DonorsChoose

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

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

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

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

About the DonorsChoose Data Set

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

Feature	Description
`project_id`	A unique identifier for the proposed project. **Example:** `p036502`

Feature	Description
`project_title`	Title of the project. **Examples:** • `Art Will Make You Happy!` • `First Grade Fun`
`project_grade_category`	Grade level of students for which the project is targeted. One of the following enumerated values: • `Grades PreK-2` • `Grades 3-5` • `Grades 6-8` • `Grades 9-12`
`project_subject_categories`	One or more (comma-separated) subject categories for the project from the following enumerated list of values: • `Applied Learning` • `Care & Hunger` • `Health & Sports` • `History & Civics` • `Literacy & Language` • `Math & Science` • `Music & The Arts` • `Special Needs` • `Warmth` **Examples:** • `Music & The Arts` • `Literacy & Language, Math & Science`
`school_state`	State where school is located (Two-letter U.S. postal code). **Example:** `WY`
`project_subject_subcategories`	One or more (comma-separated) subject subcategories for the project. **Examples:** • `Literacy` • `Literature & Writing, Social Sciences`

Feature	Description
`project_resource_summary`	An explanation of the resources needed for the project. **Example:** • `My students need hands on literacy materials to manage sensory needs!`
`project_essay_1`	First application essay*
`project_essay_2`	Second application essay*
`project_essay_3`	Third application essay*
`project_essay_4`	Fourth application essay*
`project_submitted_datetime`	Datetime when project application was submitted. **Example:** `2016-04-28 12:43:56.245`
`teacher_id`	A unique identifier for the teacher of the proposed project. **Example:** `bdf8baa8fedef6bfeec7ae4ff1c15c56`
`teacher_prefix`	Teacher's title. One of the following enumerated values: • `nan` • `Dr.` • `Mr.` • `Mrs.` • `Ms.` • `Teacher.`
`teacher_number_of_previously_posted_projects`	Number of project applications previously submitted by the same teacher. **Example:** `2`

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

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

Feature	Description

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

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

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

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

Notes on the Essay Data

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

- __project_essay_1:__ "Introduce us to your classroom"
- __project_essay_2:__ "Tell us more about your students"
- __project_essay_3:__ "Describe how your students will use the materials you're requesting"
- __project_essay_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]:
```
%config IPCompleter.greedy=True
%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
import scipy

#from plotly import plotly
```

```
#import plotly.offline as offline
#import plotly.graph_objs as go
#offline.init_notebook_mode()
from collections import Counter
from sklearn.metrics import confusion_matrix
```

1.1 Reading Data

In [2]:
```
project_data = pd.read_csv('train_data.csv')
resource_data = pd.read_csv('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_prefi
x' '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/2
582163/4084039
cols = ['Date' if x=='project_submitted_datetime' else x for x in list(
project_data.columns)]
#sort dataframe based on time pandas python: https://stackoverflow.com/
a/49702492/4084039
project_data['Date'] = pd.to_datetime(project_data['project_submitted_d
atetime'])
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/131
48611/4084039
```

```
project_data = project_data[cols]
project_data.head(2)
```

Out[4]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	D
86221	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs.	CA	20 04 00:27
18308	37728	p043609	3f60494c61921b3b43ab61bdde2904df	Ms.	UT	20 04 00:31

In [5]:
```
project_data.sample
```

Out[5]:
```
<bound method NDFrame.sample of         Unnamed: 0       id
            teacher_id teacher_prefix  \
86221         8393  p205479  2bf07ba08945e5d8b2a3f269b2b3cfe5
Mrs.
18308        37728  p043609  3f60494c61921b3b43ab61bdde2904df
  Ms.
79692        74477  p189804  4a97f3a390bfe21b99cf5e2b81981c73
Mrs.
98588       100660  p234804  cbc0e38f522143b86d372f8b43d4cff3
Mrs.
57724        33679  p137682  06f6e62e17de34fcf81020c77549e1d5
Mrs.
41908       146723  p099708  c0a28c79fe8ad5810da49de47b3fb491
Mrs.
65282        95963  p155767  e50367a62524e11fbd2dc79651b6df21
Mrs.
76584       139722  p182545  22460c54072bd0cf958cc8349fac8b8f
  Ms.
36938        72317  p087808  598621c141cda5fb184ee7e8ccdd3fcc
  Ms.
20668       114684  p049177  679f50f18ce50aabcc602d17f7627206
```

```
Mrs.
41786        57854  p099430  4000cfe0c8b2df75a218347c1765e283
  Ms.
50434       166022  p120079  8e22592f19b346df505bbdf6144c28d5
  Mr.
38495        79341  p091436  bb2599c4a114d211b3381abe9f899bf8
Mrs.
100376      128817  p239087  11a60ddd63717c59fdd5a13ea92d34aa
Mrs.
85477       127145  p203619  85c61480f0eaea60734523665a3838b4
Mrs.
108373      104404  p258140  341dc52d3229176eda913da90b6c19c7
Mrs.
54927       149397  p131036  bf5bf59287e7c676a634a00284596b64
Mrs.
83930       179302  p199881  82ae813a6e2dc0da592de93861a69561
Mrs.
85415        50256  p203475  63e9a9f2c9811a247f1aa32ee6f92644
Mrs.
61734       139237  p147271  7f2072d18c67087af27066f60b2c9d85
Mrs.
94408       146737  p224791  ff5d658932d9ad0d9ebedabea582648e
Mrs.
24494        14427  p058390  578585b8ab7349189837e9618ca0f7f4
Mrs.
26045        59671  p061990  03093ad866c578b107d5be6957837c5f
  Mr.
9844         12619  p023504  18c82623ff01c59e593f7d81ab11e62c
  Ms.
82550       148085  p196567  171f782b55614c56213131bcb8d44e06
Mrs.
58243       121622  p138958  57626865698278199f753dc0f8e3ed00
  Ms.
38778       135897  p092089  44ab4df75ae4e8b9bb23b818a7a1b1a4
Mrs.
57338       123970  p136762  362f046c8551fa0b2515f99d6e3ce6ea
  Mr.
17757       165036  p042345  3c2efbcac105fc8a55df610ed03f4e77
Mrs.
```

```
77852        107719  p185470  a3a628eaf8728f585e0e906345d9e504
  Ms.
...             ...      ...                                ...
  ...
4966          73173  p011863  e9a57ff541d9965373d9f05baec6dbb9
Mrs.
77875         63258  p185518  783c9da904d2902781a4205a8a6f2cf2
  Ms.
39452        110157  p093760  7c0bb16f949a62e91151789662b27675
Mrs.
47553        180842  p113135  ea758136dee04fab896aac935276161d
Mrs.
40182         94252  p095511  d0b4f2709a391b3953bca7e4d0655992
  Ms.
49169         67820  p117003  a22232bad54f69e12f379fe86f3f8828
  Ms.
58987         84481  p140704  e36637824051b8a2edc16c6ec0eb4832
  Mr.
33573         28930  p079867  6240693c06f02e3bb63e89afa413f379
  Mr.
15399        180481  p036737  b2f85df8fe445189b1e56d7b6561adbe
Mrs.
68038        169090  p162286  68c376fb8289fafb9831d0c886669fd1
  Ms.
16178         71940  p038577  e8e0311f1765ef3a427a0c4da811a5fe
  Ms.
90442         28565  p215499  194004c4aee808bcd24deff39b3acdb8
  Ms.
9114          20564  p021779  504e698d91890380ff7e278e3918bb2f
  Mr.
62614        150872  p149431  f308097ab4af3a20ad3d96b13083b9c4
  Ms.
31942        180953  p075974  3654cb255584baee31fded55e9fa593b
Mrs.
3203          61360  p007550  05677e17e14429f6942245da50bd3da4
Mrs.
102120        60690  p243246  1c6ad7948ab442bad6f72fd8ad64dd7f
Mrs.
90456          3550  p215525  f09efb73f135c77ed938ca4df6a33ff5
```

```
  Ms.
72968       171039  p173915  be550b77ce85b91080a76df5d3d9bf17
  Mr.
83322       167772  p198397  9a934dc531c21fd9392e6b70bd1c38ea
Mrs.
62551        41021  p149303  dc07f461cc1b8767846023734c44cc43
Mrs.
72622       103254  p173126  05c9cc90376e1d7fde1b7138f2bc4d8d
  Ms.
108171       19563  p257657  b55850e67a9e5d917958c43082be9e9b
  Ms.
28405        34853  p067693  63ab3770bef577efb8b738303ef19e54
  Ms.
93432       175286  p222440  51ffd84df3423f8d5e38943e54b8388e
Mrs.
81838        45036  p194916  29cf137e5a40b0f141d9fd7898303a5c
Mrs.
68314        12610  p162971  22fee80f2078c694c2d244d3ecb1c390
  Ms.
40730       179833  p096829  c8c81a73e29ae3bdd4140be8ad0bea00
Mrs.
77397        13791  p184393  65545a295267ad9df99f26f25c978fd0
Mrs.
11845       124250  p028318  1fff5a88945be8b2c728c6a85c31930f
Mrs.

       school_state                Date project_grade_category  \
86221            CA 2016-04-27 00:27:00          Grades PreK-2
18308            UT 2016-04-27 00:31:00             Grades 3-5
79692            CA 2016-04-27 00:46:00          Grades PreK-2
98588            GA 2016-04-27 00:53:00          Grades PreK-2
57724            WA 2016-04-27 01:05:00             Grades 3-5
41908            CA 2016-04-27 01:10:00             Grades 3-5
65282            CA 2016-04-27 01:29:00             Grades 3-5
76584            CA 2016-04-27 02:02:00             Grades 3-5
36938            CA 2016-04-27 02:04:00          Grades PreK-2
20668            HI 2016-04-27 02:18:00             Grades 3-5
41786            IL 2016-04-27 07:19:00          Grades PreK-2
50434            OH 2016-04-27 07:24:00            Grades 9-12
```

38495	OH	2016-04-27 07:24:00	Grades PreK-2
100376	KY	2016-04-27 08:02:00	Grades 3-5
85477	SC	2016-04-27 08:06:00	Grades 3-5
108373	SC	2016-04-27 08:23:00	Grades PreK-2
54927	FL	2016-04-27 08:42:00	Grades 3-5
83930	OH	2016-04-27 08:43:00	Grades 3-5
85415	CA	2016-04-27 08:45:00	Grades 3-5
61734	MO	2016-04-27 08:51:00	Grades 9-12
94408	MI	2016-04-27 08:51:00	Grades PreK-2
24494	NY	2016-04-27 09:02:00	Grades 3-5
26045	CA	2016-04-27 09:03:00	Grades 9-12
9844	NY	2016-04-27 09:08:00	Grades 3-5
82550	VA	2016-04-27 09:08:00	Grades 3-5
58243	GA	2016-04-27 09:13:00	Grades PreK-2
38778	MD	2016-04-27 09:15:00	Grades PreK-2
57338	TX	2016-04-27 09:17:00	Grades PreK-2
17757	MS	2016-04-27 09:18:00	Grades 3-5
77852	AZ	2016-04-27 09:33:00	Grades PreK-2
...	...	...	...
4966	MD	2017-04-30 20:21:00	Grades PreK-2
77875	MO	2017-04-30 20:33:00	Grades PreK-2
39452	MI	2017-04-30 20:50:00	Grades 3-5
47553	NC	2017-04-30 20:55:00	Grades PreK-2
40182	OK	2017-04-30 20:56:00	Grades PreK-2
49169	CA	2017-04-30 21:03:00	Grades 9-12
58987	CA	2017-04-30 21:25:00	Grades PreK-2
33573	NE	2017-04-30 21:26:00	Grades 3-5
15399	WI	2017-04-30 21:30:00	Grades PreK-2
68038	OH	2017-04-30 21:39:00	Grades 3-5
16178	IL	2017-04-30 21:42:00	Grades PreK-2
90442	IN	2017-04-30 21:42:00	Grades PreK-2
9114	CA	2017-04-30 21:53:00	Grades 6-8
62614	LA	2017-04-30 21:53:00	Grades 3-5
31942	CA	2017-04-30 22:01:00	Grades 3-5
3203	CA	2017-04-30 22:02:00	Grades PreK-2
102120	FL	2017-04-30 22:07:00	Grades PreK-2
90456	GA	2017-04-30 22:09:00	Grades 3-5
72968	NY	2017-04-30 22:18:00	Grades 9-12
83322	IN	2017-04-30 22:24:00	Grades PreK-2

```
62551            FL 2017-04-30 22:35:00          Grades PreK-2
72622            IN 2017-04-30 22:36:00             Grades 6-8
108171           CA 2017-04-30 22:42:00            Grades 9-12
28405            CA 2017-04-30 23:06:00             Grades 3-5
93432            CT 2017-04-30 23:10:00          Grades PreK-2
81838            HI 2017-04-30 23:11:00            Grades 9-12
68314            NM 2017-04-30 23:23:00          Grades PreK-2
40730            IL 2017-04-30 23:25:00             Grades 3-5
77397            HI 2017-04-30 23:27:00            Grades 9-12
11845            CA 2017-04-30 23:45:00          Grades PreK-2

                   project_subject_categories  \
86221                         Math & Science
18308                          Special Needs
79692                    Literacy & Language
98588                       Applied Learning
57724                    Literacy & Language
41908        Math & Science, History & Civics
65282     Literacy & Language, Math & Science
76584        Math & Science, History & Civics
36938                    Literacy & Language
20668                         Math & Science
41786                    Literacy & Language
50434      Applied Learning, Music & The Arts
38495        Math & Science, Applied Learning
100376    Math & Science, Literacy & Language
85477                         Math & Science
108373    Literacy & Language, Math & Science
54927   History & Civics, Literacy & Language
83930                    Literacy & Language
85415                    Literacy & Language
61734                    Literacy & Language
94408       Applied Learning, Health & Sports
24494        Math & Science, Music & The Arts
26045                    Literacy & Language
9844                           Special Needs
82550                         Math & Science
58243                    Literacy & Language
38778                    Literacy & Language
```

```
57338                     Literacy & Language
17757        Math & Science, Music & The Arts
77852   Applied Learning, Literacy & Language
...                                       ...
4966                         Applied Learning
77875        Math & Science, Music & The Arts
39452     Literacy & Language, Math & Science
47553     Literacy & Language, Math & Science
40182                     Literacy & Language
49169          Health & Sports, Special Needs
58987     Literacy & Language, Math & Science
33573                     Literacy & Language
15399     Literacy & Language, Math & Science
68038                     Literacy & Language
16178                        Applied Learning
90442                     Literacy & Language
9114                         Applied Learning
62614                            Special Needs
31942                           Health & Sports
3203      Literacy & Language, Math & Science
102120          Math & Science, Special Needs
90456                            Special Needs
72968                        Music & The Arts
83322                     Literacy & Language
62551                          Math & Science
72622                        Music & The Arts
108171                       Applied Learning
28405         Applied Learning, Special Needs
93432                     Literacy & Language
81838                           Health & Sports
68314                        Applied Learning
40730                          Math & Science
77397                          Math & Science
11845                     Literacy & Language

                    project_subject_subcategories  \
86221     Applied Sciences, Health & Life Science
18308                               Special Needs
79692                                    Literacy
```

```
98588                          Early Development
57724                                    Literacy
41908                Mathematics, Social Sciences
65282                       Literacy, Mathematics
76584       Applied Sciences, History & Geography
36938                               ESL, Literacy
20668               Applied Sciences, Mathematics
41786                                    Literacy
50434                Extracurricular, Visual Arts
38495         Applied Sciences, Early Development
100376            Environmental Science, Literacy
85477     Applied Sciences, Environmental Science
108373                      Literacy, Mathematics
54927   History & Geography, Literature & Writing
83930                                    Literacy
85415                                    Literacy
61734             Literacy, Literature & Writing
94408            Early Development, Gym & Fitness
24494          Environmental Science, Visual Arts
26045             Literacy, Literature & Writing
9844                                Special Needs
82550          Environmental Science, Mathematics
58243                               ESL, Literacy
38778                               ESL, Literacy
57338                               ESL, Literacy
17757               Applied Sciences, Visual Arts
77852                 Early Development, Literacy
...                                           ...
4966                           Early Development
77875                    Mathematics, Visual Arts
39452                       Literacy, Mathematics
47553           Literature & Writing, Mathematics
40182                                    Literacy
49169            Health & Wellness, Special Needs
58987                       Literacy, Mathematics
33573                                    Literacy
15399                       Literacy, Mathematics
68038                        Literature & Writing
16178      Character Education, Early Development
```

```
90442                                ESL, Literacy
9114                               Extracurricular
62614                                Special Needs
31942                                Gym & Fitness
3203            Literature & Writing, Mathematics
102120            Applied Sciences, Special Needs
90456                                Special Needs
72968                                  Visual Arts
83322                                     Literacy
62551                                  Mathematics
72622                          Music, Visual Arts
108171                       College & Career Prep
28405          Character Education, Special Needs
93432                         Literature & Writing
81838                                  Team Sports
68314                     Early Development, Other
40730     Applied Sciences, Environmental Science
77397                        Health & Life Science
11845              Literacy, Literature & Writing

                                          project_title  \
86221        Engineering STEAM into the Primary Classroom
18308                              Sensory Tools for Focus
79692      Mobile Learning with a Mobile Listening Center
98588              Flexible Seating for Flexible Learning
57724              Going Deep: The Art of Inner Thinking!
41908                  Breakout Box to Ignite Engagement!
65282   Flexible Seating: An Environment to Help Kids ...
76584               21st Century Learning with Multimedia
36938                                   iPad for Learners
20668                     Dash and Dot Robotic Duo Needed
41786            A flexible classroom for flexible minds!
50434                               Make Powerful Movies!!
38495                     Robots are Taking over 2nd Grade
100376  Time for Kids....To Learn About Science and more!
85477   STEM: Books, Games, and Kits to explore our WORLD
108373                       Getting Plugged into Learning
54927               Help us travel the world...VIRTUALLY!
83930                     Engaging Readers With Technology
```

```
85415                 Books to Power Powerful Book Clubs!
61734   Choice Novels for Freshman Students are Needed...
94408                          Pre-K Classroom Materials
24494             Duct, Duct, Craft...Spring Inspiration
26045                      Discovering Our Best Selves!
9844                    reading, writing and technology!
82550                                 Techies in Training
58243                      Literacy Classroom Materials
38778                                   Literacy Centers
57338   Bilingual and Spanish Books for our Elementary...
17757          Coming Soon! After School Photography Club
77852             STEM Kindergartners Express Themselves!
...                                                    ...
4966             Creating An Environment For All Learners
77875   Beginning Teacher, Eager Students, Endless Pos...
39452                     A Rug for Reading and Meeting!
47553                Family Time During Summer Break Pt.2
40182                     Learning Together in 2nd Grade!
49169            Art4Healing Project: Expressing Emotions
58987                     A Clean Place to Sit and Learn
33573               Nebraska Golden Sower Nominated Books
15399          Flexible Seating for Personalized Learning
68038        Exploring Literature With Graphic Toon Books
16178               Educating the Young While Having Fun!
90442                                Em\"POWER\" Reading!
9114             Academic Achievement Through Chess: Yes!
62614                      Help for Headphones and More
31942                          THE MIND-BODY CONNECTION
3203    Headphones Help Students Hear to\r\n Reach Hig...
102120                     Learning In The 21st Century
90456                                          Wiggle Room
72968   Support Computer Science and Computer Graphics...
83322                       Growing Independent Readers
62551                                   Making Math Fun!
72622                    Student Access to Missed Lessons
108171                         College Signing Day Event
28405                        3rd Grade Flexible Seating
93432                                  Learning in Color
81838                 Nanakuli Football Projection Screen
```

```
68314                              Operation  Organization
40730   Bringing Agriculture and Sustainability to the...
77397                      Cricket Cutting Machine Needed
11845                                         News for Kids

                                          project_essay_1  \
86221   I have been fortunate enough to use the Fairy ...
18308   Imagine being 8-9 years old. You're in your th...
79692   Having a class of 24 students comes with diver...
98588   I recently read an article about giving studen...
57724   My students crave challenge, they eat obstacle...
41908   It's the end of the school year. Routines have...
65282   \"Sitting still is overrated. It makes sense f...
76584   It’s not enough to read a book and write an es...
36938   Never has society so rapidly changed. Technolo...
20668   Do you remember the first time you saw Star Wa...
41786   My students yearn for a classroom environment ...
50434   Media and cinematography has been an extremely...
38495   Computer coding and robotics, my second grader...
100376  I teach 4th grade math, writing, social studie...
85477   In my classroom we explore and delve into real...
108373  A typical day in my classroom starts 30 minute...
54927   We LOVE technology! In our classroom, technolo...
83930   \"Teachers who love teaching teach children to...
85415   Do you remember the book you read that made yo...
61734   My students are learning to become lovers of r...
94408   Throughout this school year, I hope to enable ...
24494   Children spend much too much time connected to...
26045   Education is about nurturing justice, engaging...
9844    I teach six amazing children with autism. Each...
82550   Everyday my students interact with each other ...
58243   Everyday students are so excited to come to me...
38778   I am half day pre-k. I have two sets of studen...
57338   Our school is an urban public school that serv...
17757   Each day I teach a class from every grade. I'm...
77852   Today we're kindergartners at a STEM school, t...
...                                                     ...
4966    I work in a school of approximately 800 studen...
77875   As a new teacher working in a high poverty dis...
```

```
39452   Love to sing, create, move and learn? You've c...
47553   Our school is 95 percent free and reduced lunc...
40182   My students live in Oklahoma City.  Oklahoma g...
49169   My 6th period class consist of 36 students wit...
58987   I work with a wonderful group of second grader...
33573   As a teacher-librarian, I get to share my love...
15399   I am so incredibly lucky to spend my days with...
68038   I teach fifth graders in a low income, high po...
16178   My students are from low income families aroun...
90442   \"Every child deserves a champion: an adult wh...
9114    Motivated to learn, my students never cease to...
62614   Our school encountered a great loss due to a d...
31942   Although physical education is mandated for on...
3203    My students are excited, happy, frustrated, sa...
102120  My first graders are creative, innovative, and...
90456   My classroom is a revolving door. They are eag...
72968   My school will work with Microsoft's TEALS pro...
83322   Each day my students eagerly anticipate our re...
62551   I teach 17 amazing students in a Title One sch...
72622   My students often have to worry about things o...
108171  Our students come from multiple different back...
28405   My students this year love science and enginee...
93432   I teach first grade in a Title I school. Altho...
81838   Our day starts with about 100 students athlete...
68314   My students range from age four to five years ...
40730   We are a Title 1 school  650 total students. O...
77397   I teach many different types of students.  My ...
11845   My first graders are eager to learn about the ...

                                          project_essay_2  \
86221   My students come from a variety of backgrounds...
18308   Most of my students have autism, anxiety, anot...
79692   I have a class of twenty-four kindergarten stu...
98588   I teach at a low-income (Title 1) school. Ever...
57724   We are an urban, public k-5 elementary school....
41908   My students desire challenges, movement, and c...
65282   We are a room full of bright, curious, innovat...
76584   My students come from diverse family situation...
36938   Our Language Arts and Social Justice Magnet Sc...
```

```
20668   I am a teacher/librarian at an elementary scho...
41786   I have the privilege of teaching an incredible...
50434   My students in media are mostly junior and sen...
38495   We are a small, rural school in Northwest Ohio...
100376  My 4th grade students are excited learners and...
85477   My students are varied and fabulous. I teach k...
108373  Welcome to my class, where laughter and learni...
54927   I have 50 bright, hard-working, technology-mot...
83930   Each day, my students enter my classroom with ...
85415   My students are part of an exciting adventure....
61734   My students are freshman in the Honors English...
94408   My school is a Pre-K through 8 school with tea...
24494   I teach in a wonderful pre-K through 5th grade...
26045   Students from all over the city attend seeking...
9844    My students are determined to overcome challen...
82550   My students are a diverse group whose learning...
58243   They are all students who speak two languages ...
38778   My students range from 4 to 5 years in age. Th...
57338   I teach in a neighborhood school that serves 4...
17757   These children living in the heart of the Miss...
77852   My kindergarten classroom is made up of divers...
...                                                   ...
4966    By having flexible seating options in my class...
77875   My second grade students are new, eager reader...
39452   Class meetings on the carpet are currently a s...
47553   According the the ASCD \"Slowing Summer Slide\...
40182   My district is facing even MORE budget cuts!  ...
49169   My students will be creating abstract art as a...
58987   A learning environment should be clean and inv...
33573   My students deserve to see current, relevant, ...
15399   With an increase in understanding of student g...
68038   Most students have little or no experience wit...
16178   I am excited to say that the requested materia...
90442   Empowering students to read in a variety of wa...
9114    Research has proven the correlation between pl...
62614   With the new headphones, the students will be ...
31942   My students will be the first generation to ha...
3203    The 30 Panasonic Headphones will replace age o...
102120  With the addition of a new iPad and case, my s...
```

```
90456   The items we are requesting will help our stud...
72968   The class needs color cartridges for the color...
83322   Each day, the kids are always so eager to begi...
62551   I teach an amazing class of second grade schol...
72622   In our school we have a specialist program of ...
108171  Our students have worked tirelessly for the pa...
28405   The variety of materials that I have selected ...
93432   \"Love of beauty is taste. The creation of bea...
81838   Lack of equipment prohibits proper training an...
68314   Being a first year teacher I have had to spend...
40730   These computers will populate the computer lab...
77397   This cricket cutting machine will be used for ...
11845   I have used these magazines in the past, and k...

                                          project_essay_3  \
86221   Each month I try to do several science or STEM...
18308   It is tough to do more than one thing at a tim...
79692   By having a mobile listening and storage cente...
98588   We need a classroom rug that we can use as a c...
57724   With the new common core standards that have b...
41908   I will design different clues using specific c...
65282   We will use this new and modern furniture to h...
76584   Igniting imagination, problem solving and coop...
36938   \"Is it my turn, Ms. K? When am I going to be ...
20668   My students need a Dash and Dot robot duo to e...
41786   Ideally, I would love to delve right into \"fl...
50434   The materials that I am requesting will be use...
38495   We have a new technology teacher to our distri...
100376  Very few students have materials at home to he...
85477   My students work through various modules in my...
108373  We are lucky to be a one-to-one technology sch...
54927   The ability to place one virtual reality devic...
83930   The students will use the iPads in my classroo...
85415   We are fortunate enough to be a \"project scho...
61734   They are going to use these novel EVERYDAY!  M...
94408   The shopping carts would be a wonderful additi...
24494   Last year, our school implemented a once per w...
26045   I plan on using both of these texts to help st...
9844    My students need an iPad mini, a Kindle, cases...
```

```
82550   My students love to learn and work as teams.  ...
58243   With these materials, I will be able to implem...
38778   My students need; What's the Rhyme, Building W...
57338   Literacy developed in the primary language tra...
17757   As the students learn about photography and el...
77852   I am requesting writing journals, crayons and ...
...                                                     ...
4966                                                    NaN
77875                                                   NaN
39452                                                   NaN
47553                                                   NaN
40182                                                   NaN
49169                                                   NaN
58987                                                   NaN
33573                                                   NaN
15399                                                   NaN
68038                                                   NaN
16178                                                   NaN
90442                                                   NaN
9114                                                    NaN
62614                                                   NaN
31942                                                   NaN
3203                                                    NaN
102120                                                  NaN
90456                                                   NaN
72968                                                   NaN
83322                                                   NaN
62551                                                   NaN
72622                                                   NaN
108171                                                  NaN
28405                                                   NaN
93432                                                   NaN
81838                                                   NaN
68314                                                   NaN
40730                                                   NaN
77397                                                   NaN
11845                                                   NaN

                                                 project_essay_4  \
```

```
86221   It is challenging to develop high quality scie...
18308   When my students are able to calm themselves d...
79692   A mobile listening center will help keep equip...
98588   Benjamin Franklin once said, \"Tell me and I f...
57724   These remarkable gifts will provide students w...
41908   Donations to this project will immediately imp...
65282   Getting and keeping students engaged is perhap...
76584   21st Century teaching is more than technology;...
36938   By donating to this project, you will give my ...
20668   These robots incorporate the STEM subject area...
41786   This project will be so beneficial for my stud...
50434   My project will make a difference. My project ...
38495   You can help this dream of ours become a reali...
100376  This project will change the lives of my stude...
85477   My students find themselves on even footing in...
108373  If my students have good quality headphones wi...
54927   By allowing my students the opportunity to tak...
83930   My students struggle to read and technology mo...
85415   As a new teacher my book shelves are bare or t...
61734   These donations will make a massive difference...
94408   Donations to this project will allow me to enc...
24494   By having these materials available to my stud...
26045   Although literature often reflects the themes ...
9844    My students love to learn and they need to be ...
82550   With the addition of two iPads to my classroom...
58243   The donations will help improve my classroom b...
38778   Many of my students come from poverty or are E...
57338   Free voluntary reading helps all components of...
17757   My 5th grade students are begging for a way to...
77852   With your donation to my \"STEM Kindergartners...
...                                                    ...
4966                                                   NaN
77875                                                  NaN
39452                                                  NaN
47553                                                  NaN
40182                                                  NaN
49169                                                  NaN
58987                                                  NaN
33573                                                  NaN
```

```
15399                                                    NaN
68038                                                    NaN
16178                                                    NaN
90442                                                    NaN
9114                                                     NaN
62614                                                    NaN
31942                                                    NaN
3203                                                     NaN
102120                                                   NaN
90456                                                    NaN
72968                                                    NaN
83322                                                    NaN
62551                                                    NaN
72622                                                    NaN
108171                                                   NaN
28405                                                    NaN
93432                                                    NaN
81838                                                    NaN
68314                                                    NaN
40730                                                    NaN
77397                                                    NaN
11845                                                    NaN

                                 project_resource_summary  \
86221   My students need STEM kits to learn critical s...
18308   My students need Boogie Boards for quiet senso...
79692   My students need a mobile listening center to ...
98588   My students need flexible seating in the class...
57724   My students need copies of the New York Times ...
41908   My students need items from a \"Breakout Box\"...
65282   My students need flexible seating options that...
76584   My students need an iPad Mini/accessories to p...
36938                       My students need 1 ipad mini.
20668   My students need a Dash & Dot Pack and Wonder ...
41786   My students need 5 Hokki Stools and an easel o...
50434   My students need a film capable drone and GoPr...
38495   My students need a robot mouse STEM activity s...
100376  My students need a real-world cross-curricular...
85477   My students need various books, kits, and game...
```

```
108373  My students need headphones with microphones t...
54927   My students need virtual reality viewers to ho...
83930   My students need an iPad and Osmo learning sys...
85415   My students need books to help support their h...
61734   My students need YA novels to improve their lo...
94408   My students need shopping carts, a parachute, ...
24494           My students need many rolls of duct tape.
26045   My students need 30 copies of Rudolfo Anaya's ...
9844    My students need an iPad mini, headphones for ...
82550   My students need an iPad mini with a protectiv...
58243   My students need highlighters, magnetic letter...
38778   My students need; What's the Rhyme, Building W...
57338   My students need more bilingual and Spanish bo...
17757   My students need 3 cameras with accessories fo...
77852   My students need Draw and Write journals, cray...
...                                                   ...
4966    My students need wobble chairs in order to hav...
77875   My students need various supplies including co...
39452   My students need A Place for Everyone Calming ...
47553   My students need Science, Reading, and Math Jo...
40182   My students need more tools to help them learn...
49169   My students need tactile resistant bands for t...
58987   My students need a welcoming, comfortable, and...
33573   My students need award-winning books in their ...
15399   My students need a choice in comfortable seati...
68038   My students need graphic books to help them en...
16178   My students need age appropriate fidgets, soci...
90442   My students need 2 IPads to power up their rea...
9114    My students need chess sets in order to learn ...
62614   My students need headphones to access the stud...
31942   My students need cones, foam balls,  bean bags...
3203    My students need 30 Panasonic On-Ear Stereo He...
102120  My students need an iPad to become proficient ...
90456   My students need sensory items to help them ca...
72968   My students need 4 color printer cartridges fo...
83322   My students need high interest books that they...
62551   My students need a measurement center, additio...
72622   My students need access to recordings of class...
108171  My students need dorm room essentials to creat...
```

```
28405   My students need flexible seating choices to m...
93432   My students need ink and colorful paper to bri...
81838   My students need a Projection Screen to use fo...
68314   My students need organization supplies in orde...
40730   My students need 5 Windows Laptops so students...
77397   My students need a cricket machine to help mak...
11845   My students need a subscription to Scholastic ...

        teacher_number_of_previously_posted_projects  project_is_approv
ed
86221                                                53
1
18308                                                 4
1
79692                                                10
1
98588                                                 2
1
57724                                                 2
1
41908                                                 6
1
65282                                                 0
1
76584                                                 0
0
36938                                               127
1
20668                                                41
1
41786                                                 1
1
50434                                                 0
1
38495                                                72
1
100376                                                1
1
85477                                                 6
```

```
1
108373                                                  4
1
54927                                                   2
0
83930                                                   0
1
85415                                                   6
1
61734                                                   3
1
94408                                                   3
1
24494                                                 166
1
26045                                                  10
1
9844                                                    7
1
82550                                                   0
1
58243                                                   2
0
38778                                                   1
1
57338                                                   0
1
17757                                                   4
1
77852                                                  58
1
...                                                   ...
...
4966                                                    0
1
77875                                                   0
1
39452                                                  18
1
```

```
47553                                                   64
1
40182                                                    1
1
49169                                                    1
1
58987                                                    7
1
33573                                                    9
1
15399                                                    0
1
68038                                                   31
1
16178                                                   12
1
90442                                                   10
1
9114                                                     0
1
62614                                                    7
1
31942                                                    4
1
3203                                                     1
1
102120                                                  71
1
90456                                                   20
1
72968                                                    1
1
83322                                                    2
1
62551                                                   24
1
72622                                                    1
1
108171                                                   0
```

```
1
28405                                                  1
1
93432                                                 51
1
81838                                                  5
1
68314                                                  3
1
40730                                                  0
1
77397                                                  1
1
11845                                                  2
1

[109248 rows x 17 columns]>
```

In [6]:

```python
# 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 [7]:

```python
# https://gist.github.com/sebleier/554280
```

```
# we are removing the words from the stop words list: 'no', 'nor', 'no
t'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves'
, 'you', "you're", "you've",\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselve
s', 'he', 'him', 'his', 'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'it
s', 'itself', 'they', 'them', 'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'th
is', 'that', "that'll", 'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'h
ave', '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', 'h
ow', 'all', 'any', 'both', 'each', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 's
o', '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', "is
n'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"]
```

1.2 preprocessing of `project_subject_categories`

In [8]:
```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stacko
```

```
verflow.com/a/47301924/4084039

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

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

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

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

In [9]:

```
sorted_cat_dict
```

Out[9]:
```
{'Warmth': 1388,
 'Care_Hunger': 1388,
 'History_Civics': 5914,
 'Music_Arts': 10293,
 'AppliedLearning': 12135,
 'SpecialNeeds': 13642,
 'Health_Sports': 14223,
 'Math_Science': 41421,
 'Literacy_Language': 52239}
```

1.3 preprocessing of `project_subject_subcategories`

In [10]:
```
sub_catogories = list(project_data['project_subject_subcategories'].val
ues)
# remove special characters from list of strings python: https://stacko
verflow.com/a/47301924/4084039

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

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

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

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

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

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

1.3 Text preprocessing

In [11]:
```
# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) +\
                        project_data["project_essay_2"].map(str) + \
                        project_data["project_essay_3"].map(str) + \
                        project_data["project_essay_4"].map(str)
```

In [12]:
```
project_data.head(2)
```

Out[12]:

Unnamed: 0	id	teacher_id	teacher_prefix	school_state	D

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	D
86221	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs.	CA	20 04 00:27
18308	37728	p043609	3f60494c61921b3b43ab61bdde2904df	Ms.	UT	20 04 00:31

In [13]:
```
# printing some random reviews
print(project_data['essay'].values[0])
```

```
I have been fortunate enough to use the Fairy Tale STEM kits in my clas
sroom as well as the STEM journals, which my students really enjoyed.
I would love to implement more of the Lakeshore STEM kits in my classro
om for the next school year as they provide excellent and engaging STEM
lessons.My students come from a variety of backgrounds, including langu
age and socioeconomic status.  Many of them don't have a lot of experie
nce in science and engineering and these kits give me the materials to
provide these exciting opportunities for my students.Each month I try t
o do several science or STEM/STEAM projects.  I would use the kits and
robot to help guide my science instruction in engaging and meaningful w
ays.  I can adapt the kits to my current language arts pacing guide whe
re we already teach some of the material in the kits like tall tales (P
aul Bunyan) or Johnny Appleseed.  The following units will be taught in
the next school year where I will implement these kits: magnets, motio
n, sink vs. float, robots.  I often get to these units and don't know I
f I am teaching the right way or using the right materials.    The kits
will give me additional ideas, strategies, and lessons to prepare my st
udents in science.It is challenging to develop high quality science act
ivities.  These kits give me the materials I need to provide my student
s with science activities that will go along with the curriculum in my
```

```
classroom.  Although I have some things (like magnets) in my classroom,
I don't know how to use them effectively.  The kits will provide me wit
h the right amount of materials and show me how to use them in an appro
priate way.
```

In [14]:

```
from tqdm import tqdm
preprocessed_essays = []
len_essay=[]
# tqdm is for printing the status bar
for sentance1 in tqdm(project_data['essay'].values):
    sent= sentance1.lower()
    sent = decontracted(sent)
    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 stopwor
ds)
    preprocessed_essays.append(sent.lower().strip())
    len_essay.append(len(sent.split()))
num_essay=np.array(len_essay)
```

```
100%|[redacted]| 109248/109248 [01:34<00:00, 115
5.29it/s]
```

In [15]:

```
# after preprocesing
project_data['essay']=preprocessed_essays
project_data['num_essay']=num_essay
project_data.drop(['project_essay_1'], axis=1, inplace=True)
project_data.drop(['project_essay_2'], axis=1, inplace=True)
project_data.drop(['project_essay_3'], axis=1, inplace=True)
project_data.drop(['project_essay_4'], axis=1, inplace=True)
print(project_data['essay'].values[0])
```

```
fortunate enough use fairy tale stem kits classroom well stem journals
students really enjoyed would love implement lakeshore stem kits classr
oom next school year provide excellent engaging stem lessons students c
```

```
ome variety backgrounds including language socioeconomic status many no
t lot experience science engineering kits give materials provide exciti
ng opportunities students month try several science stem steam projects
would use kits robot help guide science instruction engaging meaningful
ways adapt kits current language arts pacing guide already teach materi
al kits like tall tales paul bunyan johnny appleseed following units ta
ught next school year implement kits magnets motion sink vs float robot
s often get units not know teaching right way using right materials kit
s give additional ideas strategies lessons prepare students science cha
llenging develop high quality science activities kits give materials ne
ed provide students science activities go along curriculum classroom al
though things like magnets classroom not know use effectively kits prov
ide right amount materials show use appropriate way
```

1.4 Preprocessing of `project_title`

In [16]:

```
# Combining all the above statemennts
from tqdm import tqdm
preprocessed_titles = []
len_project=[]
# tqdm is for printing the status bar
for sentence2 in tqdm(project_data['project_title'].values):
    sent = sentence2.lower()
    sent = decontracted(sent)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    # https://gist.github.com/sebleier/554280
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    preprocessed_titles.append(sent.lower().strip())
    len_project.append(len(sent.split()))
num_project=np.array(len_project)
```

```
100%|██████████████████████████████| 109248/109248 [00:04<00:00, 2720
2.39it/s]
```

In [17]:
```
# after preprocesing
project_data['project_title']=preprocessed_titles
project_data['num_project']=num_project
print(project_data['project_title'][0])
```

```
not 21st century learners across ocean
```

In [18]:
```
#Preprocessing the project_grade_category
project_grade_category_cleaned=[]
for grade in tqdm(project_data['project_grade_category'].values):
    grade = grade.replace(' ', '_')
    grade = grade.replace('-', '_')
    project_grade_category_cleaned.append(grade)
project_data['Project_grade_category']=project_grade_category_cleaned
```

```
100%|[redacted]| 109248/109248 [00:00<00:00, 74786
8.98it/s]
```

In [19]:
```
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity'
:'sum'}).reset_index()
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

Assignment 11: TruncatedSVD

- step 1 Select the top 2k words from essay text and project_title (concatinate essay text with project title and then find the top 2k words) based on their `idf_` values
- step 2 Compute the co-occurance matrix with these 2k words, with window size=5 (ref)
- step 3 Use TruncatedSVD on calculated co-occurance matrix and reduce its dimensions, choose the number of components (`n_components`) using elbow method

> - The shape of the matrix after TruncatedSVD will be 2000*n, i.e. each row represents a vector form of the corresponding word.

- Vectorize the essay text and project titles using these word vectors. (while vectorizing, do ignore all the words which are not in top 2k words)

- step 4 Concatenate these truncatedSVD matrix, with the matrix with features
  - **school_state** : categorical data
  - **clean_categories** : categorical data
  - **clean_subcategories** : categorical data
  - **project_grade_category** :categorical data
  - **teacher_prefix** : categorical data
  - **quantity** : numerical data
  - **teacher_number_of_previously_posted_projects** : numerical data
  - **price** : numerical data
  - **sentiment score's of each of the essay** : numerical data
  - **number of words in the title** : numerical data
  - **number of words in the combine essays** : numerical data
  - **word vectors calculated in** step 3 : numerical data
- step 5: Apply GBDT on matrix that was formed in step 4 of this assignment, **DO REFER THIS BLOG: XGBOOST DMATRIX**
- **step 6:Hyper parameter tuning (Consider any two hyper parameters)**
  - **Find the best hyper parameter which will give the maximum AUC value**
  - **Find the best hyper paramter using k-fold cross validation or simple cross validation data**
  - **Use gridsearch cv or randomsearch cv or you can also write your own for loops to do this task of hyperparameter tuning**

Note: Data Leakage

1. **There will be an issue of data-leakage if you vectorize the entire data and then split it into train/cv/test.**
2. **To avoid the issue of data-leakag, make sure to split your data first and then vectorize it.**

3. While vectorizing your data, apply the method fit_transform() on you train data, and apply the method transform() on cv/test data.
4. For more details please go through this link.

In [20]:
```
project_data=project_data.sample(n=50000)
```

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

In [21]:
```
from sklearn.model_selection import train_test_split
# split the data set into train and test respectively 80% and 20%
y=project_data['project_is_approved']
project_data.drop(['project_is_approved'],axis=1, inplace=True)
x=project_data
X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.20,rando
m_state=1)
print("Shape of Train data set X={} Y={}".format(X_train.shape,Y_train.
shape))
print("Shape of Test data set X={} Y={}".format(X_test.shape,Y_test.sha
pe))
```

```
Shape of Train data set X=(40000, 18) Y=(40000,)
Shape of Test data set X=(10000, 18) Y=(10000,)
```

2.2 Make Data Model Ready: encoding numerical, categorical features

1.5.1 Vectorizing Categorical data

In [25]:
```
# we use count vectorizer to convert the values into one hot encoded fe
atures
# Project categories
```

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_categories = CountVectorizer(vocabulary=list(sorted_cat_dict
.keys()),lowercase=False, binary=True)

tr_categories_one_hot=vectorizer_categories.fit_transform(X_train['clea
n_categories'].values)
print(vectorizer_categories.get_feature_names())

te_categories_one_hot =vectorizer_categories.transform(X_test['clean_ca
tegories'].values)

print(tr_categories_one_hot.toarray()[0:1])
print("\nShape of matrix after one hot encodig for 'Project categories'
\nTrain data-{},\nTest data-{}".format(tr_categories_one_hot.shape,te_c
ategories_one_hot.shape))
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearn
ing', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Langua
ge']
[[0 0 0 1 0 0 0 0 0]]

Shape of matrix after one hot encodig for 'Project categories'
Train data-(40000, 9),
Test data-(10000, 9)
```

In [26]:

```
# we use count vectorizer to convert the values into one hot encoded fe
atures
# Project subcategories
vectorizer_subcategories = CountVectorizer(vocabulary=list(sorted_sub_c
at_dict.keys()), lowercase=False, binary=True)

tr_sub_categories_one_hot=vectorizer_subcategories.fit_transform(X_trai
n['clean_subcategories'].values)
print(vectorizer_subcategories.get_feature_names())

te_sub_categories_one_hot = vectorizer_subcategories.transform(X_test[
'clean_subcategories'].values)

print(tr_sub_categories_one_hot.toarray()[0:2])
```

```
print("\nShape of matrix after one hot encodig for 'Project sub categor
ies'\nTrain data-{},\nTest data-{}".format(tr_sub_categories_one_hot.sh
ape,te_sub_categories_one_hot.shape))
```

```
['Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolveme
nt', 'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'Nutri
tionEducation', 'Warmth', 'Care_Hunger', 'SocialSciences', 'PerformingA
rts', 'CharacterEducation', 'TeamSports', 'Other', 'College_CareerPre
p', 'Music', 'History_Geography', 'Health_LifeScience', 'EarlyDevelopme
nt', 'ESL', 'Gym_Fitness', 'EnvironmentalScience', 'VisualArts', 'Healt
h_Wellness', 'AppliedSciences', 'SpecialNeeds', 'Literature_Writing',
'Mathematics', 'Literacy']
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0]]

Shape of matrix after one hot encodig for 'Project sub categories'
Train data-(40000, 30),
Test data-(10000, 30)
```

In [27]:

```
# you can do the similar thing with state, teacher_prefix and project_g
rade_category also
# we use count vectorizer to convert the values into one hot encoded fe
atures
#teacher_prefix
vectorizer_teacher_prefix = CountVectorizer(lowercase=False, binary=Tru
e)
tr_teacher_prefix_one_hot=vectorizer_teacher_prefix.fit_transform(X_tra
in['teacher_prefix'].values.astype('str'))
print(vectorizer_teacher_prefix.get_feature_names())


te_teacher_prefix_one_hot = vectorizer_teacher_prefix.transform(X_test[
'teacher_prefix'].values.astype('str'))

print(tr_teacher_prefix_one_hot.toarray()[0:1])
print("\nShape of matrix after one hot encodig for 'teacher_prefix'\nTr
ain data-{},\nTest data-{}".format(tr_teacher_prefix_one_hot.shape,te_t
eacher_prefix_one_hot.shape))
```

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

```
[[0 0 1 0 0]]

Shape of matrix after one hot encodig for 'teacher_prefix'

Train data-(40000, 5),
Test data-(10000, 5)
```

In [28]:
```
# we use count vectorizer to convert the values into one hot encoded fe
atures
#school_state
vectorizer_school_state = CountVectorizer(lowercase=False, binary=True)
tr_school_state_one_hot=vectorizer_school_state.fit_transform(X_train[
'school_state'].values.astype('str'))
print(vectorizer_school_state.get_feature_names())


te_school_state_one_hot = vectorizer_school_state.transform(X_test['sch
ool_state'].values.astype('str'))

print(tr_school_state_one_hot.toarray()[0:1])
print("\nShape of matrix after one hot encodig for 'teacher_prefix'\nTr
ain data-{},\nTest data-{}".format(tr_school_state_one_hot.shape,te_sch
ool_state_one_hot.shape))
```

```
['AK', 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA', 'H
I', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA', 'MD', 'ME', 'MI',
'MN', 'MO', 'MS', 'MT', 'NC', 'ND', 'NE', 'NH', 'NJ', 'NM', 'NV', 'NY',
'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VA', 'VT',
'WA', 'WI', 'WV', 'WY']
[[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0]]

Shape of matrix after one hot encodig for 'teacher_prefix'
Train data-(40000, 51),
Test data-(10000, 51)
```

In [29]:
```
# we use count vectorizer to convert the values into one hot encoded fe
atures
```

```
#project_grade_category
vectorizer_grade_category = CountVectorizer(lowercase=False, binary=Tru
e)
tr_grade_category_one_hot=vectorizer_grade_category.fit_transform(X_tra
in['Project_grade_category'])
print(vectorizer_grade_category.get_feature_names())


te_grade_category_one_hot = vectorizer_grade_category.transform(X_test[
'Project_grade_category'])

print(tr_grade_category_one_hot.toarray()[0:1])

print(te_grade_category_one_hot.toarray()[0:1])
print("\nShape of matrix after one hot encodig for 'project_grade_categ
ory'\nTrain data-{},\nTest data-{}".format(tr_grade_category_one_hot.sh
ape,te_grade_category_one_hot.shape))
```

```
['Grades_3_5', 'Grades_6_8', 'Grades_9_12', 'Grades_PreK_2']
[[1 0 0 0]]
[[0 0 0 1]]

Shape of matrix after one hot encodig for 'project_grade_category'
Train data-(40000, 4),
Test data-(10000, 4)
```

1.5.2 standardizing Numerical features

In [30]:

```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-learn.org/stable/modules/gene
rated/sklearn.preprocessing.StandardScaler.html
from sklearn.preprocessing import StandardScaler

# price_standardized = standardScalar.fit(X_train['price'].values)
# this will rise the error
# ValueError: Expected 2D array, got 1D array instead: array=[725.05 21
3.03 329.   ... 399.   287.73   5.5 ].
# Reshape your data either using array.reshape(-1, 1)
```

```
price_scalar = StandardScaler()
tr_price_standardized=price_scalar.fit_transform(X_train['price'].value
s.reshape(-1,1)) # finding the mean and standard deviation of this data
print(f"Mean : {price_scalar.mean_[0]}, Standard deviation : {np.sqrt(p
rice_scalar.var_[0])}")

# Now standardize the data with above maen and variance.

te_price_standardized = price_scalar.transform(X_test['price'].values.r
eshape(-1, 1))
```

```
Mean : 297.57086675, Standard deviation : 366.98452145935426
```

In [31]:

```
print("\nShape of matrix after column standardization for 'price'\nTrai
n data-{},\nTest data-{}".format(tr_price_standardized.shape,te_price_s
tandardized.shape))
```

```
Shape of matrix after column standardization for 'price'
Train data-(40000, 1),
Test data-(10000, 1)
```

In [32]:

```
#quantity
quantity_scalar = StandardScaler()
tr_quantity_standardized=quantity_scalar.fit_transform(X_train['quantit
y'].values.reshape(-1,1)) # finding the mean and standard deviation of
 this data
print(f"Mean : {quantity_scalar.mean_[0]}, Standard deviation : {np.sqr
t(quantity_scalar.var_[0])}")

# Now standardize the data with above maen and variance.

te_quantity_standardized = quantity_scalar.transform(X_test['quantity']
.values.reshape(-1, 1))
print("\nShape of matrix after column standardization for 'quantity'\nT
rain data-{},\nTest data-{}".format(tr_quantity_standardized.shape,te_q
uantity_standardized.shape))
```

```
Mean : 16.9118, Standard deviation : 26.186896928807737
```

```
Shape of matrix after column standardization for 'quantity'

Train data-(40000, 1),
Test data-(10000, 1)
```

In [33]:
```
#teacher_number_of_previously_posted_projects
teacher_number_of_previously_posted_projects_scalar = StandardScaler()
tr_teacher_number_of_previously_posted_projects_standardized=teacher_nu
mber_of_previously_posted_projects_scalar.fit_transform(X_train['teache
r_number_of_previously_posted_projects'].values.reshape(-1,1)) # findin
g the mean and standard deviation of this data
print(f"Mean : {teacher_number_of_previously_posted_projects_scalar.mea
n_[0]}, Standard deviation : {np.sqrt(teacher_number_of_previously_post
ed_projects_scalar.var_[0])}")

# Now standardize the data with above maen and variance.

te_teacher_number_of_previously_posted_projects_standardized = teacher_
number_of_previously_posted_projects_scalar.transform(X_test['teacher_n
umber_of_previously_posted_projects'].values.reshape(-1, 1))
print("\nShape of matrix after column standardization for 'teacher_numb
er_of_previously_posted_projects'\nTrain data-{},\nTest data-{}".format
(tr_teacher_number_of_previously_posted_projects_standardized.shape,te_
teacher_number_of_previously_posted_projects_standardized.shape))
```

```
Mean : 11.1628, Standard deviation : 27.96335809876918

Shape of matrix after column standardization for 'teacher_number_of_pre
viously_posted_projects'
Train data-(40000, 1),
Test data-(10000, 1)
```

In [34]:
```
#Number of words in essay
num_essay_scalar = StandardScaler()
tr_num_essay_standardized=num_essay_scalar.fit_transform(X_train['num_e
ssay'].values.reshape(-1,1)) # finding the mean and standard deviation
 of this data
print(f"Mean : {num_essay_scalar.mean_[0]}, Standard deviation : {np.sq
rt(num_essay_scalar.var_[0])}")
```

```
# Now standardize the data with above maen and variance.

te_num_essay_standardized = num_essay_scalar.transform(X_test['num_essa
y'].values.reshape(-1, 1))
print("\nShape of matrix after column standardization for 'num_essay'\n
Train data-{},\nTest data-{}".format(tr_num_essay_standardized.shape,te
_num_essay_standardized.shape))
```

```
Mean : 138.30975, Standard deviation : 36.41687802293738

Shape of matrix after column standardization for 'num_essay'
Train data-(40000, 1),
Test data-(10000, 1)
```

In [35]:

```
#Number of words in project title
num_project_scalar = StandardScaler()
tr_num_project_standardized=num_project_scalar.fit_transform(X_train['n
um_project'].values.reshape(-1,1)) # finding the mean and standard devi
ation of this data
print(f"Mean : {num_project_scalar.mean_[0]}, Standard deviation : {np.
sqrt(num_project_scalar.var_[0])}")

# Now standardize the data with above maen and variance.
te_num_project_standardized = num_project_scalar.transform(X_test['num_
project'].values.reshape(-1, 1))
print("\nShape of matrix after column standardization for 'num_project'
\nTrain data-{},\nTest data-{}".format(tr_num_project_standardized.shap
e,te_num_project_standardized.shape))
```

```
Mean : 3.692075, Standard deviation : 1.5271893118978406

Shape of matrix after column standardization for 'num_project'
Train data-(40000, 1),
Test data-(10000, 1)
```

In [36]:

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
def senti(data,col_name):
```

```
    neg=[]
    neu=[]
    pos=[]
    compound=[]
    sid = SentimentIntensityAnalyzer()
    for senti in data[col_name]:
        ss = sid.polarity_scores(senti)
        neg.append(ss['neg'])
        neu.append(ss['neu'])
        pos.append(ss['pos'])
        compound.append(ss['compound'])
    return(neg,pos,compound)
# we can use these 4 things as features/attributes (neg, neu, pos, comp
ound)
# neg: 0.01, neu: 0.745, pos: 0.245, compound: 0.9975
```

```
C:\ProgramData\Anaconda3\lib\site-packages\nltk\twitter\__init__.py:20:
UserWarning: The twython library has not been installed. Some functiona
lity from the twitter package will not be available.
  warnings.warn("The twython library has not been installed. "
```

In [37]:

```
neg_train=[]
pos_train=[]
compound_train=[]
neg_train,pos_train,compound_train=senti(X_train,'essay')
X_train['neg']=neg_train
X_train['pos']=pos_train
X_train['compound']=compound_train
```

In [38]:

```
neg_test=[]
pos_test=[]
compound_test=[]
neg_test,pos_test,compound_test=senti(X_test,'essay')
X_test['neg']=neg_test
X_test['pos']=pos_test
X_test['compound']=compound_test
```

In [40]:

```
#sentiment score neg
```

```
neg_scalar = StandardScaler()
tr_neg_standardized=neg_scalar.fit_transform(X_train['neg'].values.resh
ape(-1,1)) # finding the mean and standard deviation of this data
print(f"Mean : {neg_scalar.mean_[0]}, Standard deviation : {np.sqrt(neg
_scalar.var_[0])}")

# Now standardize the data with above maen and variance.

te_neg_standardized = neg_scalar.transform(X_test['neg'].values.reshape
(-1, 1))
print("\nShape of matrix after column standardization for 'neg'\nTrain
 data-{},\nTest data-{}".format(tr_neg_standardized.shape,te_neg_standa
rdized.shape))
```

```
Mean : 0.04799937500000001, Standard deviation : 0.035721878094654755

Shape of matrix after column standardization for 'neg'
Train data-(40000, 1),
Test data-(10000, 1)
```

In [41]:

```
#sentiment score pos
pos_scalar = StandardScaler()
tr_pos_standardized=pos_scalar.fit_transform(X_train['pos'].values.resh
ape(-1,1)) # finding the mean and standard deviation of this data
print(f"Mean : {pos_scalar.mean_[0]}, Standard deviation : {np.sqrt(pos
_scalar.var_[0])}")

# Now standardize the data with above maen and variance.
te_pos_standardized = pos_scalar.transform(X_test['pos'].values.reshape
(-1, 1))
print("\nShape of matrix after column standardization for 'pos'\nTrain
 data-{},\nTest data-{}".format(tr_pos_standardized.shape,te_pos_standa
rdized.shape))
```

```
Mean : 0.28089825, Standard deviation : 0.07790639926820839

Shape of matrix after column standardization for 'pos'
Train data-(40000, 1),
Test data-(10000, 1)
```

In [42]:
```
#sentiment score compound
compound_scalar = StandardScaler()
tr_compound_standardized=compound_scalar.fit_transform(X_train['compoun
d'].values.reshape(-1,1)) # finding the mean and standard deviation of
 this data
print(f"Mean : {compound_scalar.mean_[0]}, Standard deviation : {np.sqr
t(compound_scalar.var_[0])}")

# Now standardize the data with above maen and variance.

te_compound_standardized = compound_scalar.transform(X_test['compound']
.values.reshape(-1, 1))
print("\nShape of matrix after column standardization for 'compound'\nT
rain data-{},\nTest data-{}".format(tr_compound_standardized.shape,te_c
ompound_standardized.shape))
```

```
Mean : 0.9582628149999999, Standard deviation : 0.15277141821615645

Shape of matrix after column standardization for 'compound'
Train data-(40000, 1),
Test data-(10000, 1)
```

2.1 Selecting top 2000 words from `essay` and `project_title`

In [43]:
```
#merging essays and project title
X_train['text']=X_train['essay']+X_train['project_title']
X_test['text']=X_test['essay']+X_test['project_title']
```

In [48]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer_text = TfidfVectorizer(min_df=10,use_idf=True)
#Fitting train data and transforming  train ,cv and test to get idf val
ues
tr_text_tfidf=tfidf_vectorizer_text.fit_transform(X_train['text'])
te_text_tfidf = tfidf_vectorizer_text.transform(X_test['text'])
```

```
print("Shape of matrix TFIDF Vectorizer on text \nTrain data-{},\nTest
 data-{}".format(tr_text_tfidf.shape,te_text_tfidf.shape))
```

```
Shape of matrix TFIDF Vectorizer on text
Train data-(40000, 11898),
Test data-(10000, 11898)
```

In [49]:
```
#storing idf_values and feature_name
idf_feature=pd.DataFrame(tfidf_vectorizer_text.idf_,columns=['idf_value
s'])
idf_feature['feature_name']=tfidf_vectorizer_text.get_feature_names()
#sorting feature_name based on idf_values
idf_feature.sort_values(by=['idf_values'],ascending=False,inplace=True,
axis=0)
```

In [50]:
```
#idf values and feature name
idf_feature[1995:2000]
```

Out[50]:

	idf_values	feature_name
11644	8.824071	weird
49	8.824071	2006
6955	8.824071	nannanlegos
11628	8.824071	wednesday
10837	8.824071	tiger

2.2 Computing Co-occurance matrix

In [51]:
```
#https://stackoverflow.com/questions/35562789/how-do-i-calculate-a-word
-word-co-occurrence-matrix-with-sklearn
from collections import defaultdict
import pandas as pd
import numpy as np
```

```
def co_occurrence(sentences,vocab,window_size):
    d = defaultdict(int)
    vocab = vocab
    for text in sentences:
        # toknizing the sentance
        text = text.lower().split()
        # iterate over all token
        for i in range(len(text)):
            token = text[i]
            #checking whether token is present in vocabulary
            if token not in vocab:
                continue
            #getting words which are there with in 5 window
            next_token = text[i+1 : i+1+window_size]
            #considering next_token which are present in vocabulary
            next_token=[tok for tok in next_token if tok in vocab]

            for t in next_token:
                key = tuple( sorted([t, token]) )
                #increasing count value for occurrence of particular wo
rd
                d[key] += 1


    # formulate the dictionary into dataframe
    #vocab = sorted(vocab) # sort vocab
    df = pd.DataFrame(data=np.zeros((len(vocab), len(vocab)), dtype=np.
int16),
                      index=vocab,
                      columns=vocab)
    for key, value in d.items():
        df.at[key[0], key[1]] = value
        df.at[key[1], key[0]] = value
        df.at[key[0],key[0]]=0
    return df
```

In [52]:

```
text = ["ABC DEF IJK PQR","PQR KLM OPQ","LMN PQR XYZ ABC DEF PQR ABC"]
df = co_occurrence(text,['abc','pqr','def'],2)
```

```
print("Co-occurrence matrix with window_size=2")
df
```

```
Co-occurrence matrix with window_size=2
```

Out[52]:

	abc	pqr	def
abc	0	3	3
pqr	3	0	2
def	3	2	0

In [53]:

```
text = ["ABC DEF IJK PQR","PQR KLM OPQ","LMN PQR XYZ ABC DEF PQR ABC"]
df = co_occurrence(text,['abc','pqr','def'],5)
print("Co-occurrence matrix with window_size=5")
df
```

```
Co-occurrence matrix with window_size=5
```

Out[53]:

	abc	pqr	def
abc	0	5	3
pqr	5	0	3
def	3	3	0

In [54]:

```
#computing Co-occurrence matrix
text=X_train.essay
df = co_occurrence(text,idf_feature.feature_name[:2000].tolist(),5)
print("Co-occurrence matrix with window_size=5")
```

```
Co-occurrence matrix with window_size=5
```

In [55]:

```
df[:5]
```

Out[55]:

	wholesome	dynamath	duolingo	nannanmiss	dwindle	nannanmentor	fleece	dye

	wholesome	dynamath	duolingo	nannanmiss	dwindle	nannanmentor	fleece	dye
wholesome	0	0	0	0	0	0	0	0
dynamath	0	0	0	0	0	0	0	0
duolingo	0	0	0	0	0	0	0	0
nannanmiss	0	0	0	0	0	0	0	0
dwindle	0	0	0	0	0	0	0	0

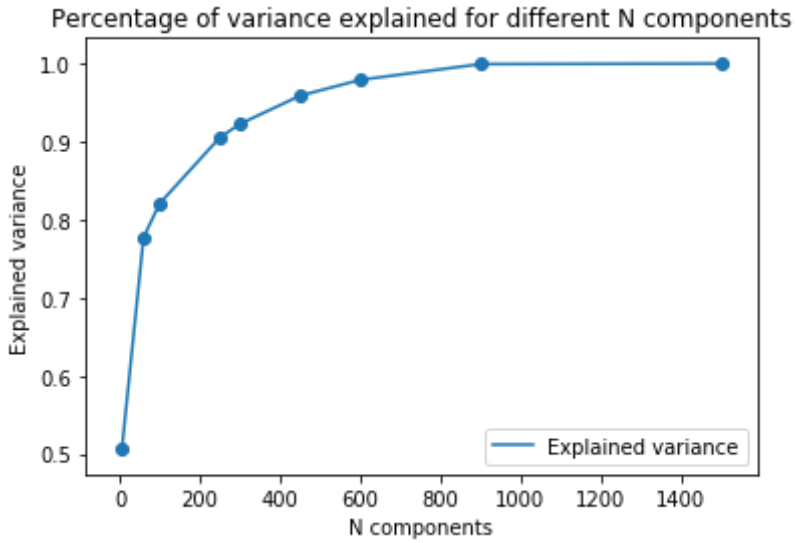
5 rows × 2000 columns

2.3 Applying TruncatedSVD and Calculating Vectors for `essay` and `project_title`

In [56]:
```
from sklearn.decomposition import TruncatedSVD
def dimension_reduction(components):
    explained_variance=[]
    for component in tqdm(components):
        svd = TruncatedSVD(n_components=component, n_iter=7, random_sta
te=42)
        svd.fit(df)
        explained_variance.append(svd.explained_variance_ratio_.sum())
    plt.plot(components,explained_variance,label="Explained variance")
    plt.scatter(components,explained_variance)
    plt.title("Percentage of variance explained for different N compone
nts")
    plt.xlabel("N components")
    plt.ylabel("Explained variance")
    plt.legend()
    plt.show()
```

In [57]:
```
%%time
components=[6,60,100,250,300,450,600,900,1500]
dimension_reduction(components)
```

```
100%|██████████████████████████████████████| 9/9 [00:23<00:00,
2.66s/it]
```




```
Wall time: 24.3 s
```

1. **From the above graph we can see that 95% of variance is explained if n_components=450**

In [58]:
```
svd = TruncatedSVD(n_components=450, n_iter=7)
#getting top 450 dim matrix using truncatedsvd
svd_feature=svd.fit_transform(df)
```

In [59]:
```
from scipy.sparse import hstack,vstack
#storing SVD feature vectors along with feature name
svd_feature_with_word = pd.DataFrame((svd_feature),index=df.index)
svd_feature_with_word.insert(0,"feature_name",idf_feature.feature_name
[:2000].tolist())
print("SVD feature vectors")
svd_feature_with_word[:2]
```

```
SVD feature vectors
```

Out[59]:

	feature_name	0	1	2	3	4	
wholesome	wholesome	-1.875912e-15	-3.503816e-15	1.354070e-14	-1.306480e-14	-3.311039e-17	2.215150e 1
dynamath	dynamath	1.435113e-04	8.321642e-05	7.055589e-01	-7.096124e-01	-1.420106e-03	1.510340e 0

2 rows × 451 columns

In [60]:

```
keys=idf_feature.feature_name[:2000]
values=svd_feature
#dictionary  having key as feature name and values as vector
dict_feat={k:v for (k,v) in zip(keys,values)}
```

In [61]:

```
#function to calculate word vector
def get_final_svd_vector(sentences):
    vector_words=[]
    for text in tqdm(sentences):
        vector_word=np.zeros((450),dtype=float)
        word_count=0
        text = text.lower().split()
        #print(text)
        for i in range(len(text)):
            token = text[i]
            #print(token.)
            if token in dict_feat.keys():
                vector_word=vector_word+dict_feat[token]
                word_count=word_count+1
            else:
                continue
        if word_count!=0:
            vector_word=vector_word/word_count
        vector_words.append(vector_word)
    return(vector_words)
```

In [62]:

```
#computing word_vector for train ,test and cv data
```

```
tr_word_vector=get_final_svd_vector(X_train['text'])
te_word_vector=get_final_svd_vector(X_test['text'])
```

```
100%|██████████████████████████████| 40000/40000 [00:02<00:00, 1367
0.87it/s]
100%|██████████████████████████████| 10000/10000 [00:00<00:00, 1342
5.86it/s]
```

In [63]:
```
tr_word_vector =scipy.sparse.csr_matrix(tr_word_vector)
```

2.4 Merge the features from step 3 and step 4

In [64]:
```
%%time
# merge two sparse matrices: https://stackoverflow.com/a/19710648/40840
39
#categorical, numerical features and word_vector
from scipy.sparse import hstack
tr_word_vector =scipy.sparse.csr_matrix(tr_word_vector)
te_word_vector =scipy.sparse.csr_matrix(te_word_vector)

# with the same hstack function we are concatinating a sparse matrix an
d a dense matirx :)
tr_X_SVD= hstack((tr_school_state_one_hot,tr_categories_one_hot,tr_sub_
categories_one_hot,tr_teacher_prefix_one_hot,tr_grade_category_one_hot,
tr_price_standardized,tr_teacher_number_of_previously_posted_projects_s
tandardized,tr_num_essay_standardized,tr_num_project_standardized,tr_ne
g_standardized,tr_pos_standardized,tr_compound_standardized,tr_word_vec
tor
)).tocsr()
te_X_SVD= hstack((te_school_state_one_hot,te_categories_one_hot,te_sub_
categories_one_hot,te_teacher_prefix_one_hot,te_grade_category_one_hot,
te_price_standardized,te_teacher_number_of_previously_posted_projects_s
tandardized,te_num_essay_standardized,te_num_project_standardized,te_ne
g_standardized,te_pos_standardized,te_compound_standardized,te_word_vec
tor
)).tocsr()
tr_X_SVD=tr_X_SVD.toarray()
```

```
te_X_SVD=te_X_SVD.toarray()
print(tr_X_SVD.shape)
print(te_X_SVD.shape)
```

```
(40000, 556)
(10000, 556)

Wall time: 784 ms
```

2.5 Apply XGBoost on the Final Features from the above section

https://xgboost.readthedocs.io/en/latest/python/python_intro.html

In [65]:
```
#Function to draw 3-d plot
import plotly.graph_objs as go
import plotly.offline as offline
def plot_3d(x1,y1,z1,x2,y2,z2):
    x1=[5, 10, 50, 100, 200,5, 10, 50, 100, 200,5, 10, 50, 100, 200,5,
10, 50, 100, 200,5, 10, 50, 100, 200,5, 10, 50, 100, 200]#estimators
    y1=[2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,6,6,6,6,6,7,7,7,7,7]#ma
x-depth
    trace1 = go.Scatter3d(x=x1,y=y1,z=z1,mode='markers', name = 'train'
)
    trace2 = go.Scatter3d(x=x1,y=y1,z=z2,mode='markers', name = 'Cross
 validation')
    data = [trace1, trace2]
    #print(data)
    layout = go.Layout(scene = dict(xaxis = dict(title='n_estimators'),
yaxis = dict(title='max_depth'),zaxis = dict(title='AUC'),))
    fig = go.Figure(data=data, layout=layout)
    offline.iplot(fig, filename='3d-scatter-colorscale')
```

In [92]:
```
#https://stackoverflow.com/questions/35572000/how-can-i-plot-a-confusio
n-matrix
#Drawing confusion matrix
```

```
def draw_confusion_matrix(clf,threshold,y_true,y_hat,tpr,fpr,t):
    result=[]
    y_pred=[]

    #finding threshold which maximises the tpr and minimises the fpr
    thr=threshold[np.argmax((tpr*(1-fpr)))]

    for probab in y_hat:

        if probab >= thr:
            y_pred.append(1)
        else:
            y_pred.append(0)

    result=confusion_matrix(y_true,y_pred,labels=[0,1])
    df_cm = pd.DataFrame(result,range(2),range(2))
    df_cm.columns = ['Predicted NO','Predicted YES']
    df_cm = df_cm.rename({0: 'Actual NO', 1: 'Actual YES'})
    plt.figure(figsize = (5,3))
    plt.title(t)
    print(df_cm)
    sns.heatmap(df_cm, annot=True,annot_kws={"size": 12}, fmt='g')
```

In [76]:

```
#3 flod cross validation
#Hyper parameter considered are learning_rate,max_depth and n_estimator
s
%%time
import xgboost as xgb
from xgboost.sklearn import XGBClassifier
params={"learning_rate"    : [0.05, 0.10, 0.15, 0.20, 0.02] ,
 "max_depth"        : [ 4, 10, 12, 15,30],
 'n_estimators':[5, 10, 50, 100, 200] }
model = XGBClassifier(objective= 'binary:logistic',seed=27,n_jobs=-1)
gs=GridSearchCV(estimator=model,cv=3,n_jobs=-1,scoring ='roc_auc',verbo
se=True,param_grid=params,return_train_score=True)
gs.fit(tr_word_vector,Y_train)
```

```
Fitting 3 folds for each of 125 candidates, totalling 375 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent work
```

```
ers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed: 21.6min
[Parallel(n_jobs=-1)]: Done 192 tasks      | elapsed: 146.9min
[Parallel(n_jobs=-1)]: Done 375 out of 375 | elapsed: 319.8min finished
```

```
Wall time: 5h 24min 30s
```

In [81]:

```
results=pd.DataFrame(gs.cv_results_).sort_values(by='rank_test_score').
head(56)
results
```

Out[81]:

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_m
104	184.586528	1.324414	0.916352	0.008808	0.02	
52	48.366705	0.276515	0.821803	0.003301	0.15	
28	94.690404	1.035085	0.960158	0.156301	0.1	
27	48.525312	0.335993	0.815288	0.009689	0.1	
3	93.966380	0.439344	0.825813	0.001885	0.05	
16	33.967560	0.258124	0.814105	0.005718	0.05	

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_m
40	18.296691	0.139092	0.819454	0.004714	0.1	
107	111.482266	2.373438	0.858070	0.004083	0.02	
103	93.417718	0.513658	0.866145	0.008643	0.02	
120	33.293198	0.249856	0.865812	0.034977	0.02	
11	28.287537	0.128653	0.806752	0.006193	0.05	
41	34.293377	0.251339	0.832456	0.006652	0.1	
53	94.775082	0.518642	0.847808	0.010967	0.15	
112	130.705122	0.586498	0.881804	0.013892	0.02	
29	187.095422	2.589836	0.923488	0.019690	0.1	

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_n
116	34.351489	0.107688	0.813130	0.009418	0.02	
4	185.845397	1.006443	0.865482	0.010210	0.05	
57	112.282991	0.104198	0.867477	0.011443	0.15	
2	48.035721	0.078914	0.803264	0.006997	0.05	
117	160.750848	1.741394	0.909341	0.010484	0.02	
15	18.177305	0.233932	0.805088	0.008627	0.05	
56	24.259363	0.041070	0.817803	0.010876	0.15	
115	18.343929	0.106641	0.823467	0.004991	0.02	
36	28.339985	0.044078	0.870649	0.077246	0.1	

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_m
51	11.367890	0.041990	0.798799	0.003299	0.15	
108	218.780937	2.160004	0.947443	0.007933	0.02	
102	48.448242	0.438964	0.864730	0.043982	0.02	
91	33.549692	0.300593	0.833803	0.012368	0.2	
58	224.095525	1.836793	0.961730	0.010434	0.15	
61	28.310860	0.406908	0.814097	0.012028	0.15	
54	188.568667	1.221940	0.919155	0.016515	0.15	
76	11.346292	0.067777	0.822947	0.024240	0.2	
35	15.348424	0.121563	0.792671	0.009464	0.1	

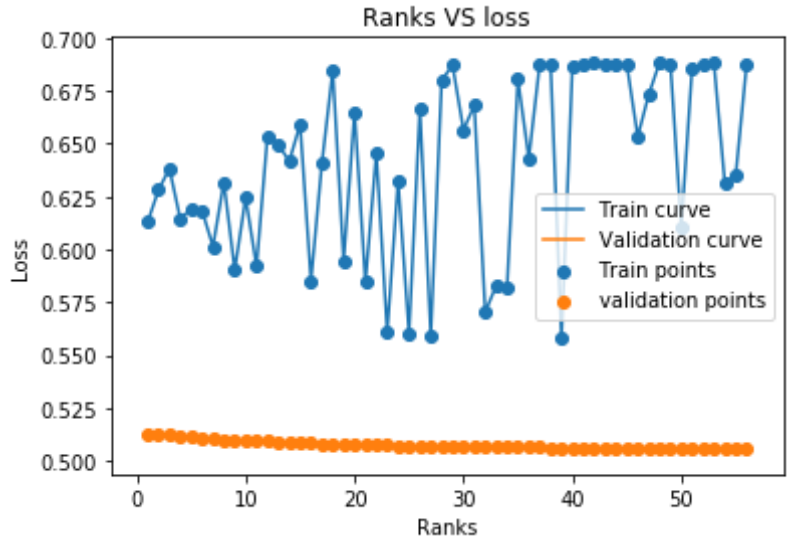
	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_m
6	23.742429	0.046703	0.811140	0.008833	0.05	
109	435.489847	3.385273	1.291869	0.014347	0.02	
121	63.412203	0.269793	0.840139	0.006977	0.02	
59	445.509655	2.891679	1.252820	0.022956	0.15	
63	258.057677	1.220507	0.986813	0.017177	0.15	
10	15.042013	0.089274	0.789947	0.002494	0.05	
9	442.687038	1.079777	1.265176	0.024359	0.05	
64	516.547098	3.670136	1.393936	0.009125	0.15	
73	465.277328	7.154720	1.398936	0.007224	0.15	

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_m
19	596.855580	0.528545	1.663917	0.010783	0.05	
68	303.332138	2.904911	1.078993	0.016802	0.15	
124	935.189753	16.352584	1.948639	0.063737	0.02	
81	24.015563	0.176258	0.807055	0.007312	0.2	
7	110.061989	0.869565	0.874484	0.026404	0.05	
99	938.063337	39.190591	2.101480	0.019469	0.2	
24	894.080565	16.936752	2.053675	0.096197	0.05	
85	15.366763	0.035055	0.822097	0.028087	0.2	
33	222.907580	1.866864	0.969057	0.054508	0.1	

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_learning_rate	param_n
34	445.462195	1.862670	1.252094	0.015609	0.1	
69	611.089819	4.894882	1.587850	0.022104	0.15	
90	18.500954	0.209588	0.860474	0.022070	0.2	
77	49.174315	0.246208	0.821885	0.002012	0.2	
67	149.774018	0.995298	0.915519	0.021995	0.15	

In [82]:

```
x=[x for x in range(1,57)]
y_train_loss=results.mean_train_score.tolist()
y_test_loss=results.mean_test_score.tolist()
#function to plot lines
#plt.xscale('log')
plt.plot(x,y_train_loss,label="Train curve")
plt.plot(x,y_test_loss,label="Validation curve")
plt.scatter(x, y_train_loss, label='Train points')
plt.scatter(x, y_test_loss, label='validation points')
plt.xlabel("Ranks")
plt.ylabel("Loss")
plt.title("Ranks VS loss")
plt.legend()
plt.show()
```



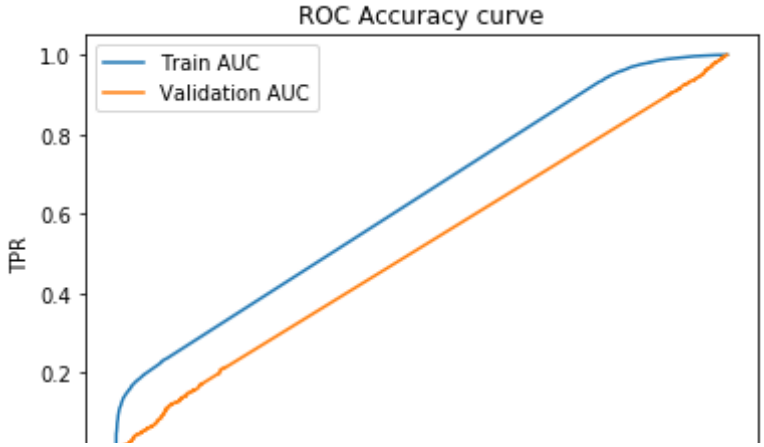
In [88]:
```
results.params[104]
```

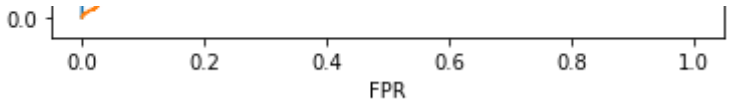
Out[88]:
```
{'learning_rate': 0.02, 'max_depth': 4, 'n_estimators': 200}
```

1. **{'learning_rate': 0.02, 'max_depth': 4, 'n_estimators': 200} This paramaters providing best result compare to other paramters**

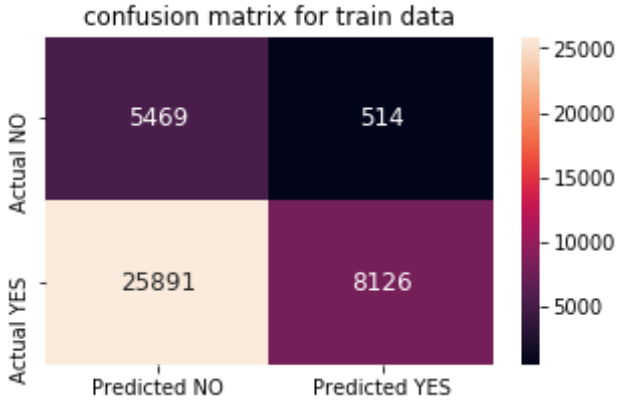
In [93]:
```
%%time
class_w={0:0.5,1:0.5}
#https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/
from sklearn.metrics import roc_curve, auc
#training model with best lambda(λ) paramter
trained_xgb_tfidf_w2v=xgb.XGBClassifier(n_estimators=200,max_depth=4,class_weight=class_w)
#trainning model
trained_xgb_tfidf_w2v.fit(tr_word_vector,Y_train)
# predict the response on the train data
predicted_labels_train=trained_xgb_tfidf_w2v.predict_proba(tr_word_vector)
```

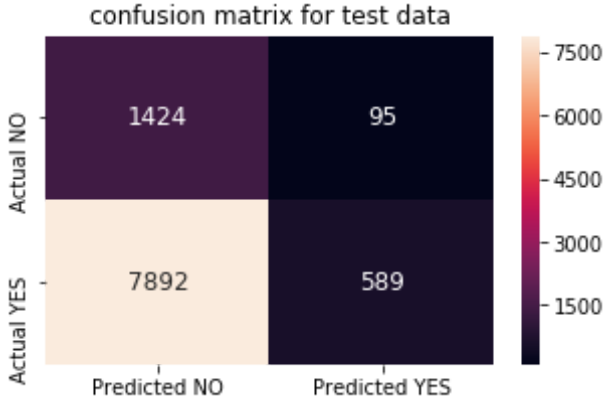
```
predicted_labels_train_temp=trained_xgb_tfidf_w2v.predict(tr_word_vecto
r)
# predict the response on the test data
predicted_labels_test=trained_xgb_tfidf_w2v.predict_proba(te_word_vecto
r)
#Calculating FPR and TPR for train and test data
tr_fpr,tr_tpr,tr_threshold=roc_curve(Y_train,predicted_labels_train[:,1
])
te_fpr,te_tpr,te_threshold=roc_curve(Y_test,predicted_labels_test[:,1])
#drawing ROC ROC Accuracy curve for test and train data
plt.plot(tr_fpr,tr_tpr,label="Train AUC")
plt.plot(te_fpr,te_tpr,label="Validation AUC")
plt.title("ROC Accuracy curve")
plt.xlabel("FPR")
plt.ylabel("TPR")
plt.legend()
plt.show()
print("Train AUC =",round(auc(tr_fpr,tr_tpr),2))
print("Test AUC =",round(auc(te_fpr,te_tpr),2))
#drawing confusion matrix for test and train data
t2="confusion matrix for train data"
draw_confusion_matrix(trained_xgb_tfidf_w2v,tr_threshold,Y_train,predic
ted_labels_train[:,1],tr_tpr,tr_fpr,t2)
t1="confusion matrix for test data"
draw_confusion_matrix(trained_xgb_tfidf_w2v,tr_threshold,Y_test,predict
ed_labels_test[:,1],te_tpr,te_fpr,t1)
```






```
Train AUC = 0.63
Test AUC = 0.51
            Predicted NO  Predicted YES
Actual NO           5469            514
Actual YES         25891           8126
            Predicted NO  Predicted YES
Actual NO           1424             95
Actual YES          7892            589
Wall time: 4min 56s
```





In [94]:
```
from prettytable import PrettyTable
print("XGboost")
table = PrettyTable()

table.field_names = ["XGboost",  "Max Depth","n_estimators", 'learning_
rate','AUC"]

table.add_row(["Train",  "4","200","0.02",0.63])
table.add_row(["Test",  "4","200","0.02",0.51])
print(table)
```

```
XGboost
+---------+-----------+--------------+---------------+------+
| XGboost | Max Depth | n_estimators | learning_rate | AUC  |
+---------+-----------+--------------+---------------+------+
|  Train  |     4     |     200      |      0.02     | 0.63 |
|   Test  |     4     |     200      |      0.02     | 0.51 |
+---------+-----------+--------------+---------------+------+
```