorizer.transform(X_test['clean_categories'].values)

categories_one_hot_cv = vectorizer.transform(X_cv['clean_categories'].values)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearnin
g', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language']
```

In [639]:

```
print("Shape of  Train data matrix after one hot encoding ",categories_one_hot_train.sh
ape)
print("Shape of  Test data matrix after one hot encoding ",categories_one_hot_test.shap
e)
print("Shape of  CV data matrix after one hot encoding ",categories_one_hot_cv.shape)
```

```
Shape of  Train data matrix after one hot encoding  (49041, 9)
Shape of  Test data matrix after one hot encoding  (36052, 9)
Shape of  CV data matrix after one hot encoding  (24155, 9)
```

Storing the feature names received from above in a list

In [640]:

```
all_feature_names = []
```

In [641]:

```
for i in vectorizer.get_feature_names():
    all_feature_names.append(i)
print(all_feature_names)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearnin
g', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language']
```

In [642]:

```
len(all_feature_names)
```

Out[642]:

```
9
```

One Hot Encoding Of Cleaned Project Sub Category

In [643]:

```
# we use count vectorizer to convert the values into one hot encoded features


from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys()), lowercase=Fal
se, binary=True)

# we will be using the X_train for fitting our model because that is the only data a us
er knows rest all are for testing purposes
vectorizer.fit(X_train['clean_subcategories'].values)

print(vectorizer.get_feature_names())

subcategories_one_hot_train = vectorizer.transform(X_train['clean_subcategories'].value
s)

subcategories_one_hot_test = vectorizer.transform(X_test['clean_subcategories'].values)

subcategories_one_hot_cv = vectorizer.transform(X_cv['clean_subcategories'].values)
```

```
['Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvemen
t', 'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'Nutrition
Education', 'Warmth', 'Care_Hunger', 'PerformingArts', 'SocialSciences',
'CharacterEducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Musi
c', 'History_Geography', 'ESL', 'EarlyDevelopment', 'Health_LifeScience',
'Gym_Fitness', 'EnvironmentalScience', 'VisualArts', 'Health_Wellness', 'A
ppliedSciences', 'SpecialNeeds', 'Literature_Writing', 'Mathematics', 'Lit
eracy']
```

In [644]:

```
print("Shape of  Train data matrix after one hot encoding ",subcategories_one_hot_train
.shape)
print("Shape of  Test data matrix after one hot encoding ",subcategories_one_hot_test.s
hape)
print("Shape of  CV data matrix after one hot encoding ",subcategories_one_hot_cv.shape
)
```

```
Shape of  Train data matrix after one hot encoding  (49041, 30)
Shape of  Test data matrix after one hot encoding  (36052, 30)
Shape of  CV data matrix after one hot encoding  (24155, 30)
```

In [645]:

```
for i in vectorizer.get_feature_names():
    all_feature_names.append(i)
print(all_feature_names)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearnin
g', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language',
'Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement',
'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'NutritionEduc
ation', 'Warmth', 'Care_Hunger', 'PerformingArts', 'SocialSciences', 'Char
acterEducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Music', 'Hi
story_Geography', 'ESL', 'EarlyDevelopment', 'Health_LifeScience', 'Gym_Fi
tness', 'EnvironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedS
ciences', 'SpecialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy']
```

In [646]:

```
len(all_feature_names)
```

Out[646]:

```
39
```

One hot encoding of teacher prefix

In [647]:

```
mylist_teacher_prefix = list(X_train['teacher_prefix'])
```

In [648]:

```
# We are removing the duplicate values from our list of the teacher prefix

# Source :- https://www.w3schools.com/python/python_howto_remove_duplicates.asp

mylist_teacher_prefix_actual_Train = list(dict.fromkeys(mylist_teacher_prefix))
```

In [649]:

```
# removing the nan from the teacher prefix category as there is no such category of tea
cher which exists

mylist_teacher_prefix_actual_Train = [p for p in mylist_teacher_prefix_actual_Train if
str(p) != 'nan']
mylist_teacher_prefix_actual_Train
```

Out[649]:

```
['Mr.', 'Ms.', 'Mrs.', 'Teacher', 'Dr.']
```

In [650]:

```
# we use count vectorizer to convert the values into one hot encoded features

# now we are working on teacher prefix data

from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary=mylist_teacher_prefix_actual_Train, lowercase=False, binary=True)

# I was getting an error like "np.nan is an invalid document, expected byte or unicode
 string."
# below is the solution

# https://stackoverflow.com/questions/39303912/tfidfvectorizer-in-scikit-learn-valueerror-np-nan-is-an-invalid-document


vectorizer.fit(X_train['teacher_prefix'].values.astype('U'))
print(vectorizer.get_feature_names())

teacher_prefix_one_hot_train = vectorizer.transform(X_train['teacher_prefix'].values.astype('U'))
teacher_prefix_one_hot_test = vectorizer.transform(X_test['teacher_prefix'].values.astype('U'))
teacher_prefix_one_hot_cv = vectorizer.transform(X_cv['teacher_prefix'].values.astype('U'))
```

```
['Mr.', 'Ms.', 'Mrs.', 'Teacher', 'Dr.']
```

In [651]:

```
print("Shape of matrix of Train data after one hot encoding ",teacher_prefix_one_hot_train.shape)
print("Shape of matrix of Test data after one hot encoding ",teacher_prefix_one_hot_test.shape)
print("Shape of matrix of Cross Validation data after one hot encoding ",teacher_prefix_one_hot_cv.shape)
```

```
Shape of matrix of Train data after one hot encoding  (49041, 5)
Shape of matrix of Test data after one hot encoding  (36052, 5)
Shape of matrix of Cross Validation data after one hot encoding  (24155,
5)
```

In [652]:

```
for i in vectorizer.get_feature_names():
    all_feature_names.append(i)
print(all_feature_names)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearnin
g', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language',
'Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement',
'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'NutritionEduc
ation', 'Warmth', 'Care_Hunger', 'PerformingArts', 'SocialSciences', 'Char
acterEducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Music', 'Hi
story_Geography', 'ESL', 'EarlyDevelopment', 'Health_LifeScience', 'Gym_Fi
tness', 'EnvironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedS
ciences', 'SpecialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy',
'Mr.', 'Ms.', 'Mrs.', 'Teacher', 'Dr.']
```

In [653]:

```
len(all_feature_names)
```

Out[653]:

44

One Hot encoding of project grade category

In [654]:

```
mylist_project_grade_category = list(X_train['project_grade_category'])
```

In [655]:

```
# We are removing the duplicate values from our list of the project grade category

# Source :- https://www.w3schools.com/python/python_howto_remove_duplicates.asp

mylist_project_grade_category_actual = list(dict.fromkeys(mylist_project_grade_category
))
```

In [656]:

```
type(mylist_project_grade_category_actual)
print(mylist_project_grade_category_actual[0])

n = len(mylist_project_grade_category_actual)
print(n)

# I already saw by running the code that the word Grades is unnecessarily present in th
e elements of list hence trying to remove that word
# how to remove a word from a sentence --> https://codescracker.com/python/program/pyth
on-program-remove-word-from-sentence.htm
for m in range(0,4,1):
    words = mylist_project_grade_category_actual[m].split()
    mylist_project_grade_category_actual[m] = ''.join([j for j in words if j not in "Gr
ades"])
print(mylist_project_grade_category_actual)
```

```
Grades 9-12
4
['9-12', 'PreK-2', '6-8', '3-5']
```

In [657]:

```
# we use count vectorizer to convert the values into one hot encoded features

# now we are working on project grade category data

from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary=mylist_project_grade_category_actual, lowercase
=False, binary=True)


vectorizer.fit(X_train['project_grade_category'].values)
print(vectorizer.get_feature_names())

project_grade_categories_one_hot_train = vectorizer.transform(X_train['project_grade_ca
tegory'].values)
project_grade_categories_one_hot_test = vectorizer.transform(X_test['project_grade_cate
gory'].values)
project_grade_categories_one_hot_cv = vectorizer.transform(X_cv['project_grade_categor
y'].values)
```

```
['9-12', 'PreK-2', '6-8', '3-5']
```

In [658]:

```
print("Shape of matrix of Train data after one hot encoding ",project_grade_categories_
one_hot_train.shape)
print("Shape of matrix of Test data after one hot encoding ",project_grade_categories_o
ne_hot_test.shape)
print("Shape of matrix of Cross Validation data after one hot encoding ",project_grade_
categories_one_hot_cv.shape)
```

```
Shape of matrix of Train data after one hot encoding  (49041, 4)
Shape of matrix of Test data after one hot encoding  (36052, 4)
Shape of matrix of Cross Validation data after one hot encoding  (24155,
4)
```

In [659]:

```
for i in vectorizer.get_feature_names():
    all_feature_names.append(i)
print(all_feature_names)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearnin
g', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language',
'Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement',
'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'NutritionEduc
ation', 'Warmth', 'Care_Hunger', 'PerformingArts', 'SocialSciences', 'Char
acterEducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Music', 'Hi
story_Geography', 'ESL', 'EarlyDevelopment', 'Health_LifeScience', 'Gym_Fi
tness', 'EnvironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedS
ciences', 'SpecialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy',
'Mr.', 'Ms.', 'Mrs.', 'Teacher', 'Dr.', '9-12', 'PreK-2', '6-8', '3-5']
```

In [660]:

```
len(all_feature_names)
```

Out[660]:

48

One hot encoding of School States

In [661]:

```
type(list(project_data['school_state']))
```

Out[661]:

list

In [662]:

```
mylist = list(X_train['school_state'])
```

In [663]:

```
# We are removing the duplicate values from our list of the state codes

# Source :- https://www.w3schools.com/python/python_howto_remove_duplicates.asp

mylist_actual = list(dict.fromkeys(mylist))
```

In [664]:

```
# we use count vectorizer to convert the values into one hot encoded features

# now we are working on school state data

from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(vocabulary=mylist_actual, lowercase=False, binary=True)

vectorizer.fit(X_train['school_state'].values)

print(vectorizer.get_feature_names())

school_state_categories_one_hot_train = vectorizer.transform(X_train['school_state'].va
lues)
school_state_categories_one_hot_test = vectorizer.transform(X_test['school_state'].valu
es)
school_state_categories_one_hot_cv = vectorizer.transform(X_cv['school_state'].values)
```

```
['MI', 'MT', 'MO', 'KS', 'CA', 'PA', 'UT', 'WA', 'ID', 'IL', 'SC', 'NY',
'GA', 'TX', 'AZ', 'CO', 'WI', 'NH', 'TN', 'NC', 'OR', 'AL', 'LA', 'VA', 'O
K', 'MD', 'FL', 'NM', 'IN', 'OH', 'AR', 'KY', 'NJ', 'CT', 'MA', 'AK', 'M
S', 'DE', 'NV', 'WV', 'DC', 'NE', 'HI', 'MN', 'IA', 'ME', 'RI', 'VT', 'N
D', 'SD', 'WY']
```

In [665]:

```
print("Shape of matrix of Train data after one hot encoding ",school_state_categories_o
ne_hot_train.shape)
print("Shape of matrix of Test data after one hot encoding ",school_state_categories_on
e_hot_test.shape)
print("Shape of matrix of Cross Validation data after one hot encoding ",school_state_c
ategories_one_hot_cv.shape)
```

```
Shape of matrix of Train data after one hot encoding  (49041, 51)
Shape of matrix of Test data after one hot encoding  (36052, 51)
Shape of matrix of Cross Validation data after one hot encoding  (24155, 5
1)
```

In [666]:

```
for i in vectorizer.get_feature_names():
    all_feature_names.append(i)
print(all_feature_names)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearnin
g', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language',
'Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement',
'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'NutritionEduc
ation', 'Warmth', 'Care_Hunger', 'PerformingArts', 'SocialSciences', 'Char
acterEducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Music', 'Hi
story_Geography', 'ESL', 'EarlyDevelopment', 'Health_LifeScience', 'Gym_Fi
tness', 'EnvironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedS
ciences', 'SpecialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy',
'Mr.', 'Ms.', 'Mrs.', 'Teacher', 'Dr.', '9-12', 'PreK-2', '6-8', '3-5', 'M
I', 'MT', 'MO', 'KS', 'CA', 'PA', 'UT', 'WA', 'ID', 'IL', 'SC', 'NY', 'G
A', 'TX', 'AZ', 'CO', 'WI', 'NH', 'TN', 'NC', 'OR', 'AL', 'LA', 'VA', 'O
K', 'MD', 'FL', 'NM', 'IN', 'OH', 'AR', 'KY', 'NJ', 'CT', 'MA', 'AK', 'M
S', 'DE', 'NV', 'WV', 'DC', 'NE', 'HI', 'MN', 'IA', 'ME', 'RI', 'VT', 'N
D', 'SD', 'WY']
```

In [667]:

```
len(all_feature_names)
```

Out[667]:

99

Now as we have done the vectorizing of categorical data now we will be vectorizing the text data using different techniques

Vectorizing the text data

Technique -1 Bag of words(BOW)

Essays

Essays Train Data

In [668]:

```
# I am not setting the value of min_df here because i read here
# https://stackoverflow.com/questions/27697766/understanding-min-df-and-max-df-in-scikit-countvectorizer
# that min_df helps in better performance but might also give poor clusters hence am trying without it this time



vectorizer = CountVectorizer(ngram_range=(1, 3),max_features = 50000)
vectorizer.fit(preprocessed_essays_Train)

essay_bow_train = vectorizer.transform(preprocessed_essays_Train)

print("Shape of matrix after bag of words ",essay_bow_train.shape)
```

```
Shape of matrix after bag of words  (49041, 50000)
```

Essay Test Data

In [669]:

```
# I am not setting the value of min_df here because i read here
# https://stackoverflow.com/questions/27697766/understanding-min-df-and-max-df-in-scikit-countvectorizer
# that min_df helps in better performance but might also give poor clusters hence am trying without it this time


# now note that the below two lines are wrong because we have already trained the
# model on the training data now using that model we should get the bow representation
# of test data. After all training from test data only and then cecking for its accuracy will ofcourse give
# good accuracy.

# lines not to be used (I used them but then going through the code realised the mistake)

#vectorizer = CountVectorizer()
#vectorizer.fit(preprocessed_essays_Test)

essay_bow_test = vectorizer.transform(preprocessed_essays_Test)

print("Shape of matrix after bag of words ",essay_bow_test.shape)
```

```
Shape of matrix after bag of words  (36052, 50000)
```

Essay Cross Validation data

In [670]:

```
# I am not setting the value of min_df here because i read here
# https://stackoverflow.com/questions/27697766/understanding-min-df-and-max-df-in-scikit-countvectorizer
# that min_df helps in better performance but might also give poor clusters hence am trying without it this time

# similarly below two lines should not be used

#vectorizer = CountVectorizer()
#vectorizer.fit(preprocessed_essays_Cv)

essay_bow_cv = vectorizer.transform(preprocessed_essays_Cv)

print("Shape of matrix after bag of words ",essay_bow_cv.shape)
```

```
Shape of matrix after bag of words  (24155, 50000)
```

In [671]:

```
# trying a dummy example for the code of my next cell

a = [2,3,4,5]
b = a.copy()
b.append(6)

print(b)
print(a)
```

```
[2, 3, 4, 5, 6]
[2, 3, 4, 5]
```

In [672]:

```
# appending the feature names from the BOW technique also

all_feature_names_bow = all_feature_names.copy()

for i in vectorizer.get_feature_names():
    all_feature_names_bow.append(i)
#print(all_feature_names)
```

In [673]:

```
len(all_feature_names_bow)
```

Out[673]:

```
50099
```

Project Title

Bag of words on Project Title Train data

In [674]:

```
vectorizer.fit(preprocessed_project_titles_Train)
project_title_bow_train = vectorizer.transform(preprocessed_project_titles_Train)
print("Shape of matrix after bag of words ",project_title_bow_train.shape)
```

```
Shape of matrix after bag of words  (49041, 50000)
```

Bag of words on Project Title Test data

In [675]:

```
project_title_bow_test = vectorizer.transform(preprocessed_project_titles_Test)
print("Shape of matrix after bag of words ",project_title_bow_test.shape)
```

```
Shape of matrix after bag of words  (36052, 50000)
```

Bag of words on Project Title Cross Validation data

In [676]:

```
project_title_bow_cv = vectorizer.transform(preprocessed_project_titles_Cv)
print("Shape of matrix after bag of words ",project_title_bow_cv.shape)
```

```
Shape of matrix after bag of words  (24155, 50000)
```

In [677]:

```
# appending the feature names from the BOW technique also

for i in vectorizer.get_feature_names():
    all_feature_names_bow.append(i)
#print(all_feature_names)
```

In [678]:

```
len(all_feature_names_bow)
```

Out[678]:

```
100099
```

Technique-2 TFIDF

Essay Data

Essay Train Data

In [679]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(ngram_range=(1, 3),max_features = 6000, min_df = 10)
vectorizer.fit(preprocessed_essays_Train)

text_tfidf_train = vectorizer.transform(preprocessed_essays_Train)
print("Shape of matrix after tfidf ",text_tfidf_train.shape)

print(type(text_tfidf_train))

# we are converting a dictionary with word as a key, and the idf as a value

dictionary = dict(zip(vectorizer.get_feature_names(), list(vectorizer.idf_)))
tfidf_words = set(dictionary.keys())
```

```
Shape of matrix after tfidf  (49041, 6000)
<class 'scipy.sparse.csr.csr_matrix'>
```

Essay Test data

In [680]:

```
text_tfidf_test = vectorizer.transform(preprocessed_essays_Test)
print("Shape of matrix after tfidf ",text_tfidf_test.shape)
```

```
Shape of matrix after tfidf  (36052, 6000)
```

Essay Cross Validation Data

In [681]:

```
text_tfidf_cv = vectorizer.transform(preprocessed_essays_Cv)
print("Shape of matrix after tfidf ",text_tfidf_cv.shape)
```

```
Shape of matrix after tfidf  (24155, 6000)
```

In [682]:

```
all_feature_names_tfidf = all_feature_names.copy()
```

In [683]:

```
# appending the feature names from the tf-idf technique also

for i in vectorizer.get_feature_names():
    all_feature_names_tfidf.append(i)

#print(all_feature_names)
```

In [684]:

```
len(all_feature_names_tfidf)
```

Out[684]:

6099

Project Title

Project title train data

In [685]:

```
vectorizer = TfidfVectorizer(ngram_range=(1,3),max_features = 2000, min_df = 10)

vectorizer.fit(preprocessed_project_titles_Train)

project_title_tfidf_train = vectorizer.transform(preprocessed_project_titles_Train)

print("Shape of matrix after tfidf ",project_title_tfidf_train.shape)

# we are converting a dictionary with word as a key, and the idf as a value

dictionary = dict(zip(vectorizer.get_feature_names(), list(vectorizer.idf_)))
tfidf_project_title_words = set(dictionary.keys())
```

```
Shape of matrix after tfidf  (49041, 2000)
```

Project title test data

In [686]:

```
project_title_tfidf_test = vectorizer.transform(preprocessed_project_titles_Test)
print("Shape of matrix after tfidf ",project_title_tfidf_test.shape)
```

```
Shape of matrix after tfidf  (36052, 2000)
```

Project title cross validation data

In [687]:

```
title_tfidf_cv = vectorizer.transform(preprocessed_project_titles_Cv)
print("Shape of matrix after tfidf ",title_tfidf_cv.shape)
```

```
Shape of matrix after tfidf  (24155, 2000)
```

In [688]:

```
# appending the feature names from the tf-idf technique also

for i in vectorizer.get_feature_names():
    all_feature_names_tfidf.append(i)
#print(all_feature_names)
```

In [689]:

```
len(all_feature_names_tfidf)
```

Out[689]:

8099

Numerical Features

Vectorizing numerical features

Price for projects

In [690]:

```
# now firstly we will try to add the price and the quantity of the items required from
 the resource dataframe

price_quantity_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'
}).reset_index()

price_quantity_data.head(2)
```

Out[690]:

	id	price	quantity
0	p000001	459.56	7
1	p000002	515.89	21

now we need to join the above dataframe with our train,test,cv data that we already have

In [691]:

```
X_train = pd.merge(X_train, price_quantity_data, on='id', how='left')
X_test = pd.merge(X_test, price_quantity_data, on='id', how='left')
X_cv = pd.merge(X_cv, price_quantity_data, on='id', how='left')
```

we will be performing the normalization of the numerical data here

Normalizing the price data

In [692]:

```
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))

price_train = normalizer.transform(X_train['price'].values.reshape(-1,1))
price_cv = normalizer.transform(X_cv['price'].values.reshape(-1,1))
price_test = normalizer.transform(X_test['price'].values.reshape(-1,1))

print("After vectorizations")
print(price_train.shape, y_train.shape)
print(price_cv.shape, y_cv.shape)
print(price_test.shape, y_test.shape)
```

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

Normalizing the quantity data

In [693]:

```
from sklearn.preprocessing import Normalizer

normalizer = Normalizer()

# normalizer.fit(X_train['quantity'].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))

quantity_train = normalizer.transform(X_train['quantity'].values.reshape(-1,1))
quantity_test = normalizer.transform(X_test['quantity'].values.reshape(-1,1))
quantity_cv = normalizer.transform(X_cv['quantity'].values.reshape(-1,1))

print("After vectorizations")
print(quantity_train.shape, y_train.shape)
print(quantity_test.shape, y_test.shape)
print(quantity_cv.shape, y_cv.shape)
```

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

Normalizing the number of previously posted projects by a teacher

In [694]:

```
normalizer = Normalizer()

# normalizer.fit(X_train['teacher_number_of_previously_posted_projects'].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))

prev_projects_train = normalizer.transform(X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))
prev_projects_cv = normalizer.transform(X_cv['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))
prev_projects_test = normalizer.transform(X_test['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))

print("After converting into vectors form")
print(prev_projects_train.shape, y_train.shape)
print(prev_projects_cv.shape, y_cv.shape)
print(prev_projects_test.shape, y_test.shape)
```

```
After converting into vectors form
(49041, 1) (49041,)
(24155, 1) (24155,)
(36052, 1) (36052,)
```

In [695]:

```
all_feature_names_bow.extend(["price","quantity","teacher_number_of_previously_posted_projects"])
```

In [696]:

```
all_feature_names_tfidf.extend(["price","quantity","teacher_number_of_previously_posted_projects"])
```

In [697]:

```
len(all_feature_names_bow)
```

Out[697]:

```
100102
```

In [698]:

```
len(all_feature_names_tfidf)
```

Out[698]:

```
8102
```

In [ ]:

```
```

Set 1: categorical, numerical features + project_title(BOW) + preprocessed_essay (BOW)

Now we need to merge all the numerical vectors(categorical features,text features,numerical features) given above for set-1 which we created using different methods

In [699]:

```
from scipy.sparse import hstack

X_train_merge = hstack((categories_one_hot_train,subcategories_one_hot_train,teacher_prefix_one_hot_train,project_grade_categories_one_hot_train,school_state_categories_one_hot_train,essay_bow_train,project_title_bow_train,price_train,quantity_train,prev_projects_train)).tocsr()
X_test_merge = hstack((categories_one_hot_test,subcategories_one_hot_test,teacher_prefix_one_hot_test,project_grade_categories_one_hot_test,school_state_categories_one_hot_test,essay_bow_test,project_title_bow_test,price_test,quantity_test,prev_projects_test)).tocsr()
X_cv_merge = hstack((categories_one_hot_cv,subcategories_one_hot_cv,teacher_prefix_one_hot_cv,project_grade_categories_one_hot_cv,school_state_categories_one_hot_cv,essay_bow_cv,project_title_bow_cv,price_cv,quantity_cv,prev_projects_cv)).tocsr()
```

In [700]:

```
# this will be our finally created data matrix dimensions

print(X_train_merge.shape, y_train.shape)
print(X_test_merge.shape, y_test.shape)
print(X_cv_merge.shape, y_cv.shape)
```

```
(49041, 100102) (49041,)
(36052, 100102) (36052,)
(24155, 100102) (24155,)
```

Trying random Alpha Values To get the best one

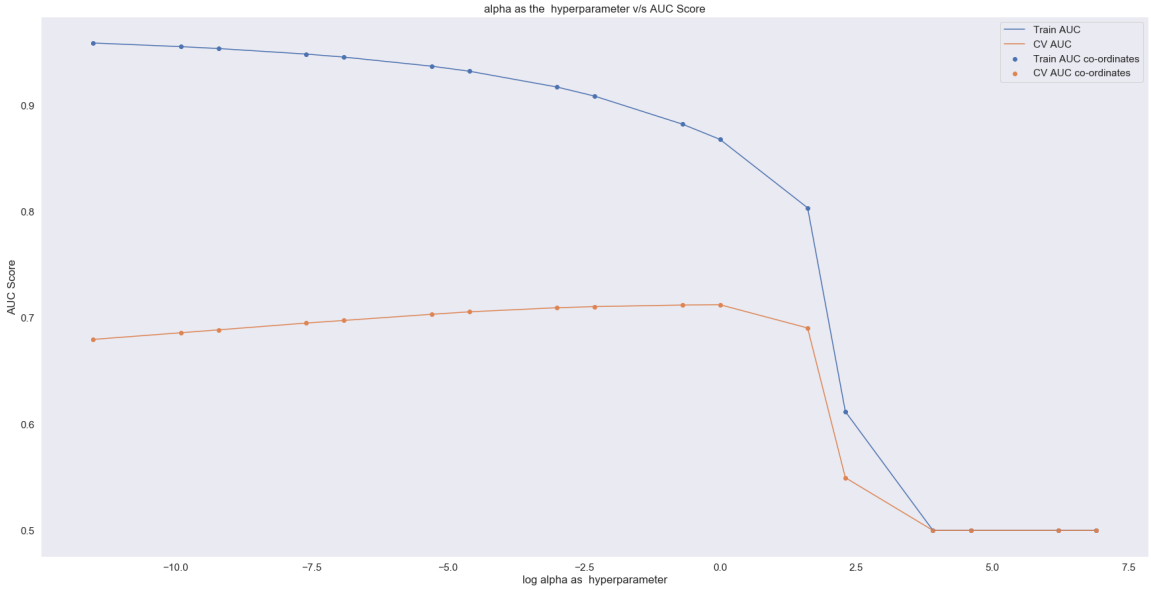
In [701]:

```
import matplotlib.pyplot as plt

from sklearn.metrics import roc_auc_score

from sklearn.naive_bayes import MultinomialNB

import math
```

In [702]:

```
train_auc = []
cv_auc = []
log_alphas = []

random_alpha_values = [0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1,
0.5, 1, 5, 10, 50, 100, 500, 1000]

for j in tqdm(random_alpha_values):

    nb = MultinomialNB(alpha = j)

    nb.fit(X_train_merge, y_train)

    y_train_pred =  nb.predict_proba(X_train_merge)[:,1]

    y_cv_pred =  nb.predict_proba(X_cv_merge)[:,1]


    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates
 of the positive 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))



for i in tqdm(random_alpha_values):
    log_alphas.append(math.log(i))


# plotiing the auc score vs the Alpha value lets see which suits the best

plt.figure(figsize = (30,15))
plt.plot(log_alphas, train_auc, label='Train AUC')
plt.plot(log_alphas, cv_auc, label='CV AUC')

plt.scatter(log_alphas, train_auc, label='Train AUC co-ordinates')
plt.scatter(log_alphas, cv_auc, label='CV AUC co-ordinates')

plt.legend()
plt.xlabel("log alpha as  hyperparameter")
plt.ylabel("AUC Score")
plt.title("alpha as the  hyperparameter v/s AUC Score")
plt.grid()
plt.show()
```

```
100%|████████████████████████████████████████████████████████████████| 17/17 [00:02<00:00,  7.42it/s]
100%|████████████████████████████████████████████████████████████████| 17/17 [00:00<00:00, 17050.02it/s]
```



From above we can see that we considered a wide range of values for alpha randomly ranging from 0.00001 to 1000 and we can see from our CV_AUC score that the best AUC score is received at alpha = 0.5 (after taking the anti log). The above plot is showing us that taking very less values of alpha or taking very large values of alpha does not help in the creation of the best model hence we should avoid that.

We can also see that the train data is getting best performance for very low value of alpha this directly links to the overfitting scenario and this is happening because in our data we have very rare words also and we associate a probability score along with them also and one should notice that they have very small numnemator hence if there is a small change in the training data (as very rare words have high probability of not falling under the train category everytime) will change the model drastically which is a direct sign of overfitting

Hence values of alpha around 0.00001 works the best for training data but worst for the cross validation data and the values of alpha around 1 works considerably well for the training data as well as the cross validation data and lastkly the values of alphja as large as 2500 works worst for training as well as cross validation data.

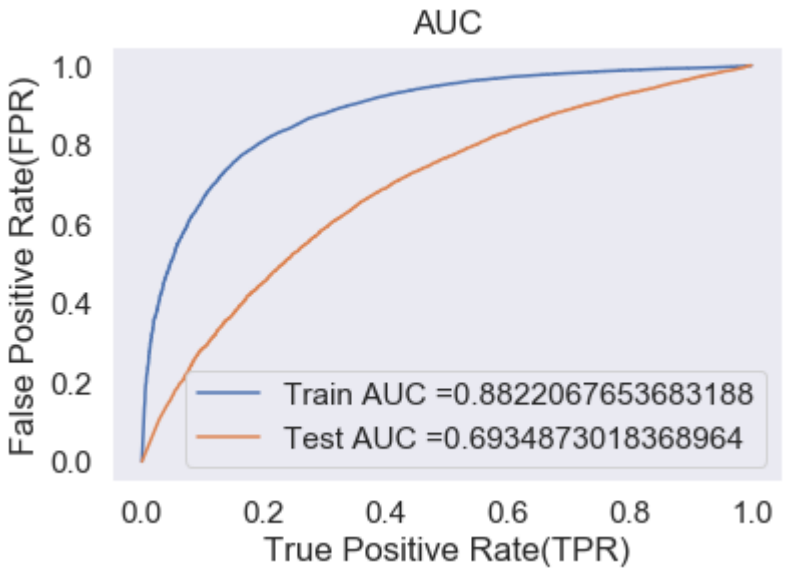
In [703]:

```
# hence selecting the best alpha value
best_alpha_bow = 0.5
```

Training our model using the best value of alpha from above that is alpha = 0.5

In [704]:

```
nb_BOW = MultinomialNB(alpha = best_alpha_bow)

nb_BOW.fit(X_train_merge, y_train)

# this below line of code is giving us the probability of a feature beinf present given
to us that it belongs to class 1

y_train_pred =  nb_BOW.predict_proba(X_train_merge)[:,1]

y_test_pred =  nb_BOW.predict_proba(X_test_merge)[:,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("True Positive Rate(TPR)")
plt.ylabel("False Positive Rate(FPR)")
plt.title("AUC")
plt.grid()
plt.show()
```



Creating the confusion matrix for the above results

In [705]:

```
# 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.rou
nd(t,3))
    predictions = []
    for i in proba:
        if i>=t:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
```

Confusion matrix for train and test data for BOW vectorization

In [706]:

```
print("="*100)
from sklearn.metrics import confusion_matrix
print("Train confusion matrix")
print(confusion_matrix(y_train, predict(y_train_pred, tr_thresholds, train_fpr, train_f
pr)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict(y_test_pred, tr_thresholds, test_fpr, test_fpr
)))
```

```
====================================================================================
==========================
Train confusion matrix
the maximum value of tpr*(1-fpr) 0.2499999818661462 for threshold 0.0
[[ 3712  3714]
 [ 1992 39623]]
Test confusion matrix
the maximum value of tpr*(1-fpr) 0.24999999161092998 for threshold 0.15
[[ 1951  3508]
 [ 4291 26302]]
```

Visually plotting the confusion matrix for training data

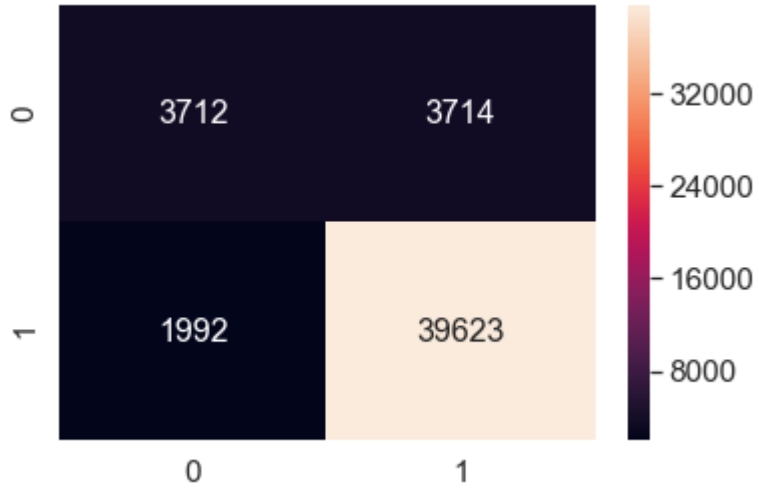
In [707]:

```
# Code for this segment from here -->> https://stackoverflow.com/questions/35572000/how
-can-i-plot-a-confusion-matrix

import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt


df_cm = pd.DataFrame(confusion_matrix(y_train,predict(y_train_pred, tr_thresholds, trai
n_fpr, train_fpr)), range(2),
                     range(2))
#plt.figure(figsize = (10,7))
sn.set(font_scale=1.4)#for label size
sn.heatmap(df_cm, annot=True,annot_kws={"size": 16},fmt='g')# font size
```

```
the maximum value of tpr*(1-fpr) 0.2499999818661462 for threshold 0.0
```

Out[707]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x19e0e7f2e80>
```



Visually plotting the confusion matrix for test data

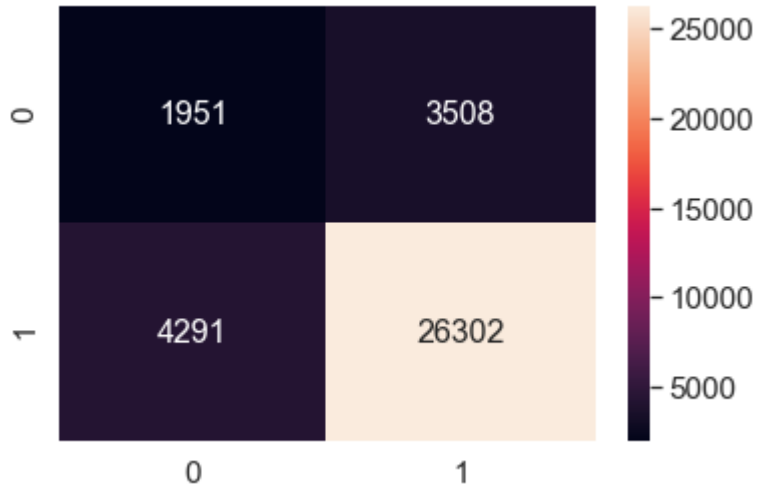
In [708]:

```
# Code for this segment from here -->> https://stackoverflow.com/questions/35572000/how
-can-i-plot-a-confusion-matrix

import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

df_cm_test = pd.DataFrame(confusion_matrix(y_test,predict(y_test_pred, tr_thresholds, t
est_fpr, test_fpr)), range(2),
                  range(2))
#plt.figure(figsize = (10,7))
sn.set(font_scale=1.4)#for label size
sn.heatmap(df_cm_test, annot=True,annot_kws={"size": 16},fmt='g')# font size
```

```
the maximum value of tpr*(1-fpr) 0.24999999161092998 for threshold 0.15
```

Out[708]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x19e0d2cc630>
```



Top 10 features from the positive class

In [709]:

```
len(all_feature_names_bow)
```

Out[709]:

```
100102
```

In [710]:

```
#all_feature_names_bow.extend(["price","quantity","teacher_number_of_previously_posted_
projects"])
```

In [711]:

```
len(all_feature_names_bow)
```

Out[711]:

```
100102
```

In [712]:

```
nb_BOW.feature_log_prob_.shape
```

Out[712]:

```
(2, 100102)
```

Here note that 2 corresponds to the two rows we have where one row corresponds to the positive class while the second row corresponds to the negative class

all_feature_names basically consist of all the features using which our model was trained

Now lets get all the feature probabilities corresponding to positive class

In [713]:

```
feature_probability_bow_positive = []

for j in range(len(all_feature_names_bow)):
    feature_probability_bow_positive.append(nb_BOW.feature_log_prob_[1,j])

#feature_probability_bow.shape

# note that i put [1,j] above instead of [0,j] because i want the positive class featur
e probabilities and it is in second row i.e index  1
```

In [714]:

```
feature_probability_bow_positive[4]
```

Out[714]:

```
-7.690692544964957
```

In [715]:

```
len(all_feature_names_bow)
```

Out[715]:

```
100102
```

In [716]:

```
type(feature_probability_bow_positive)
```

Out[716]:

```
list
```

In [717]:

```
len(feature_probability_bow_positive)
```

Out[717]:

```
100102
```

Now we will be creating a dataframe which will consist of the feature name along wtih the probability of having the feature if our data belonged to class 1 or the positive class or we can say that the project was approved

In [718]:

```
bow_features_with_probability_positive_class = pd.DataFrame({'feature_names_bow' : all_
feature_names_bow , 'feature_probabiltiy' : feature_probability_bow_positive })
```

In [719]:

```
bow_features_with_probability_positive_class.shape
```

Out[719]:

```
(100102, 2)
```

In [720]:

```
bow_features_with_probability_positive_class_sorted = bow_features_with_probability_pos
itive_class.sort_values(by = ['feature_probabiltiy'],ascending = False)
```

Top 10 features of the Positive Class using BOW vectorization

In [721]:

```
bow_features_with_probability_positive_class_sorted.head(10)
```

Out[721]:

	feature_names_bow	feature_probabiltiy
40280	students	-3.468348
35980	school	-4.618918
23200	learning	-4.976545
7270	classroom	-5.008127
29221	not	-5.277137
22557	learn	-5.313118
18744	help	-5.342802
100100	quantity	-5.459496
100099	price	-5.459496
26122	many	-5.484805

Least Important 10 features of the Positive Class using BOW vectorization

In [722]:

```
bow_features_with_probability_positive_class_sorted.tail(10)
```

Out[722]:

	feature_names_bow	feature_probabiltiy
90552	safe and	-16.788872
90553	safe and sound	-16.788872
90554	safe by	-16.788872
86806	recycle club	-16.788872
69496	mrs masciel classroom	-16.788872
72225	need is friend	-16.788872
72224	need is	-16.788872
72223	need ipod	-16.788872
90555	safe by using	-16.788872
67168	mirroring gnirorrim dual	-16.788872

Now lets get all the feature probabilities corresponding to negative class

In [723]:

```
feature_probability_bow_negative = []

for j in range(len(all_feature_names_bow)):
    feature_probability_bow_negative.append(nb_BOW.feature_log_prob_[0,j])



# note that i put [0,j] above instead of [0,j] because i want the negative class featur
e probabilities and it is in second row i.e index  1
```

In [724]:

```
feature_probability_bow_negative[7]
```

Out[724]:

```
-6.361565630922475
```

In [725]:

```
type(feature_probability_bow_negative)
```

Out[725]:

```
list
```

In [726]:

```
len(feature_probability_bow_negative)
```

Out[726]:

```
100102
```

Now we will be creating a dataframe which will consist of the feature name along wtih the probability of having the feature if our data belonged to class 0 or the negative class or we can say that the project was not approved

In [727]:

```
bow_features_with_probability_negative_class = pd.DataFrame({'feature_names_bow' : all_
feature_names_bow , 'feature_probabiltiy' : feature_probability_bow_negative })
```

In [728]:

```
bow_features_with_probability_negative_class.shape
```

Out[728]:

```
(100102, 2)
```

In [729]:

```
bow_features_with_probability_negative_class_sorted = bow_features_with_probability_pos
itive_class.sort_values(by = ['feature_probabiltiy'],ascending = False)
```

Top 10 features of the Negative Class using BOW vectorization

In [730]:

```
bow_features_with_probability_negative_class_sorted.head(10)
```

Out[730]:

	feature_names_bow	feature_probabiltiy
40280	students	-3.468348
35980	school	-4.618918
23200	learning	-4.976545
7270	classroom	-5.008127
29221	not	-5.277137
22557	learn	-5.313118
18744	help	-5.342802
100100	quantity	-5.459496
100099	price	-5.459496
26122	many	-5.484805

Least Important 10 features of the Negative Class using BOW vectorization

In [731]:

```
bow_features_with_probability_negative_class_sorted.tail(10)
```

Out[731]:

	feature_names_bow	feature_probabiltiy
90552	safe and	-16.788872
90553	safe and sound	-16.788872
90554	safe by	-16.788872
86806	recycle club	-16.788872
69496	mrs masciel classroom	-16.788872
72225	need is friend	-16.788872
72224	need is	-16.788872
72223	need ipod	-16.788872
90555	safe by using	-16.788872
67168	mirroring gnirorrim dual	-16.788872

We are getting most of the features same and this is happening because there are lot of common words in both the positive class and negative class data

Set 2 : categorical, numerical features + project_title(TFIDF) + preprocessed_essay (TFIDF)

Now we need to merge all the numerical vectors(categorical features,text features,numerical features) given above for set-2 which we created using different methods

In [732]:

```
from scipy.sparse import hstack

X_train_merge_set_2 = hstack((categories_one_hot_train,subcategories_one_hot_train,teac
her_prefix_one_hot_train,project_grade_categories_one_hot_train,school_state_categories
_one_hot_train,text_tfidf_train,project_title_tfidf_train,price_train,quantity_train,pr
ev_projects_train)).tocsr()
X_test_merge_set_2 = hstack((categories_one_hot_test,subcategories_one_hot_test,teacher
_prefix_one_hot_test,project_grade_categories_one_hot_test,school_state_categories_one_
hot_test,text_tfidf_test,project_title_tfidf_test,price_test,quantity_test,prev_project
s_test)).tocsr()
X_cv_merge_set_2 = hstack((categories_one_hot_cv,subcategories_one_hot_cv,teacher_prefi
x_one_hot_cv,project_grade_categories_one_hot_cv,school_state_categories_one_hot_cv,tex
t_tfidf_cv,title_tfidf_cv,price_cv,quantity_cv,prev_projects_cv)).tocsr()
```

In [733]:

```
# this will be our finally created data matrix dimensions

print(X_train_merge_set_2.shape, y_train.shape)
print(X_test_merge_set_2.shape, y_test.shape)
print(X_cv_merge_set_2.shape, y_cv.shape)
```

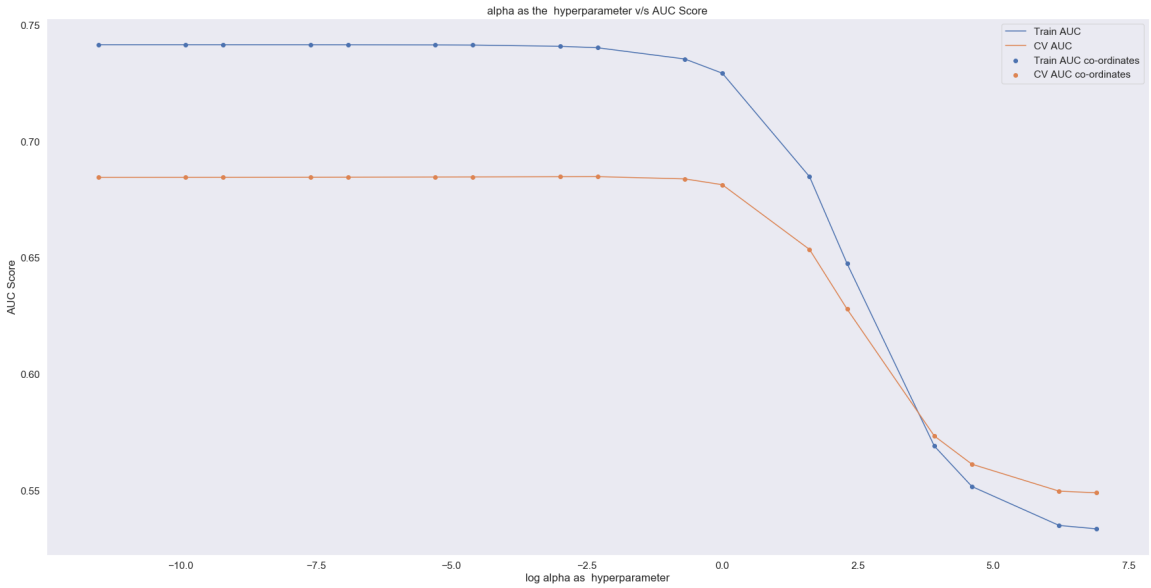
```
(49041, 8102) (49041,)
(36052, 8102) (36052,)
(24155, 8102) (24155,)
```

Trying random Alpha Values To get the best one

In [734]:

```
train_auc = []
cv_auc = []
log_alphas = []

random_alpha_values = [0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1,
0.5, 1, 5, 10, 50, 100, 500, 1000]

for j in tqdm(random_alpha_values):

    nb = MultinomialNB(alpha = j)

    nb.fit(X_train_merge_set_2, y_train)

    y_train_pred =  nb.predict_proba(X_train_merge_set_2)[:,1]

    y_cv_pred =  nb.predict_proba(X_cv_merge_set_2)[:,1]


    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates
 of the positive 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))



for i in tqdm(random_alpha_values):
    log_alphas.append(math.log(i))


# plotiing the auc score vs the Alpha value lets see which suits the best

plt.figure(figsize = (30,15))
plt.plot(log_alphas, train_auc, label='Train AUC')
plt.plot(log_alphas, cv_auc, label='CV AUC')

plt.scatter(log_alphas, train_auc, label='Train AUC co-ordinates')
plt.scatter(log_alphas, cv_auc, label='CV AUC co-ordinates')

plt.legend()
plt.xlabel("log alpha as  hyperparameter")
plt.ylabel("AUC Score")
plt.title("alpha as the  hyperparameter v/s AUC Score")
plt.grid()
plt.show()
```

```
100%|████████████████████████████████████████████████████████████████| 17/17 [00:01<00:00, 10.93it/s]
100%|████████████████████████████████████████████████████████████████| 17/17 [00:00<?, ?it/s]
```



From above we can see that the best AUC score we are getting at alpha value = 0.1(log value) hence we will be taking this value to train our model now.

Very less value of alpha(0.00001) leads to best performance for the train data but not the CV data

AUC score is constant for most of the values less than 1 throughout changing the alpha value is not causing any change in the AUC score.

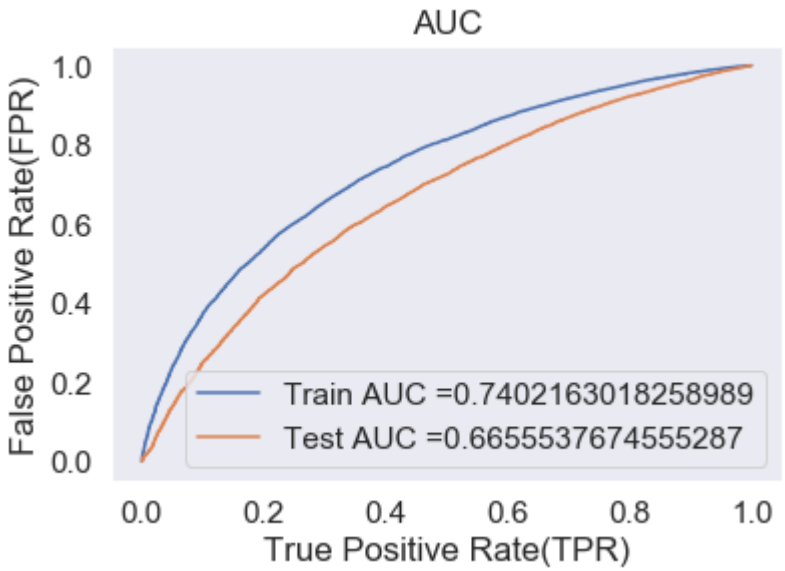
In [735]:

```
best_alpha_tfidf = 0.1
```

Training our model using the best value of alpha from above that is alpha = 0.1

In [736]:

```
nb_TFIDF = MultinomialNB(alpha = best_alpha_tfidf)

nb_TFIDF.fit(X_train_merge_set_2, y_train)

y_train_pred =  nb_TFIDF.predict_proba(X_train_merge_set_2)[:,1]

y_test_pred =  nb_TFIDF.predict_proba(X_test_merge_set_2)[:,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("True Positive Rate(TPR)")
plt.ylabel("False Positive Rate(FPR)")
plt.title("AUC")
plt.grid()
plt.show()
```



Creating the confusion matrix for the above results

Confusion matrix for train and test data for BOW vectorization

In [737]:

```
print("="*100)
from sklearn.metrics import confusion_matrix
print("Train confusion matrix")
print(confusion_matrix(y_train, predict(y_train_pred, tr_thresholds, train_fpr, train_f
pr)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict(y_test_pred, tr_thresholds, test_fpr, test_fpr
)))
```

```
====================================================================================
==========================
Train confusion matrix
the maximum value of tpr*(1-fpr) 0.2499999818661462 for threshold 0.771
[[ 3712  3714]
 [ 7790 33825]]
Test confusion matrix
the maximum value of tpr*(1-fpr) 0.24999999161092998 for threshold 0.827
[[ 2944  2515]
 [ 9237 21356]]
```

Visually plotting the confusion matrix for training data

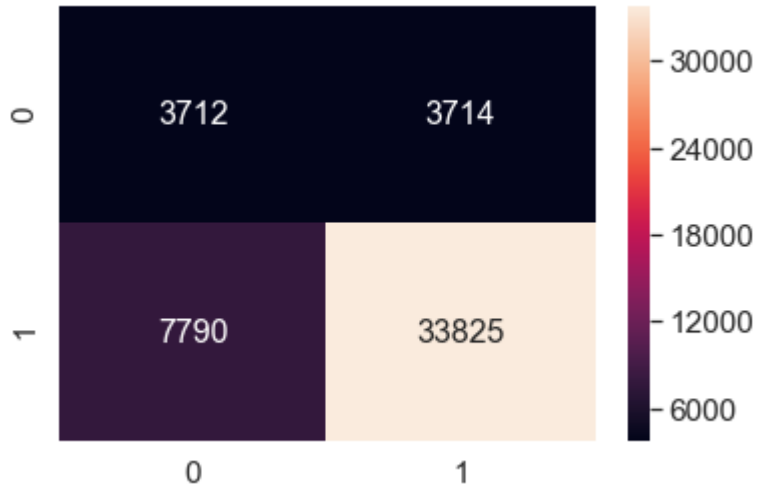
In [738]:

```
# Code for this segment from here -->> https://stackoverflow.com/questions/35572000/how
-can-i-plot-a-confusion-matrix

import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt


df_cm = pd.DataFrame(confusion_matrix(y_train,predict(y_train_pred, tr_thresholds, trai
n_fpr, train_fpr)), range(2),
                     range(2))
#plt.figure(figsize = (10,7))
sn.set(font_scale=1.4)#for label size
sn.heatmap(df_cm, annot=True,annot_kws={"size": 16},fmt='g')# font size
```

```
the maximum value of tpr*(1-fpr) 0.2499999818661462 for threshold 0.771
```

Out[738]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x19e589a40b8>
```



Visually plotting the confusion matrix for test data

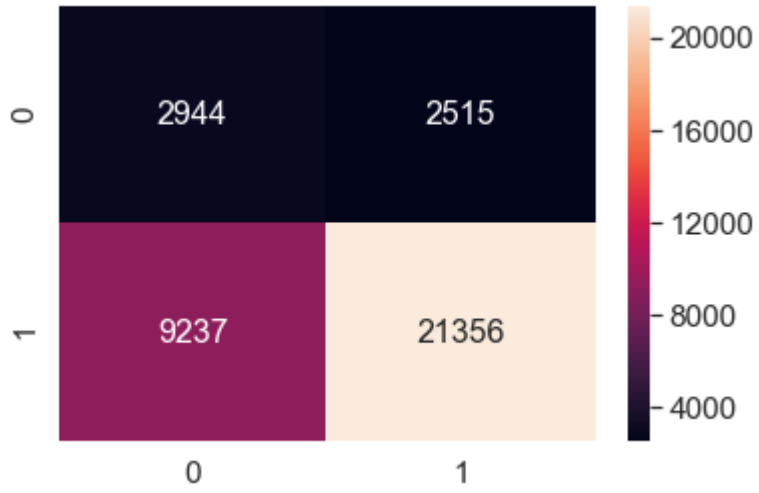
In [739]:

```
# Code for this segment from here -->> https://stackoverflow.com/questions/35572000/how
-can-i-plot-a-confusion-matrix

import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

df_cm_test = pd.DataFrame(confusion_matrix(y_test,predict(y_test_pred, tr_thresholds, t
est_fpr, test_fpr)), range(2),
                  range(2))
#plt.figure(figsize = (10,7))
sn.set(font_scale=1.4)#for label size
sn.heatmap(df_cm_test, annot=True,annot_kws={"size": 16},fmt='g')# font size
```

```
the maximum value of tpr*(1-fpr) 0.24999999161092998 for threshold 0.827
```

Out[739]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x19e270f4978>
```



Top 10 features from the positive class

In [ ]:

```

```

In [740]:

```
len(all_feature_names_tfidf)
```

Out[740]:

```
8102
```

In [741]:

```
nb_TFIDF.feature_log_prob_.shape
```

Out[741]:

```
(2, 8102)
```

Here note that 2 corresponds to the two rows we have where one row corresponds to the positive class while the second row corresponds to the negative class

all_feature_names basically consist of all the features using which our model was trained

Now lets get all the feature probabilities corresponding to positive class

In [743]:

```
feature_probability_tfidf_positive = []

for j in range(len(all_feature_names_tfidf)):
    feature_probability_tfidf_positive.append(nb_TFIDF.feature_log_prob_[1,j])

#feature_probability_bow.shape

# note that i put [1,j] above instead of [0,j] because i want the positive class featur
e probabilities and it is in second row i.e index  1
```

In [744]:

```
len(feature_probability_tfidf_positive)
```

Out[744]:

```
8102
```

Now we will be creating a dataframe which will consist of the feature name along wtih the probability of having the feature if our data belonged to class 1 or the positive class or we can say that the project was approved

In [745]:

```
tfidf_features_with_probability_positive_class = pd.DataFrame({'feature_names_bow' : al
l_feature_names_tfidf , 'feature_probabiltiy' : feature_probability_tfidf_positive })
```

In [747]:

```
tfidf_features_with_probability_positive_class.shape
```

Out[747]:

```
(8102, 2)
```

In [748]:

```
tfidf_features_with_probability_positive_class_sorted = tfidf_features_with_probability
_positive_class.sort_values(by = ['feature_probabiltiy'],ascending = False)
```

Top 10 most useful features for positive class using tfidf vectorization

In [749]:

```
tfidf_features_with_probability_positive_class_sorted.head(10)
```

Out[749]:

	feature_names_bow	feature_probabiltiy
8100	quantity	-2.928873
8099	price	-2.928873
8101	teacher_number_of_previously_posted_projects	-3.238066
8	Literacy_Language	-3.643022
7	Math_Science	-3.916941
38	Literacy	-4.073557
37	Mathematics	-4.296756
36	Literature_Writing	-4.509757
52	CA	-4.883214
6	Health_Sports	-4.972401

Least most useful 10 features for positive class using tfidf vectorization

In [750]:

```
tfidf_features_with_probability_positive_class_sorted.tail(10)
```

Out[750]:

	feature_names_bow	feature_probabiltiy
7335	oh the places	-11.728854
7769	supplies supplies	-11.771145
43	Dr.	-15.867676
41	Mrs.	-15.867676
40	Ms.	-15.867676
47	3-5	-15.867676
46	6-8	-15.867676
44	9-12	-15.867676
39	Mr.	-15.867676
45	PreK-2	-15.867676

Now lets get the feature probabilities corresponding to negative class

In [752]:

```
feature_probability_tfidf_negative = []

for j in range(len(all_feature_names_tfidf)):
    feature_probability_tfidf_negative.append(nb_TFIDF.feature_log_prob_[1,j])

#feature_probability_bow.shape

# note that i put [0,j] above instead of [1,j] because i want the negative class featur
e probabilities and it is in first row i.e index  0
```

In [753]:

```
len(feature_probability_tfidf_negative)
```

Out[753]:

```
8102
```

In [755]:

```
tfidf_features_with_probability_negative_class = pd.DataFrame({'feature_names_tfidf' :
all_feature_names_tfidf , 'feature_probabiltiy' : feature_probability_tfidf_negative })
```

In [756]:

```
tfidf_features_with_probability_negative_class_sorted = tfidf_features_with_probability
_negative_class.sort_values(by = ['feature_probabiltiy'],ascending = False)
```

Top 10 negative class features using the tfidf vectorization

In [757]:

```
tfidf_features_with_probability_negative_class_sorted.head(10)
```

Out[757]:

	feature_names_tfidf	feature_probabiltiy
8100	quantity	-2.928873
8099	price	-2.928873
8101	teacher_number_of_previously_posted_projects	-3.238066
8	Literacy_Language	-3.643022
7	Math_Science	-3.916941
38	Literacy	-4.073557
37	Mathematics	-4.296756
36	Literature_Writing	-4.509757
52	CA	-4.883214
6	Health_Sports	-4.972401

Least important 10 features of negative class

In [758]:

```
tfidf_features_with_probability_negative_class_sorted.tail(10)
```

Out[758]:

	feature_names_tfidf	feature_probabiltiy
7335	oh the places	-11.728854
7769	supplies supplies	-11.771145
43	Dr.	-15.867676
41	Mrs.	-15.867676
40	Ms.	-15.867676
47	3-5	-15.867676
46	6-8	-15.867676
44	9-12	-15.867676
39	Mr.	-15.867676
45	PreK-2	-15.867676

We can see again that we are getting the same features for both the positive as well as the negative class due to the same reason as stated previously.

Conclusions

Naive bayes is faster in comparison to KNN which we did before which is a good sign of using naive bayes(Clearly observable while running the algorithm)

In [759]:

```
from prettytable import PrettyTable

p = PrettyTable()

p.field_names = ["Vectorization Technique", "Model", "Alpha value(Hyper Parameter)", "A
UC Score"]


p.add_row(["TFIDF", "Naive Bayes", 0.1, 0.67])

p.add_row(["BOW", "Naive Bayes", 0.5, 0.69])

print(p)
```

```
+-------------------------+-------------+------------------------------+--
---------+
| Vectorization Technique |    Model    | Alpha value(Hyper Parameter) | A
UC Score |
+-------------------------+-------------+------------------------------+--
---------+
|          TFIDF          | Naive Bayes |             0.1              |
0.67   |
|           BOW           | Naive Bayes |             0.5              |
0.69   |
+-------------------------+-------------+------------------------------+--
---------+
```

We are getting the better test AUC for the BOW vectorization in comparison to tfidf vectorization(although the difference is very less but still it is there)

You might be thinking why did we use Naive Bayes when we had a classifier knn already done on the same data set the answer lies in the below cells.

In [760]:

```
from prettytable import PrettyTable

p = PrettyTable()

p.field_names = ["Vectorization Technique", "Model", "K value(Hyper Parameter)", "AUC S
core"]


p.add_row(["TFIDF", "KNN", 41, 0.56])

p.add_row(["BOW", "KNN", 31, 0.62])

print(p)
```

```
+-------------------------+-------+--------------------------+-----------+
| Vectorization Technique | Model | K value(Hyper Parameter) | AUC Score |
+-------------------------+-------+--------------------------+-----------+
|          TFIDF          |  KNN  |            41            |    0.56   |
|           BOW           |  KNN  |            31            |    0.62   |
+-------------------------+-------+--------------------------+-----------+
```

Naive bayes is giving better test Auc than the KNN model which we built before which is also a good sign to use the Naive bayes algorithm

Hence naive bayes Rocks!!!

In [ ]: