DonorsChoose

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

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

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

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

About the DonorsChoose Data Set

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

Feature	Description
`project_id`	A unique identifier for the proposed project. **Example:** `p036502`
`project_title`	Title of the project. **Examples:** • `Art Will Make You Happy!` • `First Grade Fun`
`project_grade_category`	Grade level of students for which the project is targeted. One of the following enumerated values: • `Grades PreK-2` • `Grades 3-5` • `Grades 6-8` • `Grades 9-12`

Feature	Description
`project_subject_categories`	One or more (comma-separated) subject categories for the project from the following enumerated list of values: • `Applied Learning` • `Care & Hunger` • `Health & Sports` • `History & Civics` • `Literacy & Language` • `Math & Science` • `Music & The Arts` • `Special Needs` • `Warmth` **Examples:** • `Music & The Arts` • `Literacy & Language, Math & Science`
`school_state`	State where school is located (Two-letter U.S. postal code). **Example:** `WY`
`project_subject_subcategories`	One or more (comma-separated) subject subcategories for the project. **Examples:** • `Literacy` • `Literature & Writing, Social Sciences`
`project_resource_summary`	An explanation of the resources needed for the project. **Example:** • `My students need hands on literacy materials to manage sensory needs!`
`project_essay_1`	First application essay*
`project_essay_2`	Second application essay*
`project_essay_3`	Third application essay*
`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`

Feature	Description
`teacher_prefix`	Teacher's title. One of the following enumerated values: • `nan` • `Dr.` • `Mr.` • `Mrs.` • `Ms.` • `Teacher.`
`teacher_number_of_previously_posted_projects`	Number of project applications previously submitted by the same teacher. **Example:** `2`

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

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

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

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

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

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

Notes on the Essay Data

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

- __project_essay_1:__ "Introduce us to your classroom"
- __project_essay_2:__ "Tell us more about your students"
- __project_essay_3:__ "Describe how your students will use the materials you're requesting"
- __project_essay_3:__ "Close by sharing why your project will make a difference"

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

- __project_essay_1:__ "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
- __project_essay_2:__ "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

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

In [1]:

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

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

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

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

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

from tqdm import tqdm
import os

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

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_prefix' 'school_state'
 'project_submitted_datetime' 'project_grade_category'
 'project_subject_categories' 'project_subject_subcategories'
 'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects' 'project_is_approved']
```

In [4]:
```
print("Number of data points in train data", resource_data.shape)
print(resource_data.columns.values)
resource_data.head(2)
```

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

Out[4]:

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

1.2 preprocessing of `project_subject_categories`

In [5]:
```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47301924/4084039

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

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

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

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

1.3 preprocessing of `project_subject_subcategories`

In [6]:
```
sub_catogories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47301924/4084039

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

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

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

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

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

1.4 preprocessing of project_grade_category

In [7]:

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

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

prj_grade_cat_list = []
for i in prj_grade_cat:
    for j in i.split(' '): # it will split by space
        j=j.replace('Grades','') # if we have the words "Grades" we are going to replace it with ''(i.e removing 'Grades')
    prj_grade_cat_list.append(j.strip())

project_data['clean_grade'] = prj_grade_cat_list
project_data.drop(['project_grade_category'], axis=1, inplace=True)

# count of all the words in corpus python: https://stackoverflow.com/a/22898595/4084039
my_counter = Counter()
```

```
for word in project_data['clean_grade'].values:
    my_counter.update(word.split())

prj_grade_cat_dict = dict(my_counter)
sorted_prj_grade_cat_dict = dict(sorted(prj_grade_cat_dict.items(), key=lambda kv: kv[1]))

project_data['clean_grade'].values
```

Out[7]:
```
array(['PreK-2', '6-8', '6-8', ..., 'PreK-2', '3-5', '6-8'], dtype=object)
```

1.5 preprocessing of teacher_prefix

In [8]:
```
#tea_pfx_cat = list(project_data['teacher_prefix'].values)
tea_pfx_cat = list(project_data['teacher_prefix'].astype(str).values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47301924/4084039

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

##https://stackoverflow.com/questions/52736900/how-to-solve-the-attribute-error-float-object-has-no-attribute-split-in-pyth
#vectorizer.fit(project_data['teacher_prefix'].astype(str).values)

tea_pfx_cat_list = []
for i in tea_pfx_cat:
    #for j in i.split(' '): # it will split by space
    #j=j.replace('.','') # if we have the words "Grades" we are going to replace it with ''(i.e removing 'Grades')
    i=i.replace('.','') # if we have the words "Grades" we are going to replace it with ''(i.e removing 'Grades')
    i=i.replace('nan','') # if we have the words "Grades" we are going to replace it with ''(i.e removing 'Grades')
    tea_pfx_cat_list.append(i.strip())

project_data['clean_tea_pfx'] = tea_pfx_cat_list
project_data.drop(['teacher_prefix'], axis=1, inplace=True)

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

tea_pfx_cat_dict = dict(my_counter)
sorted_tea_pfx_cat_dict = dict(sorted(tea_pfx_cat_dict.items(), key=lambda kv: kv[1]))

project_data['clean_tea_pfx'].values
```

Out[8]:
```
array(['Mrs', 'Mr', 'Ms', ..., 'Mrs', 'Mrs', 'Ms'], dtype=object)
```

1.6 Text preprocessing

In [9]:
```
# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) +\
                        project_data["project_essay_2"].map(str) + \
                        project_data["project_essay_3"].map(str) + \
                        project_data["project_essay_4"].map(str)
```

In [10]:
```
project_data.head(2)
```

Out[10]:

	Unnamed: 0	id	teacher_id	school_state	project_submitted_datetime	project_title	project_essay_1	project_essay_2	project
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	IN	05-12-2016 13:43	Educational Support for English Learners at Home	My students are English learners that are work...	\"The limits of your language are the limits o...	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	FL	25-10-2016 09:22	Wanted: Projector for Hungry Learners	Our students arrive to our school eager to lea...	The projector we need for our school is very c...	

In [11]:
```
#### 1.4.2.3 Using Pretrained Models: TFIDF weighted W2V
```

In [12]:
```
# printing some random reviews
print(project_data['essay'].values[0])
print("="*50)
print(project_data['essay'].values[150])
print("="*50)
print(project_data['essay'].values[1000])
```

```
print("="*50)
print(project_data['essay'].values[20000])
print("="*50)
print(project_data['essay'].values[99999])
print("="*50)
```

```
My students are English learners that are working on English as their second or third languages. We are a melting pot of refugees,
immigrants, and native-born Americans bringing the gift of language to our school. \r\n\r\n We have over 24 languages represented
in our English Learner program with students at every level of mastery.  We also have over 40 countries represented with the famil
ies within our school.  Each student brings a wealth of knowledge and experiences to us that open our eyes to new cultures, belief
s, and respect.\"The limits of your language are the limits of your world.\"-Ludwig Wittgenstein  Our English learner's have a str
ong support system at home that begs for more resources.  Many times our parents are learning to read and speak English along side
of their children.  Sometimes this creates barriers for parents to be able to help their child learn phonetics, letter recognitio
n, and other reading skills.\r\n\r\nBy providing these dvd's and players, students are able to continue their mastery of the Engli
sh language even if no one at home is able to assist.  All families with students within the Level 1 proficiency status, will be a
offered to be a part of this program.  These educational videos will be specially chosen by the English Learner Teacher and will b
e sent home regularly to watch.  The videos are to help the child develop early reading skills.\r\n\r\nParents that do not have ac
cess to a dvd player will have the opportunity to check out a dvd player to use for the year.  The plan is to use these videos and
educational dvd's for the years to come for other EL students.\r\nnannan
==================================================
The 51 fifth grade students that will cycle through my classroom this year all love learning, at least most of the time. At our sc
hool, 97.3% of the students receive free or reduced price lunch. Of the 560 students, 97.3% are minority students. \r\nThe school
has a vibrant community that loves to get together and celebrate. Around Halloween there is a whole school parade to show off the
beautiful costumes that students wear. On Cinco de Mayo we put on a big festival with crafts made by the students, dances, and gam
es. At the end of the year the school hosts a carnival to celebrate the hard work put in during the school year, with a dunk tank
being the most popular activity.My students will use these five brightly colored Hokki stools in place of regular, stationary, 4-l
egged chairs. As I will only have a total of ten in the classroom and not enough for each student to have an individual one, they
will be used in a variety of ways. During independent reading time they will be used as special chairs students will each use on o
ccasion. I will utilize them in place of chairs at my small group tables during math and reading times. The rest of the day they w
ill be used by the students who need the highest amount of movement in their life in order to stay focused on school.\r\n\r\nWhene
ver asked what the classroom is missing, my students always say more Hokki Stools. They can't get their fill of the 5 stools we al
ready have. When the students are sitting in group with me on the Hokki Stools, they are always moving, but at the same time doing
their work. Anytime the students get to pick where they can sit, the Hokki Stools are the first to be taken. There are always stud
ents who head over to the kidney table to get one of the stools who are disappointed as there are not enough of them. \r\n\r\nWe a
sk a lot of students to sit for 7 hours a day. The Hokki stools will be a compromise that allow my students to do desk work and mo
ve at the same time. These stools will help students to meet their 60 minutes a day of movement by allowing them to activate their
core muscles for balance while they sit. For many of my students, these chairs will take away the barrier that exists in schools f
or a child who can't sit still.nannan
==================================================
How do you remember your days of school? Was it in a sterile environment with plain walls, rows of desks, and a teacher in front o
f the room? A typical day in our room is nothing like that. I work hard to create a warm inviting themed room for my students look
forward to coming to each day.\r\n\r\nMy class is made up of 28 wonderfully unique boys and girls of mixed races in Arkansas.\r\nT
hey attend a Title I school, which means there is a high enough percentage of free and reduced-price lunch to qualify. Our school
is an \"open classroom\" concept, which is very unique as there are no walls separating the classrooms. These 9 and 10 year-old st
udents are very eager learners; they are like sponges, absorbing all the information and experiences and keep on wanting more.With
these resources such as the comfy red throw pillows and the whimsical nautical hanging decor and the blue fish nets, I will be abl
e to help create the mood in our classroom setting to be one of a themed nautical environment. Creating a classroom environment is
```

```
very important in the success in each and every child's education. The nautical photo props will be used with each child as they s
tep foot into our classroom for the first time on Meet the Teacher evening. I'll take pictures of each child with them, have them
developed, and then hung in our classroom ready for their first day of 4th grade.  This kind gesture will set the tone before even
the first day of school! The nautical thank you cards will be used throughout the year by the students as they create thank you ca
rds to their team groups.\r\n\r\nYour generous donations will help me to help make our classroom a fun, inviting, learning environ
ment from day one.\r\n\r\nIt costs lost of money out of my own pocket on resources to get our classroom ready. Please consider hel
ping with this project to make our new school year a very successful one. Thank you!nannan
==================================================
My kindergarten students have varied disabilities ranging from speech and language delays, cognitive delays, gross/fine motor dela
ys, to autism. They are eager beavers and always strive to work their hardest working past their limitations. \r\n\r\nThe material
s we have are the ones I seek out for my students. I teach in a Title I school where most of the students receive free or reduced
price lunch.  Despite their disabilities and limitations, my students love coming to school and come eager to learn and explore.Ha
ve you ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting? This is how my kids f
eel all the time. The want to be able to move as they learn or so they say.Wobble chairs are the answer and I love then because th
ey develop their core, which enhances gross motor and in Turn fine motor skills. \r\nThey also want to learn through games, my kid
s don't want to sit and do worksheets. They want to learn to count by jumping and playing. Physical engagement is the key to our s
uccess. The number toss and color and shape mats can make that happen. My students will forget they are doing work and just have t
he fun a 6 year old deserves.nannan
==================================================
The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires. -William A.
Ward\r\n\r\nMy school has 803 students which is makeup is 97.6% African-American, making up the largest segment of the student bod
y. A typical school in Dallas is made up of 23.2% African-American students. Most of the students are on free or reduced lunch. We
aren't receiving doctors, lawyers, or engineers children from rich backgrounds or neighborhoods. As an educator I am inspiring min
ds of young children and we focus not only on academics but one smart, effective, efficient, and disciplined students with good ch
aracter.In our classroom we can utilize the Bluetooth for swift transitions during class. I use a speaker which doesn't amplify th
e sound enough to receive the message. Due to the volume of my speaker my students can't hear videos or books clearly and it isn't
making the lessons as meaningful. But with the bluetooth speaker my students will be able to hear and I can stop, pause and replay
it at any time.\r\nThe cart will allow me to have more room for storage of things that are needed for the day and has an extra par
t to it I can use.  The table top chart has all of the letter, words and pictures for students to learn about different letters an
d it is more accessible.nannan
==================================================
```

In [13]:

```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 [14]:
```
sent = decontracted(project_data['essay'].values[20000])
print(sent)
print("="*50)
```

```
My kindergarten students have varied disabilities ranging from speech and language delays, cognitive delays, gross/fine motor dela
ys, to autism. They are eager beavers and always strive to work their hardest working past their limitations. \r\n\r\nThe material
s we have are the ones I seek out for my students. I teach in a Title I school where most of the students receive free or reduced
price lunch.  Despite their disabilities and limitations, my students love coming to school and come eager to learn and explore.Ha
ve you ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting? This is how my kids f
eel all the time. The want to be able to move as they learn or so they say.Wobble chairs are the answer and I love then because th
ey develop their core, which enhances gross motor and in Turn fine motor skills. \r\nThey also want to learn through games, my kid
s do not want to sit and do worksheets. They want to learn to count by jumping and playing. Physical engagement is the key to our
success. The number toss and color and shape mats can make that happen. My students will forget they are doing work and just have
the fun a 6 year old deserves.nannan
==================================================
```

In [15]:
```
# \r \n \t remove from string python: http://texthandler.com/info/remove-line-breaks-python/
sent = sent.replace('\\r', ' ')
sent = sent.replace('\\"', ' ')
sent = sent.replace('\\n', ' ')
print(sent)
```

```
My kindergarten students have varied disabilities ranging from speech and language delays, cognitive delays, gross/fine motor dela
ys, to autism. They are eager beavers and always strive to work their hardest working past their limitations.     The materials we
have are the ones I seek out for my students. I teach in a Title I school where most of the students receive free or reduced price
lunch.  Despite their disabilities and limitations, my students love coming to school and come eager to learn and explore.Have you
ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting? This is how my kids feel all
the time. The want to be able to move as they learn or so they say.Wobble chairs are the answer and I love then because they devel
op their core, which enhances gross motor and in Turn fine motor skills.   They also want to learn through games, my kids do not w
ant to sit and do worksheets. They want to learn to count by jumping and playing. Physical engagement is the key to our success. T
he number toss and color and shape mats can make that happen. My students will forget they are doing work and just have the fun a
6 year old deserves.nannan
```

In [16]:
```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
print(sent)
```

```
My kindergarten students have varied disabilities ranging from speech and language delays cognitive delays gross fine motor delays
to autism They are eager beavers and always strive to work their hardest working past their limitations The materials we have are
the ones I seek out for my students I teach in a Title I school where most of the students receive free or reduced price lunch Des
pite their disabilities and limitations my students love coming to school and come eager to learn and explore Have you ever felt l
```

```
ike you had ants in your pants and you needed to groove and move as you were in a meeting This is how my kids feel all the time Th
e want to be able to move as they learn or so they say Wobble chairs are the answer and I love then because they develop their cor
e which enhances gross motor and in Turn fine motor skills They also want to learn through games my kids do not want to sit and do
worksheets They want to learn to count by jumping and playing Physical engagement is the key to our success The number toss and co
lor and shape mats can make that happen My students will forget they are doing work and just have the fun a 6 year old deserves na
nnan
```

In [17]:

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

In [18]:

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

```
100%|██████████████████████████████████████████████████████████████████████████| 109248/109248 [00:58<00:00, 1863.09it/s]
```

In [19]:

```
# after preprocesing
preprocessed_essays[20000]
```

Out[19]:
```
'my kindergarten students varied disabilities ranging speech language delays cognitive delays gross fine motor delays autism they
eager beavers always strive work hardest working past limitations the materials ones i seek students i teach title i school studen
ts receive free reduced price lunch despite disabilities limitations students love coming school come eager learn explore have eve
r felt like ants pants needed groove move meeting this kids feel time the want able move learn say wobble chairs answer i love dev
elop core enhances gross motor turn fine motor skills they also want learn games kids not want sit worksheets they want learn coun
t jumping playing physical engagement key success the number toss color shape mats make happen my students forget work fun 6 year
old deserves nannan'
```

In [20]:
```
preprocessed_essays[0]
```

Out[20]:
```
'my students english learners working english second third languages we melting pot refugees immigrants native born americans brin
ging gift language school we 24 languages represented english learner program students every level mastery we also 40 countries re
presented families within school each student brings wealth knowledge experiences us open eyes new cultures beliefs respect the li
mits language limits world ludwig wittgenstein our english learner strong support system home begs resources many times parents le
arning read speak english along side children sometimes creates barriers parents able help child learn phonetics letter recognitio
n reading skills by providing dvd players students able continue mastery english language even no one home able assist all familie
s students within level 1 proficiency status offered part program these educational videos specially chosen english learner teache
r sent home regularly watch the videos help child develop early reading skills parents not access dvd player opportunity check dvd
player use year the plan use videos educational dvd years come el students nannan'
```

1.7 Preprocessing of $project_t it \leq$

In [21]:
```
project_data.head(2)
```

Out[21]:

	Unnamed: 0	id	teacher_id	school_state	project_submitted_datetime	project_title	project_essay_1	project_essay_2	project
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	IN	05-12-2016 13:43	Educational Support for English Learners at Home	My students are English learners that are work...	\"The limits of your language are the limits o...	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	FL	25-10-2016 09:22	Wanted: Projector for Hungry Learners	Our students arrive to our school eager to lea...	The projector we need for our school is very c...	

In [22]:
```
# printing some random essays.
print(project_data['project_title'].values[0])
print("="*50)
print(project_data['project_title'].values[150])
print("="*50)
print(project_data['project_title'].values[1000])
print("="*50)
print(project_data['project_title'].values[20000])
print("="*50)
print(project_data['project_title'].values[99999])
print("="*50)
```

```
Educational Support for English Learners at Home
==================================================
More Movement with Hokki Stools
==================================================
Sailing Into a Super 4th Grade Year
==================================================
We Need To Move It While We Input It!
==================================================
Inspiring Minds by Enhancing the Educational Experience
==================================================
```

In [23]:
```
sent_title = decontracted(project_data['project_title'].values[20000])
print(sent_title)
print("="*50)
```

```
We Need To Move It While We Input It!
==================================================
```

In [24]:
```
# \r \n \t remove from string python: http://texthandler.com/info/remove-line-breaks-python/
sent_title = sent_title.replace('\\r', ' ')
sent_title = sent_title.replace('\\"', ' ')
sent_title = sent_title.replace('\\n', ' ')
print(sent_title)
```

```
We Need To Move It While We Input It!
```

In [25]:
```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent_title = re.sub('[^A-Za-z0-9]+', ' ', sent_title)
print(sent_title)
```

```
We Need To Move It While We Input It
```

In [26]:
```
# Combining all the above statemennts
from tqdm import tqdm
```

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

```
100%|██████████████████████████████████████████████████████████████████████████| 109248/109248 [00:02<00:00, 42490.26it/s]
```

In [27]:

```
# after preprocesing
preprocessed_title[10]
```

Out[27]:

```
'reading changes lives'
```

In [28]:

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

```
100%|██████████████████████████████████████████████████████████████████████████| 109248/109248 [00:06<00:00, 18035.34it/s]
```

1.8 Numeric feature for Text

1.8.1 Numerric feature for essay

In [29]:

```
# Suggestion 5.you can try improving the score using feature engineering hacks.Try including length,summary
# and observe the results and re-submit the assignment.
```

```
# https://stackoverflow.com/questions/18827198/python-count-number-of-words-in-a-list-strings
preprocessed_essays_wc = []
for item in tqdm(preprocessed_essays):
    preprocessed_essays_wc.append(len(item.split()))

print(preprocessed_essays_wc[101])
```

```
100%|██████████████████████████████████████████████████████████████████████████| 109248/109248 [00:01<00:00, 95211.06it/s]
141
```

1.8.2 Numerric feature for title

In [30]:
```
# Suggestion 5.you can try improving the score using feature engineering hacks.Try including length,summary
# and observe the results and re-submit the assignment.

# https://stackoverflow.com/questions/18827198/python-count-number-of-words-in-a-list-strings
preprocessed_title_wc = []
for item in tqdm(preprocessed_title):
    preprocessed_title_wc.append(len(item.split()))

print(preprocessed_title_wc[101])
```

```
100%|██████████████████████████████████████████████████████████████████████████| 109248/109248 [00:00<00:00, 1250056.13it/s]
3
```

1.8.3 Vectorizing Numerical features

In [31]:
```
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')
```

In [32]:
```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
from sklearn.preprocessing import StandardScaler

# price_standardized = standardScalar.fit(project_data['price'].values)
# this will rise the error
# ValueError: Expected 2D array, got 1D array instead: array=[725.05 213.03 329.   ... 399.   287.73   5.5 ].
# Reshape your data either using array.reshape(-1, 1)

price_scalar = StandardScaler()
price_scalar.fit(project_data['price'].values.reshape(-1,1)) # finding the mean and standard deviation of this data
print(f"Mean : {price_scalar.mean_[0]}, Standard deviation : {np.sqrt(price_scalar.var_[0])}")
```

```
# Now standardize the data with above maen and variance.
price_standardized = price_scalar.transform(project_data['price'].values.reshape(-1, 1))
```

```
Mean : 298.1193425966608, Standard deviation : 367.49634838483496
```

In [33]:
```
price_standardized
```

Out[33]:
```
array([[-0.3905327 ],
       [ 0.00239637],
       [ 0.59519138],
       ...,
       [-0.15825829],
       [-0.61243967],
       [-0.51216657]])
```

Computing Sentiment Scores

In [34]:
```
## https://monkeylearn.com/sentiment-analysis/
## http://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html
#
#import nltk
#from nltk.sentiment.vader import SentimentIntensityAnalyzer
#
#import nltk
#nltk.download('vader_lexicon')
#
#sid = SentimentIntensityAnalyzer()
#
#for_sentiment = 'a person is a person no matter how small dr seuss i teach the smallest students with the biggest enthusiasm \
#for learning my students learn in many different ways using all of our senses and multiple intelligences i use a wide range\
#of techniques to help all my students succeed students in my class come from a variety of different backgrounds which makes\
#for wonderful sharing of experiences and cultures including native americans our school is a caring community of successful \
#learners which can be seen through collaborative student project based learning in and out of the classroom kindergarteners \
#in my class love to work with hands on materials and have many different opportunities to practice a skill before it is\
#mastered having the social skills to work cooperatively with friends is a crucial aspect of the kindergarten curriculum\
#montana is the perfect place to learn about agriculture and nutrition my students love to role play in our pretend kitchen\
#in the early childhood classroom i have had several kids ask me can we try cooking with real food i will take their idea \
#and create common core cooking lessons where we learn important math and writing concepts while cooking delicious healthy \
#food for snack time my students will have a grounded appreciation for the work that went into making the food and knowledge \
#of where the ingredients came from as well as how it is healthy for their bodies this project would expand our learning of \
#nutrition and agricultural cooking recipes by having us peel our own apples to make homemade applesauce make our own bread \
#and mix up healthy plants from our classroom garden in the spring we will also create our own cookbooks to be printed and \
#shared with families students will gain math and literature skills as well as a life long enjoyment for healthy cooking \
#nannan'
```

```
#ss = sid.polarity_scores(for_sentiment)
#
## The end=' ' is just to say that you want a space after the end of the statement instead of a new line character.
#for k in ss:
#    print('{0}: {1}, '.format(k, ss[k]), end='')
#
#for k in ss:
#    print('{0}: {1}, '.format(k, ss[k]))
#
# we can use these 4 things as features/attributes (neg, neu, pos, compound)
# neg: 0.0, neu: 0.753, pos: 0.247, compound: 0.93
#print(type(ss))
#print(ss)
```

In [35]:

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

import nltk
nltk.download('vader_lexicon')

sid = SentimentIntensityAnalyzer()

from tqdm import tqdm
from tqdm import tqdm_notebook
preprocessed_sentiments = []
# tqdm is for printing the status bar
for sentance in tqdm_notebook(project_data['essay'].values):
    sentiment = []
    sentiment = sid.polarity_scores(sentance)
    preprocessed_sentiments.append([sentiment['neg'], sentiment['pos'], sentiment['neu'], sentiment['compound']])
```

```
[nltk_data] Downloading package vader_lexicon to C:\Users\Prabhat
[nltk_data]     .LAPTOP-486AQERF\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

In [36]:

```
print(type(preprocessed_sentiments))
print(preprocessed_sentiments[1:5])
#print(preprocessed_sentiments([sentiment['neg']]))
print(sentiment['neg'])

project_data[['neg', 'pos', 'neu', 'compound']] = pd.DataFrame(preprocessed_sentiments)
```

```
<class 'list'>
```

```
[[0.037, 0.112, 0.851, 0.9267], [0.058, 0.179, 0.764, 0.995], [0.052, 0.214, 0.733, 0.9931], [0.016, 0.087, 0.897, 0.9192]]
0.023
```

In [37]:
```
print(project_data.columns.values)
project_data['neg'].values
```

```
['Unnamed: 0' 'id' 'teacher_id' 'school_state'
 'project_submitted_datetime' 'project_title' 'project_essay_1'
 'project_essay_2' 'project_essay_3' 'project_essay_4'
 'project_resource_summary' 'teacher_number_of_previously_posted_projects'
 'project_is_approved' 'clean_categories' 'clean_subcategories'
 'clean_grade' 'clean_tea_pfx' 'essay' 'price' 'quantity' 'neg' 'pos'
 'neu' 'compound']
```

Out[37]:
```
array([0.008, 0.037, 0.058, ..., 0.   , 0.013, 0.023])
```

Adding word count for essay and Title

In [38]:
```
project_data['essay_wc'] = preprocessed_essays_wc

project_data['title_wc'] = preprocessed_title_wc
```

Adding Preprocessed essay and Preprocessed Title

In [39]:
```
project_data['essay'] = preprocessed_essays
project_data['project_title'] = preprocessed_title
```

In [40]:
```
project_data.columns
```

Out[40]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'school_state',
       'project_submitted_datetime', 'project_title', 'project_essay_1',
       'project_essay_2', 'project_essay_3', 'project_essay_4',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'clean_grade',
       'clean_tea_pfx', 'essay', 'price', 'quantity', 'neg', 'pos', 'neu',
       'compound', 'essay_wc', 'title_wc'],
      dtype='object')
```

1.9 Preparing data for models

In [41]:
```
project_data.columns
```

Out[41]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'school_state',
       'project_submitted_datetime', 'project_title', 'project_essay_1',
       'project_essay_2', 'project_essay_3', 'project_essay_4',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'clean_grade',
       'clean_tea_pfx', 'essay', 'price', 'quantity', 'neg', 'pos', 'neu',
       'compound', 'essay_wc', 'title_wc'],
      dtype='object')
```

we are going to consider

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

- project_title : text data
- text : text data
- project_resource_summary: text data (optinal)

- quantity : numerical (optinal)
- teacher_number_of_previously_posted_projects : numerical
- price : numerical
```

Computing Sentiment Scores

In [42]:
```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# import nltk
# nltk.download('vader_lexicon')

sid = SentimentIntensityAnalyzer()

for_sentiment = 'a person is a person no matter how small dr seuss i teach the smallest students with the biggest enthusiasm \
for learning my students learn in many different ways using all of our senses and multiple intelligences i use a wide range\
of techniques to help all my students succeed students in my class come from a variety of different backgrounds which makes\
for wonderful sharing of experiences and cultures including native americans our school is a caring community of successful \
learners which can be seen through collaborative student project based learning in and out of the classroom kindergarteners \
in my class love to work with hands on materials and have many different opportunities to practice a skill before it is\
mastered having the social skills to work cooperatively with friends is a crucial aspect of the kindergarten curriculum\
```

```
montana is the perfect place to learn about agriculture and nutrition my students love to role play in our pretend kitchen\
in the early childhood classroom i have had several kids ask me can we try cooking with real food i will take their idea \
and create common core cooking lessons where we learn important math and writing concepts while cooking delicious healthy \
food for snack time my students will have a grounded appreciation for the work that went into making the food and knowledge \
of where the ingredients came from as well as how it is healthy for their bodies this project would expand our learning of \
nutrition and agricultural cooking recipes by having us peel our own apples to make homemade applesauce make our own bread \
and mix up healthy plants from our classroom garden in the spring we will also create our own cookbooks to be printed and \
shared with families students will gain math and literature skills as well as a life long enjoyment for healthy cooking \
nannan'
ss = sid.polarity_scores(for_sentiment)

for k in ss:
    print('{0}: {1}, '.format(k, ss[k]), end='')

# we can use these 4 things as features/attributes (neg, neu, pos, compound)
# neg: 0.01, neu: 0.745, pos: 0.245, compound: 0.9975
```

```
neg: 0.01, neu: 0.745, pos: 0.245, compound: 0.9975,
```

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 _ 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**

In [43]:
```
##taking 50K datapoint
project_data50K=project_data[:50000]
#project_data100K=project_data[:100000]
#X=project_data100K
X=project_data50K
print(project_data50K.shape)
#print(project_data100K.shape)
print(X.shape)
```

```
(50000, 26)
(50000, 26)
```

In [44]:
```
y = project_data['
                 '].values
project_data=project_data.drop(['project_is_approved'], axis=1, inplace=True)
#print(y.shape)
project_data.head(1)

y50K=y[:50000]
y=y50K
```

In [45]:
```
print(X.shape)
print(y.shape)
```

```
(50000, 26)
(50000,)
```

In [46]:
```
# train test split | https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
# spliting Xq and Yq in Train(further into Train and CV) and Test matrix
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, stratify=y)
#X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.33, stratify=y_train)

print(X_train.shape, y_train.shape)
#print(X_cv.shape, y_cv.shape)
print(X_test.shape, y_test.shape)

print("="*100)
```

```
(33500, 26) (33500,)
(16500, 26) (16500,)
====================================================================================================
```

2.1.1 Make Data Model Ready: encoding school_state categorical data

In [47]:
```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(min_df=10,ngram_range=(1,2), max_features=5000)
vectorizer.fit(X_train['school_state'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_state_ohe = vectorizer.transform(X_train['school_state'].values)
#X_cv_state_ohe = vectorizer.transform(X_cv['school_state'].values)
X_test_state_ohe = vectorizer.transform(X_test['school_state'].values)

print("school_state After vectorizations")
print(X_train_state_ohe.shape, y_train.shape)
#print(X_cv_state_ohe.shape, y_cv.shape)
print(X_test_state_ohe.shape, y_test.shape)
st=vectorizer.get_feature_names()
print(vectorizer.get_feature_names())
print("="*100)
```

```
school_state After vectorizations
(33500, 51) (33500,)
(16500, 51) (16500,)
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga', 'hi', 'ia', 'id', 'il', 'in', 'ks', 'ky', 'la', 'ma', 'md', 'm
e', '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']
====================================================================================================
```

2.1.2 Make Data Model Ready: encoding clean_categories

In [48]:
```
from sklearn.feature_extraction.text import CountVectorizer
#vectorizer = CountVectorizer(min_df=10,ngram_range=(1,2), max_features=5000)
vectorizer = CountVectorizer(vocabulary =list(sorted_cat_dict.keys()),lowercase =False,binary=True)
vectorizer.fit(X_train['clean_categories'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_clean_ohe = vectorizer.transform(X_train['clean_categories'].values)
#X_cv_clean_ohe = vectorizer.transform(X_cv['clean_categories'].values)
X_test_clean_ohe = vectorizer.transform(X_test['clean_categories'].values)

print("clean_categories After vectorizations")
print(X_train_clean_ohe.shape, y_train.shape)
#print(X_cv_clean_ohe.shape, y_cv.shape)
print(X_test_clean_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
cc=vectorizer.get_feature_names()
print(cc)
print("="*100)
```

```
clean_categories After vectorizations
(33500, 9) (33500,)
(16500, 9) (16500,)
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearning', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Lit
eracy_Language']
====================================================================================================
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearning', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Lit
eracy_Language']
====================================================================================================
```

2.1.3 Make Data Model Ready: encoding clean_subcategories

In [49]:
```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary =list(sorted_sub_cat_dict.keys()),lowercase =False,binary=True)
vectorizer.fit(X_train['clean_subcategories'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_cleanSub_ohe = vectorizer.transform(X_train['clean_subcategories'].values)
#X_cv_cleanSub_ohe = vectorizer.transform(X_cv['clean_subcategories'].values)
X_test_cleanSub_ohe = vectorizer.transform(X_test['clean_subcategories'].values)

print("clean_subcategories After vectorizations")
print(X_train_cleanSub_ohe.shape, y_train.shape)
```

```
#print(X_cv_cleanSub_ohe.shape, y_cv.shape)
print(X_test_cleanSub_ohe.shape, y_test.shape)
cst=vectorizer.get_feature_names()
#print(cst)
print("="*100)
```

```
clean_subcategories After vectorizations
(33500, 30) (33500,)
(16500, 30) (16500,)
====================================================================================================
```

2.1.4 Make Data Model Ready: encoding project_grade_category

In [50]:

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary =list(sorted_prj_grade_cat_dict.keys()),lowercase =False,binary=True)
vectorizer.fit(X_train['clean_grade'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_grade_ohe = vectorizer.transform(X_train['clean_grade'].values)
#X_cv_grade_ohe = vectorizer.transform(X_cv['clean_grade'].values)
X_test_grade_ohe = vectorizer.transform(X_test['clean_grade'].values)

print("project_grade_category After vectorizations")
print(X_train_grade_ohe.shape, y_train.shape)
#print(X_cv_grade_ohe.shape, y_cv.shape)
print(X_test_grade_ohe.shape, y_test.shape)

pgc=vectorizer.get_feature_names()
print(pgc)
print("="*100)
```

```
project_grade_category After vectorizations
(33500, 4) (33500,)
(16500, 4) (16500,)
['9-12', '6-8', '3-5', 'PreK-2']
====================================================================================================
```

2.1.5 Make Data Model Ready: encoding teacher_prefix

In [51]:

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary =list(sorted_tea_pfx_cat_dict.keys()),lowercase =False,binary=True)
#https://stackoverflow.com/questions/52736900/how-to-solve-the-attribute-error-float-object-has-no-attribute-split-in-pyth
vectorizer.fit(X_train['clean_tea_pfx'].astype(str).values) # fit has to happen only on train data
```

```
# we use the fitted CountVectorizer to convert the text to vector
X_train_teacher_ohe = vectorizer.transform(X_train['clean_tea_pfx'].astype(str).values)
#X_cv_teacher_ohe = vectorizer.transform(X_cv['clean_tea_pfx'].astype(str).values)
X_test_teacher_ohe = vectorizer.transform(X_test['clean_tea_pfx'].astype(str).values)

print("teacher_prefix After vectorizations")
print(X_train_teacher_ohe.shape, y_train.shape)
#print(X_cv_teacher_ohe.shape, y_cv.shape)
print(X_test_teacher_ohe.shape, y_test.shape)
tp=vectorizer.get_feature_names()
print(tp)
print("="*100)
```

```
teacher_prefix After vectorizations
(33500, 5) (33500,)
(16500, 5) (16500,)
['Dr', 'Teacher', 'Mr', 'Ms', 'Mrs']
====================================================================================================
```

2.2 Make Data Model Ready: encoding numerical, categorical features

2.2.1 Make Data Model Ready: encoding numerical | quantity

In [52]:

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

X_train_quantity_norm = normalizer.transform(X_train['quantity'].values.reshape(-1,1))
#X_cv_quantity_norm = normalizer.transform(X_cv['quantity'].values.reshape(-1,1))
X_test_quantity_norm = normalizer.transform(X_test['quantity'].values.reshape(-1,1))

print("quantity After vectorizations")
print(X_train_quantity_norm.shape, y_train.shape)
#print(X_cv_quantity_norm.shape, y_cv.shape)
```

```
print(X_test_quantity_norm.shape, y_test.shape)
print("="*100)
```

```
quantity After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
====================================================================================================
```

2.2.2 Make Data Model Ready: encoding numerical| teacher_number_of_previously_posted_projects

In [53]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))

X_train_TprevPrj_norm = normalizer.transform(X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))
#X_cv_TprevPrj_norm = normalizer.transform(X_cv['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))
X_test_TprevPrj_norm = normalizer.transform(X_test['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))

print("teacher_number_of_previously_posted_projects After vectorizations")
print(X_train_TprevPrj_norm.shape, y_train.shape)
#print(X_cv_TprevPrj_norm.shape, y_cv.shape)
print(X_test_TprevPrj_norm.shape, y_test.shape)
print("="*100)
```

```
teacher_number_of_previously_posted_projects After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
====================================================================================================
```

2.2.3 Make Data Model Ready: encoding numerical | price

In [54]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
```

```
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['price'].values.reshape(-1,1))

X_train_price_norm = normalizer.transform(X_train['price'].values.reshape(-1,1))
#X_cv_price_norm = normalizer.transform(X_cv['price'].values.reshape(-1,1))
X_test_price_norm = normalizer.transform(X_test['price'].values.reshape(-1,1))

print("Price After vectorizations")
print(X_train_price_norm.shape, y_train.shape)
#print(X_cv_price_norm.shape, y_cv.shape)
print(X_test_price_norm.shape, y_test.shape)
print("="*100)
```

```
Price After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
====================================================================================================
```

In [55]:

```
h=['price','quantity','teacher_number_of_previously_posted_projects']
print(type(h))
```

```
<class 'list'>
```

2.2.4 Make Data Model Ready: encoding numerical | sentimental score

2.2.4.1 Make Data Model Ready: encoding numerical | sentimental score | neg

In [56]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['neg'].values.reshape(-1,1))

X_train_neg_norm = normalizer.transform(X_train['neg'].values.reshape(-1,1))
#X_cv_neg_norm = normalizer.transform(X_cv['neg'].values.reshape(-1,1))
X_test_neg_norm = normalizer.transform(X_test['neg'].values.reshape(-1,1))

print("neg After vectorizations")
```

```
print(X_train_neg_norm.shape, y_train.shape)
#print(X_cv_neg_norm.shape, y_cv.shape)
print(X_test_neg_norm.shape, y_test.shape)
print("="*100)
```

```
neg After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
====================================================================================================
```

2.2.4.2 Make Data Model Ready: encoding numerical | sentimental score | pos

In [57]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['pos'].values.reshape(-1,1))

X_train_pos_norm = normalizer.transform(X_train['pos'].values.reshape(-1,1))
#X_cv_pos_norm = normalizer.transform(X_cv['pos'].values.reshape(-1,1))
X_test_pos_norm = normalizer.transform(X_test['pos'].values.reshape(-1,1))

print("pos After vectorizations")
print(X_train_pos_norm.shape, y_train.shape)
#print(X_cv_pos_norm.shape, y_cv.shape)
print(X_test_pos_norm.shape, y_test.shape)
print("="*100)
```

```
pos After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
====================================================================================================
```

2.2.4.3 Make Data Model Ready: encoding numerical | sentimental score | neu

In [58]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
```

```
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['neu'].values.reshape(-1,1))

X_train_neu_norm = normalizer.transform(X_train['neu'].values.reshape(-1,1))
#X_cv_neu_norm = normalizer.transform(X_cv['neu'].values.reshape(-1,1))
X_test_neu_norm = normalizer.transform(X_test['neu'].values.reshape(-1,1))

print("neu After vectorizations")
print(X_train_neu_norm.shape, y_train.shape)
#print(X_cv_neu_norm.shape, y_cv.shape)
print(X_test_neu_norm.shape, y_test.shape)
print("="*100)
```

```
neu After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
====================================================================================================
```

2.2.4.4 Make Data Model Ready: encoding numerical | sentimental score | compound

In [59]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['compound'].values.reshape(-1,1))

X_train_compound_norm = normalizer.transform(X_train['compound'].values.reshape(-1,1))
#X_cv_compound_norm = normalizer.transform(X_cv['compound'].values.reshape(-1,1))
X_test_compound_norm = normalizer.transform(X_test['compound'].values.reshape(-1,1))

print("compound After vectorizations")
print(X_train_compound_norm.shape, y_train.shape)
#print(X_cv_compound_norm.shape, y_cv.shape)
print(X_test_compound_norm.shape, y_test.shape)
print("="*100)
```

```
compound After vectorizations
(33500, 1) (33500,)
```

```
(16500, 1) (16500,)
==================================================================================================
```

2.2.5 Make Data Model Ready: encoding numerical | number of words in the title

In [60]:
```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['title_wc'].values.reshape(-1,1))

X_train_title_wc_norm = normalizer.transform(X_train['title_wc'].values.reshape(-1,1))
#X_cv_title_wc_norm = normalizer.transform(X_cv['title_wc'].values.reshape(-1,1))
X_test_title_wc_norm = normalizer.transform(X_test['title_wc'].values.reshape(-1,1))

print("title_wc After vectorizations")
print(X_train_title_wc_norm.shape, y_train.shape)
#print(X_cv_title_wc_norm.shape, y_cv.shape)
print(X_test_title_wc_norm.shape, y_test.shape)
print("="*100)
```

```
title_wc After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
==================================================================================================
```

2.2.6 Make Data Model Ready: encoding numerical | number of words in the essay

In [61]:
```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['essay_wc'].values.reshape(-1,1))

X_train_essay_wc_norm = normalizer.transform(X_train['essay_wc'].values.reshape(-1,1))
```

```
#X_cv_essay_wc_norm = normalizer.transform(X_cv['essay_wc'].values.reshape(-1,1))
X_test_essay_wc_norm = normalizer.transform(X_test['essay_wc'].values.reshape(-1,1))

print("essay_wc After vectorizations")
print(X_train_essay_wc_norm.shape, y_train.shape)
#print(X_cv_essay_wc_norm.shape, y_cv.shape)
print(X_test_essay_wc_norm.shape, y_test.shape)
print("="*100)
```

```
essay_wc After vectorizations
(33500, 1) (33500,)
(16500, 1) (16500,)
====================================================================================================
```

2. TruncatedSVD

2.1 Selecting top 2000 words from $essay$ and $project_t it \leq$

In [62]:
```
# please write all the code with proper documentation, and proper titles for each subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debugging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the reader
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

- 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 _ values

In [63]:
```
# https://stackoverflow.com/questions/19377969/combine-two-columns-of-text-in-dataframe-in-pandas-python
# dataframe["period"] = dataframe["Year"].map(str) + dataframe["quarter"]
#project_data.info()
#print("Essay")
#print(project_data.essay[0])
#print(project_data.project_title[0])
#print(project_data.essay.head(2))
#print("Project_title")
#print(project_data.project_title.head(2))
```

```
X_train["EssayTitle"] = X_train.essay + X_train.project_title
X_test["EssayTitle"] = X_test.essay + X_test.project_title

#X_train["EssayTitle"] = X_train.preprocessed_essays+X_train.preprocessed_title
#X_test["EssayTitle"] = X_test.preprocessed_essays+X_test.preprocessed_title
#print(project_data.columns)
print(X_train.columns)
print(X_train.shape)
#print("EssayTitle")
#print(project_data.EssayTitle[0])
```

```
Index(['Unnamed: 0', 'id', 'teacher_id', 'school_state',
       'project_submitted_datetime', 'project_title', 'project_essay_1',
       'project_essay_2', 'project_essay_3', 'project_essay_4',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'clean_grade',
       'clean_tea_pfx', 'essay', 'price', 'quantity', 'neg', 'pos', 'neu',
       'compound', 'essay_wc', 'title_wc', 'EssayTitle'],
      dtype='object')
(33500, 27)
```

In [64]:
```
X_train["EssayTitle"]
```

Out[64]:
```
9115     liberty elementary title 1 school large percen...
13389    my students come diverse backgrounds they come...
13827    environment shapes experience no less true cla...
7263     my students diverse group ambitious enthusiast...
45303    my students diverse ethnicity also abilities t...
30987    my students love coming school everyday i teac...
16803    music truly helps change lives better for stud...
37341    our school consists entirely k 5 students spec...
37610    i 23 first graders nine girls fourteen boys th...
45753    my students enthusiastic dynamic resourceful l...
17107    for children playtime venue create understandi...
43244    every student deserves best education possible...
31783    my school small community central wi my studen...
8527     as teacher low income school high poverty town...
5423     my students diverse population within rural co...
9259     this first year wonderful elementary school my...
26995    i work public charter school really done lot t...
8239     i wonderful group active first graders i wide ...
30132    our robots not battlebots students put hearts ...
34230    we moving brand new building school year as i ...
31962    my students come diverse backgrounds they amaz...
26300    i privilege teaching 24 wonderful first grader...
46371    cuties i moving new school 2016 2017 school ye...
```

```
9057     my second grade classroom place learning disco...
48570    females traditionally left male dominated spor...
22359    technology become centerpiece learning 21st ce...
31051    my students incredible i 24 motivated english ...
41004    my students come school every day not knowing ...
150      the 51 fifth grade students cycle classroom ye...
40054    as end year approaching i realized much studen...
                                ...
32726    this year i working 14 students 4 different ki...
35896    in school district schools less continue strug...
41031    my school newly added special needs groups we ...
32581    my eager able students 3 5 years old they happ...
27011    my students live low income area we title 1 sc...
13302    connally junior high student population 90 low...
1037     our students amazing kids eagle explore school...
28714    i beyond proud teach school home many amazing ...
15737    i teach writing amazing group diverse kinderga...
29408    my students amazing they come world speak doze...
46521    my students 8th grade students steam program a...
36384    learning two languages inspires students const...
39285    my seventh graders dedicated excited learners ...
37181    my students not number experiences many childr...
4216     welcome classroom 207 our students the student...
32391    imagine kindergarten classroom moment when ste...
11685    my school empowers 538 students grades pre k f...
16889    my students come variety backgrounds some spea...
43528    our classroom filled opportunities movement co...
14353    i blessed beyond measure teach third grade tit...
36611    i amazing fun loving 6 7 year old children wan...
12583    my students special needs various socioeconomi...
9696     i teach special education students variety dis...
42208    i second year teacher i honor shaping 16 brill...
35789    i intervention specialist resource room studen...
18470    the students pre k classroom excited learn exp...
34404    my students attend small rural school 48 free ...
45647    my students attend high school standing high m...
24593    i lucky teach diverse school wonderful student...
13046    most students come low income background still...
Name: EssayTitle, Length: 33500, dtype: object
```

In [65]:

```
# https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
#idf_ : array, shape (n_features)
#The inverse document frequency (IDF) vector; only defined if use_idf is True.

from sklearn.feature_extraction.text import TfidfVectorizer

Tfidf_vectorizer = TfidfVectorizer(min_df=10,ngram_range=(1,1), max_features=5000,use_idf=True)
```

```
X_text_tfidf = Tfidf_vectorizer.fit_transform(X_train['EssayTitle'].values)

print("Essay After vectorizations")
tf=Tfidf_vectorizer.get_feature_names()
print(tf)
print("="*100)
```

```
Essay After vectorizations
['00', '000', '10', '100', '1000', '10th', '11', '110', '11th', '12', '120', '12th', '13', '14', '15', '150', '16', '17', '18', '1
80', '19', '1st', '20', '200', '2015', '2016', '2017', '21', '21st', '22', '23', '24', '25', '26', '27', '28', '29', '2nd', '30',
'300', '31', '32', '33', '34', '35', '36', '3d', '3doodler', '3rd', '40', '400', '45', '450', '48', '4th', '50', '500', '55', '5t
h', '60', '600', '65', '6th', '70', '700', '74', '75', '7th', '80', '800', '84', '85', '8th', '90', '900', '92', '94', '95', '96',
'97', '98', '99', '9th', 'abc', 'abilities', 'ability', 'able', 'about', 'absent', 'absolute', 'absolutely', 'absorb', 'abstract',
'abundance', 'abuse', 'academic', 'academically', 'academics', 'academy', 'accelerated', 'accept', 'acceptance', 'accepted', 'acce
pting', 'access', 'accessed', 'accessibility', 'accessible', 'accessing', 'accessories', 'accidents', 'accommodate', 'accommodatio
ns', 'accompany', 'accomplish', 'accomplished', 'accomplishing', 'accomplishment', 'accomplishments', 'according', 'account', 'acc
ountability', 'accountable', 'accounts', 'accuracy', 'accurate', 'accurately', 'accustomed', 'achieve', 'achieved', 'achievement',
'achievements', 'achievers', 'achieving', 'acquire', 'acquired', 'acquiring', 'acquisition', 'across', 'act', 'acting', 'action',
'actions', 'activate', 'active', 'actively', 'activities', 'activity', 'actual', 'actually', 'adapt', 'adaptations', 'adapted', 'a
daptive', 'add', 'added', 'adding', 'addition', 'additional', 'additionally', 'additions', 'address', 'addressed', 'addressing',
'adds', 'adequate', 'adhd', 'adjust', 'adjustable', 'administration', 'administrators', 'adolescents', 'adopted', 'adorable', 'ado
re', 'adult', 'adulthood', 'adults', 'advance', 'advanced', 'advancement', 'advances', 'advancing', 'advantage', 'advantages', 'ad
venture', 'adventures', 'adventurous', 'adversity', 'advocate', 'affect', 'affected', 'affects', 'affluent', 'afford', 'afforded',
'afraid', 'africa', 'african', 'after', 'afternoon', 'again', 'age', 'aged', 'ages', 'ago', 'agree', 'agreed', 'agricultural', 'ag
riculture', 'ahead', 'aid', 'aide', 'aides', 'aids', 'aim', 'aims', 'air', 'alabama', 'alaska', 'albert', 'alert', 'algebra', 'ali
gn', 'aligned', 'alike', 'alive', 'all', 'alleviate', 'allow', 'allowed', 'allowing', 'allows', 'almost', 'alone', 'along', 'along
side', 'aloud', 'alouds', 'alphabet', 'already', 'also', 'alternate', 'alternative', 'alternatives', 'although', 'always', 'amaz
e', 'amazed', 'amazes', 'amazing', 'amazingly', 'amazon', 'ambassadors', 'ambitious', 'america', 'american', 'americans', 'among',
'amongst', 'amount', 'amounts', 'ample', 'an', 'analysis', 'analytical', 'analyze', 'analyzing', 'anatomy', 'anchor', 'ancient',
'and', 'angeles', 'anger', 'animal', 'animals', 'animation', 'annotate', 'announcements', 'annual', 'another', 'answer', 'answerin
g', 'answers', 'anxiety', 'anxious', 'any', 'anymore', 'anyone', 'anything', 'anytime', 'anywhere', 'ap', 'apart', 'apartment', 'a
partments', 'app', 'appalachian', 'apparent', 'appeal', 'appealing', 'apple', 'applicable', 'application', 'applications', 'applie
d', 'applies', 'apply', 'applying', 'appreciate', 'appreciated', 'appreciation', 'appreciative', 'approach', 'approaches', 'approa
ching', 'appropriate', 'appropriately', 'approved', 'approximately', 'apps', 'ar', 'arabic', 'archery', 'architects', 'architectur
e', 'are', 'area', 'areas', 'arise', 'arizona', 'arkansas', 'arms', 'around', 'arrangement', 'arrangements', 'array', 'arrive', 'a
rrived', 'art', 'article', 'articles', 'artifacts', 'artist', 'artistic', 'artists', 'arts', 'artwork', 'as', 'asd', 'asian', 'asi
de', 'ask', 'asked', 'asking', 'aspect', 'aspects', 'aspirations', 'aspire', 'aspiring', 'assemblies', 'assess', 'assessment', 'as
sessments', 'asset', 'assign', 'assigned', 'assignment', 'assignments', 'assist', 'assistance', 'assisting', 'associate', 'associa
ted', 'association', 'assortment', 'assure', 'at', 'athletes', 'athletic', 'athletics', 'atlanta', 'atmosphere', 'attached', 'atta
in', 'attempt', 'attempting', 'attend', 'attendance', 'attended', 'attending', 'attention', 'attentive', 'attitude', 'attitudes',
'audience', 'audio', 'auditory', 'august', 'authentic', 'author', 'authors', 'autism', 'autistic', 'availability', 'available', 'a
venue', 'avenues', 'average', 'avid', 'avoid', 'award', 'awarded', 'awards', 'aware', 'awareness', 'away', 'awe', 'awesome', 'babi
es', 'baby', 'baccalaureate', 'back', 'background', 'backgrounds', 'backpack', 'backpacks', 'backs', 'bad', 'bag', 'bags', 'balanc
e', 'balanced', 'balancing', 'ball', 'balls', 'baltimore', 'band', 'bands', 'bank', 'bar', 'bare', 'barely', 'barrier', 'barrier
s', 'bars', 'base', 'baseball', 'based', 'bases', 'basic', 'basically', 'basics', 'basis', 'basketball', 'basketballs', 'baskets',
'bass', 'bathroom', 'batteries', 'battery', 'battle', 'bay', 'be', 'beach', 'beads', 'bean', 'beanbag', 'beanbags', 'bears', 'bea
t', 'beautiful', 'beautifully', 'beauty', 'became', 'because', 'become', 'becomes', 'becoming', 'bed', 'beds', 'bee', 'bees', 'bef
```

```
ore', 'beg', 'began', 'begging', 'begin', 'beginning', 'begins', 'begun', 'behaved', 'behavior', 'behavioral', 'behaviorally', 'be
haviors', 'behind', 'being', 'beings', 'belief', 'beliefs', 'believe', 'believer', 'believes', 'believing', 'bell', 'bellies', 'be
lls', 'belong', 'belonging', 'belongings', 'belongs', 'beloved', 'benches', 'beneficial', 'benefit', 'benefits', 'benjamin', 'besi
des', 'best', 'better', 'beyond', 'bi', 'big', 'bigger', 'biggest', 'bike', 'bikes', 'bilingual', 'bill', 'bin', 'binder', 'binder
s', 'bins', 'biographies', 'biology', 'birds', 'birthday', 'bit', 'black', 'blank', 'blend', 'blended', 'blending', 'blessed', 'bl
essing', 'block', 'blocks', 'blog', 'blood', 'blossom', 'blow', 'blue', 'bluetooth', 'board', 'boards', 'bodies', 'body', 'bond',
'bones', 'bonus', 'boogie', 'book', 'books', 'bookshelf', 'bookshelves', 'boost', 'bored', 'boring', 'born', 'borrow', 'borrowed',
'boston', 'bot', 'both', 'bots', 'bottle', 'bottles', 'bottom', 'bought', 'bounce', 'bouncing', 'bouncy', 'bound', 'boundaries',
'box', 'boxes', 'boy', 'boys', 'brain', 'brainpop', 'brains', 'brainstorm', 'brainstormed', 'brainstorming', 'brand', 'brave', 'br
eak', 'breakfast', 'breaking', 'breakout', 'breaks', 'breathe', 'breathing', 'bricks', 'bridge', 'bridges', 'bright', 'brighten',
'brighter', 'brightest', 'brilliant', 'bring', 'bringing', 'brings', 'broad', 'broaden', 'broken', 'bronx', 'brooklyn', 'brought',
'brown', 'brushes', 'bubble', 'bubbles', 'bucket', 'buddies', 'budding', 'buddy', 'budget', 'budgets', 'bugs', 'build', 'builder
s', 'building', 'buildings', 'builds', 'built', 'bulletin', 'bullying', 'bunch', 'bundle', 'burden', 'burn', 'burning', 'burstin
g', 'bus', 'business', 'businesses', 'busy', 'but', 'butterflies', 'butterfly', 'button', 'buttons', 'buy', 'buying', 'buzz', 'b
y', 'ca', 'cabinet', 'caddies', 'cafeteria', 'calculate', 'calculator', 'calculators', 'calculus', 'calendar', 'california', 'cal
l', 'called', 'calm', 'calming', 'calories', 'came', 'camera', 'cameras', 'camp', 'campus', 'can', 'candy', 'cannot', 'canvas', 'c
apabilities', 'capability', 'capable', 'capacity', 'capture', 'car', 'card', 'cardboard', 'cards', 'cardstock', 'care', 'cared',
'career', 'careers', 'careful', 'carefully', 'cares', 'caring', 'carolina', 'carpet', 'carpets', 'carry', 'carrying', 'cars', 'car
t', 'cartridges', 'carts', 'case', 'cases', 'cash', 'catch', 'catching', 'category', 'cater', 'caucasian', 'caught', 'cause', 'cau
sed', 'causes', 'causing', 'cd', 'cds', 'cease', 'celebrate', 'celebrated', 'celebrating', 'celebration', 'cell', 'cells', 'cente
r', 'centered', 'centers', 'central', 'century', 'cerebral', 'certain', 'certainly', 'certified', 'chain', 'chair', 'chairs', 'cha
lk', 'challenge', 'challenged', 'challenges', 'challenging', 'chance', 'chances', 'change', 'changed', 'changer', 'changes', 'chan
ging', 'channel', 'chaos', 'chaotic', 'chapter', 'character', 'characteristics', 'characters', 'charge', 'charged', 'charging', 'c
hart', 'charter', 'charts', 'cheap', 'check', 'checked', 'checking', 'cheer', 'chemical', 'chemistry', 'chess', 'chicago', 'chicke
n', 'child', 'childhood', 'children', 'china', 'chinese', 'chips', 'choice', 'choices', 'choir', 'choose', 'choosing', 'chore', 'c
horus', 'chose', 'chosen', 'christmas', 'chrome', 'chromebook', 'chromebooks', 'circle', 'circles', 'circuit', 'circuits', 'circum
stances', 'cities', 'citizen', 'citizens', 'citizenship', 'city', 'civil', 'class', 'classes', 'classic', 'classics', 'classifie
d', 'classmate', 'classmates', 'classroom', 'classrooms', 'classwork', 'clay', 'clean', 'cleaner', 'cleaning', 'clear', 'clearly',
'clever', 'click', 'climate', 'climb', 'climbing', 'clip', 'clipboard', 'clipboards', 'clips', 'clock', 'close', 'closed', 'closel
y', 'closer', 'closet', 'closing', 'clothes', 'clothing', 'club', 'clubs', 'clues', 'clutter', 'co', 'coach', 'coaches', 'coachin
g', 'coast', 'coats', 'code', 'coded', 'codes', 'coding', 'coffee', 'cognitive', 'coins', 'cold', 'collaborate', 'collaborating',
'collaboration', 'collaborative', 'collaboratively', 'collage', 'collar', 'colleagues', 'collect', 'collected', 'collecting', 'col
lection', 'college', 'colleges', 'color', 'colored', 'colorful', 'coloring', 'colors', 'com', 'combat', 'combination', 'combine',
'combined', 'combining', 'come', 'comes', 'comfort', 'comfortable', 'comfortably', 'comfy', 'comic', 'comics', 'coming', 'commen
t', 'comments', 'commitment', 'committed', 'common', 'communicate', 'communicating', 'communication', 'communicators', 'communitie
s', 'community', 'companies', 'company', 'compare', 'compared', 'comparing', 'compass', 'compassion', 'compassionate', 'compete',
'competent', 'competing', 'competition', 'competitions', 'competitive', 'complain', 'complete', 'completed', 'completely', 'comple
ting', 'completion', 'complex', 'complicated', 'compliment', 'component', 'components', 'compose', 'composed', 'composition', 'com
post', 'comprehend', 'comprehension', 'comprehensive', 'comprised', 'computer', 'computers', 'concentrate', 'concentrating', 'conc
entration', 'concept', 'concepts', 'conceptual', 'concern', 'concerned', 'concerns', 'concert', 'concerts', 'conclusions', 'concre
te', 'condition', 'conditioning', 'conditions', 'conducive', 'conduct', 'conducting', 'cones', 'conference', 'conferences', 'confi
dence', 'confident', 'confined', 'conflict', 'conflicts', 'conjunction', 'connect', 'connected', 'connecting', 'connection', 'conn
ections', 'connects', 'conquer', 'conscious', 'conservation', 'consider', 'consideration', 'considered', 'considering', 'consist',
'consistency', 'consistent', 'consistently', 'consisting', 'consists', 'constant', 'constantly', 'constraints', 'construct', 'cons
tructing', 'construction', 'constructive', 'consuming', 'contact', 'contagious', 'contain', 'contained', 'containers', 'contains',
'contemporary', 'content', 'context', 'continually', 'continue', 'continued', 'continues', 'continuing', 'continuous', 'continuous
ly', 'contrast', 'contribute', 'contributing', 'contribution', 'contributions', 'control', 'controlled', 'conversation', 'conversa
```

```
tions', 'cook', 'cookies', 'cooking', 'cool', 'cooperate', 'cooperation', 'cooperative', 'cooperatively', 'coordination', 'cope',
'copier', 'copies', 'coping', 'copy', 'cords', 'core', 'corner', 'correct', 'correctly', 'correlate', 'cost', 'costly', 'costs',
'costumes', 'couch', 'could', 'counseling', 'counselor', 'count', 'counter', 'counters', 'counting', 'countless', 'countries', 'co
untry', 'counts', 'county', 'couple', 'courage', 'course', 'courses', 'court', 'cover', 'covered', 'covering', 'covers', 'cozy',
'crackers', 'craft', 'crafts', 'crates', 'crave', 'craving', 'crayons', 'crazy', 'cream', 'create', 'created', 'creates', 'creatin
g', 'creation', 'creations', 'creative', 'creatively', 'creativity', 'creators', 'credit', 'crew', 'cricut', 'crime', 'crisis', 'c
ritical', 'critically', 'cross', 'crowded', 'crucial', 'cry', 'cubbies', 'cube', 'cubes', 'cues', 'culminating', 'cultivate', 'cul
tural', 'culturally', 'culture', 'cultures', 'cups', 'curiosity', 'curious', 'curl', 'current', 'currently', 'curricular', 'curric
ulum', 'cushion', 'cushions', 'cut', 'cuts', 'cutting', 'cycle', 'cycles', 'dad', 'daily', 'dallas', 'damage', 'damaged', 'dance',
'dances', 'dancing', 'dangerous', 'dark', 'dash', 'data', 'date', 'daunting', 'david', 'day', 'days', 'dc', 'de', 'deaf', 'deal',
'dealing', 'debate', 'debates', 'decades', 'decide', 'decided', 'decision', 'decisions', 'decoding', 'decorate', 'decrease', 'decr
eased', 'dedicated', 'dedication', 'deep', 'deepen', 'deeper', 'deeply', 'deficit', 'deficits', 'define', 'defined', 'definitely',
'degree', 'degrees', 'delay', 'delayed', 'delays', 'deliver', 'delivery', 'dell', 'delve', 'demand', 'demanding', 'demands', 'demo
graphic', 'demographics', 'demonstrate', 'demonstrated', 'demonstrating', 'demonstration', 'demonstrations', 'dense', 'departmen
t', 'depend', 'dependent', 'depending', 'depth', 'describe', 'described', 'describes', 'desert', 'deserve', 'deserves', 'deservin
g', 'design', 'designated', 'designed', 'designers', 'designing', 'designs', 'desire', 'desired', 'desires', 'desk', 'desks', 'des
ktop', 'desktops', 'desperate', 'desperately', 'despite', 'destroyed', 'detail', 'detailed', 'details', 'determination', 'determin
e', 'determined', 'detroit', 'develop', 'developed', 'developing', 'development', 'developmental', 'developmentally', 'develops',
'device', 'devices', 'devoted', 'diagnosed', 'diagnosis', 'diagrams', 'dialogue', 'diary', 'dice', 'dictionaries', 'dictionary',
'did', 'diego', 'diet', 'difference', 'differences', 'different', 'differentiate', 'differentiated', 'differentiating', 'different
iation', 'differently', 'differing', 'difficult', 'difficulties', 'difficulty', 'dig', 'digital', 'digitally', 'diligently', 'dime
nsional', 'dinner', 'diploma', 'dire', 'direct', 'directed', 'direction', 'directions', 'directly', 'dirty', 'disabilities', 'disa
bility', 'disabled', 'disadvantage', 'disadvantaged', 'disadvantages', 'disc', 'discipline', 'disciplines', 'discouraged', 'discov
er', 'discovered', 'discoveries', 'discovering', 'discovery', 'discs', 'discuss', 'discussed', 'discussing', 'discussion', 'discus
sions', 'disorder', 'disorders', 'display', 'displayed', 'displaying', 'displays', 'disposal', 'disrupting', 'disruption', 'disrup
tive', 'dissect', 'dissection', 'distance', 'distract', 'distracted', 'distracting', 'distraction', 'distractions', 'district', 'd
istricts', 'disturbance', 'disturbing', 'dive', 'diverse', 'diversity', 'divide', 'divided', 'dividers', 'division', 'dna', 'do',
'docs', 'doctors', 'document', 'documenting', 'documents', 'does', 'doh', 'doing', 'dojo', 'dollar', 'dollars', 'dolls', 'dominica
n', 'donate', 'donated', 'donating', 'donation', 'donations', 'done', 'donor', 'donors', 'donorschoose', 'door', 'doors', 'dot',
'dots', 'double', 'doubt', 'dough', 'down', 'download', 'downloaded', 'downs', 'downtown', 'dr', 'drama', 'dramatic', 'dramaticall
y', 'drastically', 'draw', 'drawer', 'drawers', 'drawing', 'drawings', 'drawn', 'dream', 'dreamers', 'dreams', 'dress', 'drills',
'drink', 'drinking', 'drinks', 'drive', 'driven', 'drives', 'driving', 'drone', 'drones', 'drop', 'drug', 'drugs', 'drum', 'drummi
ng', 'drums', 'dry', 'drying', 'dual', 'duct', 'due', 'dull', 'durable', 'during', 'duty', 'dvd', 'dvds', 'dynamic', 'dynamics',
'dyslexia', 'each', 'eager', 'eagerly', 'eagerness', 'ear', 'earbuds', 'earlier', 'earliest', 'early', 'earn', 'earned', 'earnin
g', 'ears', 'earth', 'ease', 'easel', 'easels', 'easier', 'easily', 'east', 'eastern', 'easy', 'eat', 'eating', 'ebooks', 'echo',
'eclectic', 'economic', 'economical', 'economically', 'economics', 'economy', 'ecosystem', 'ecosystems', 'ed', 'edge', 'edit', 'ed
iting', 'edu', 'educate', 'educated', 'educating', 'education', 'educational', 'educationally', 'educator', 'educators', 'effect',
'effective', 'effectively', 'effects', 'efficient', 'efficiently', 'effort', 'efforts', 'eggs', 'eight', 'eighth', 'eighty', 'eins
tein', 'either', 'el', 'ela', 'election', 'elective', 'electric', 'electrical', 'electricity', 'electronic', 'electronics', 'eleme
nt', 'elementary', 'elements', 'eleven', 'eligible', 'eliminate', 'ell', 'ells', 'elmo', 'else', 'elsewhere', 'email', 'embark',
'embarrassed', 'embedded', 'embrace', 'embraced', 'embraces', 'embracing', 'emergent', 'emerging', 'emotional', 'emotionally', 'em
otions', 'empathetic', 'empathy', 'emphasis', 'emphasize', 'emphasizes', 'employment', 'empower', 'empowered', 'empowering', 'empo
wers', 'empty', 'enable', 'enables', 'enabling', 'encompasses', 'encounter', 'encourage', 'encouraged', 'encouragement', 'encourag
es', 'encouraging', 'end', 'endeavor', 'endeavors', 'ended', 'ending', 'endless', 'ends', 'endurance', 'endure', 'energetic', 'ene
rgized', 'energy', 'engage', 'engaged', 'engagement', 'engages', 'engaging', 'engineer', 'engineering', 'engineers', 'english', 'e
nhance', 'enhanced', 'enhances', 'enhancing', 'enjoy', 'enjoyable', 'enjoyed', 'enjoying', 'enjoyment', 'enormous', 'enough', 'enr
ich', 'enriched', 'enriching', 'enrichment', 'enrolled', 'enrollment', 'ensemble', 'ensure', 'ensures', 'ensuring', 'enter', 'ente
```

```
red', 'entering', 'entertaining', 'enthusiasm', 'enthusiastic', 'entire', 'entirely', 'entrepreneurs', 'entry', 'environment', 'en
vironmental', 'environments', 'envision', 'epic', 'equal', 'equally', 'equations', 'equip', 'equipment', 'equipped', 'equitable',
'equity', 'era', 'erase', 'erasers', 'error', 'escape', 'ese', 'esl', 'esol', 'especially', 'essay', 'essays', 'essential', 'essen
tials', 'establish', 'established', 'establishing', 'esteem', 'etc', 'ethic', 'ethnic', 'ethnically', 'ethnicities', 'ethnicity',
'evaluate', 'even', 'event', 'events', 'eventually', 'ever', 'every', 'everybody', 'everyday', 'everyone', 'everything', 'everywhe
re', 'evidence', 'evident', 'evolving', 'exact', 'exactly', 'exam', 'examine', 'example', 'examples', 'exams', 'exceed', 'excel',
'excellence', 'excellent', 'excelling', 'except', 'exception', 'exceptional', 'excess', 'exchange', 'excite', 'excited', 'exciteme
nt', 'exciting', 'excuses', 'exercise', 'exercises', 'exercising', 'exhibit', 'exist', 'existing', 'exists', 'expand', 'expanded',
'expanding', 'expect', 'expectation', 'expectations', 'expected', 'expecting', 'expensive', 'experience', 'experienced', 'experien
ces', 'experiencing', 'experiential', 'experiment', 'experimentation', 'experimenting', 'experiments', 'experts', 'explain', 'expl
ained', 'explaining', 'exploration', 'explorations', 'explore', 'explored', 'explorers', 'exploring', 'expo', 'expose', 'exposed',
'exposing', 'exposure', 'express', 'expressed', 'expressing', 'expression', 'expressive', 'extend', 'extended', 'extension', 'exte
nsive', 'external', 'extra', 'extracurricular', 'extraordinary', 'extras', 'extreme', 'extremely', 'eye', 'eyed', 'eyes', 'fabulou
s', 'face', 'faced', 'faces', 'facilitate', 'facilities', 'facility', 'facing', 'fact', 'factor', 'factors', 'facts', 'faculty',
'fail', 'failed', 'failing', 'failure', 'failures', 'fair', 'fairly', 'fairy', 'fall', 'fallen', 'falling', 'falls', 'familiar',
'families', 'family', 'famous', 'fantastic', 'fantasy', 'far', 'farm', 'farming', 'farms', 'farther', 'fascinated', 'fascinating',
'fashion', 'fast', 'faster', 'favorite', 'favorites', 'fear', 'feature', 'features', 'fed', 'federal', 'feed', 'feedback', 'feedin
g', 'feel', 'feeling', 'feelings', 'feels', 'feet', 'fell', 'fellow', 'felt', 'female', 'fewer', 'fiction', 'fictional', 'fidget',
'fidgeting', 'fidgets', 'fidgety', 'field', 'fields', 'fifteen', 'fifth', 'fifty', 'fight', 'fighting', 'figure', 'figures', 'figu
ring', 'file', 'files', 'fill', 'filled', 'filling', 'film', 'films', 'final', 'finally', 'finances', 'financial', 'financially',
'find', 'finding', 'findings', 'fine', 'finger', 'fingers', 'fingertips', 'finish', 'finished', 'finishing', 'fire', 'fires', 'fir
m', 'firmly', 'first', 'firsthand', 'firsties', 'fish', 'fishing', 'fit', 'fitbit', 'fitbits', 'fitness', 'fits', 'fitting', 'fiv
e', 'fix', 'fixed', 'flash', 'flashcards', 'flat', 'flex', 'flexibility', 'flexible', 'flip', 'flood', 'flooded', 'flooding', 'flo
or', 'floors', 'florida', 'flourish', 'flow', 'flower', 'flowers', 'flowing', 'fluency', 'fluent', 'fluently', 'fluorescent', 'fl
y', 'foam', 'focus', 'focused', 'focuses', 'focusing', 'foldables', 'folder', 'folders', 'follow', 'followed', 'following', 'follo
ws', 'food', 'foods', 'foot', 'football', 'footballs', 'for', 'force', 'forced', 'forces', 'forefront', 'foreign', 'forest', 'fore
ver', 'forget', 'forgotten', 'form', 'formal', 'format', 'formation', 'formats', 'formed', 'former', 'forming', 'forms', 'forth',
'fortunate', 'forty', 'forward', 'foster', 'fostered', 'fostering', 'fosters', 'found', 'foundation', 'foundational', 'foundation
s', 'four', 'fourth', 'fraction', 'fractions', 'frame', 'frames', 'francisco', 'franklin', 'free', 'freedom', 'freely', 'french',
'frequency', 'frequent', 'frequently', 'fresh', 'freshman', 'freshmen', 'friday', 'fridays', 'friend', 'friendly', 'friends', 'fri
endship', 'friendships', 'frog', 'from', 'front', 'fruit', 'fruits', 'frustrated', 'frustrating', 'frustration', 'fuel', 'fulfil
l', 'fulfilling', 'full', 'fullest', 'fully', 'fun', 'function', 'functional', 'functioning', 'functions', 'fund', 'fundamental',
'fundamentals', 'funded', 'funding', 'fundraisers', 'fundraising', 'funds', 'funny', 'furniture', 'furthermore', 'future', 'future
s', 'gain', 'gained', 'gaining', 'gains', 'game', 'games', 'gaming', 'gang', 'gangs', 'gap', 'gaps', 'garden', 'gardening', 'garde
ns', 'gas', 'gate', 'gateway', 'gather', 'gathering', 'gave', 'gear', 'geared', 'gears', 'gel', 'gender', 'general', 'generally',
'generate', 'generation', 'generational', 'generations', 'generosity', 'generous', 'genius', 'genre', 'genres', 'genuine', 'genuin
ely', 'geography', 'geometric', 'geometry', 'george', 'georgia', 'german', 'germs', 'get', 'gets', 'getting', 'giant', 'gift', 'gi
fted', 'gifts', 'girl', 'girls', 'give', 'given', 'gives', 'giving', 'glass', 'glasses', 'global', 'globally', 'globe', 'gloves',
'glue', 'go', 'goal', 'goals', 'goes', 'goggles', 'going', 'gold', 'golf', 'gone', 'gonoodle', 'good', 'google', 'got', 'gotten',
'government', 'grab', 'grade', 'grader', 'graders', 'grades', 'graduate', 'graduated', 'graduates', 'graduating', 'graduation', 'g
rammar', 'grandparent', 'grandparents', 'grant', 'granted', 'grants', 'graph', 'graphic', 'graphics', 'graphing', 'graphs', 'gras
p', 'grateful', 'great', 'greater', 'greatest', 'greatly', 'greatness', 'greek', 'green', 'greenhouse', 'greet', 'greeted', 'greet
ing', 'grew', 'grip', 'grit', 'grocery', 'gross', 'ground', 'group', 'grouped', 'grouping', 'groups', 'grow', 'growing', 'grown',
'grows', 'growth', 'guarantee', 'guardians', 'guidance', 'guide', 'guided', 'guides', 'guiding', 'guitar', 'guitars', 'gym', 'habi
t', 'habitat', 'habitats', 'habits', 'hair', 'half', 'hall', 'halls', 'hallway', 'hallways', 'hand', 'handed', 'handful', 'handl
e', 'hands', 'handwriting', 'handy', 'hang', 'hanging', 'happen', 'happened', 'happening', 'happens', 'happier', 'happiness', 'hap
py', 'hard', 'harder', 'hardest', 'hardly', 'hardship', 'hardships', 'hardworking', 'harness', 'harry', 'harsh', 'harvest', 'hat
```

```
e', 'hats', 'have', 'having', 'he', 'head', 'headphone', 'headphones', 'heads', 'headsets', 'health', 'healthier', 'healthy', 'hea
r', 'heard', 'hearing', 'heart', 'hearted', 'hearts', 'heat', 'heavily', 'heavy', 'height', 'heights', 'held', 'hello', 'help', 'h
elped', 'helpers', 'helpful', 'helping', 'helps', 'here', 'heritage', 'hero', 'heroes', 'hi', 'hidden', 'high', 'higher', 'highes
t', 'highlight', 'highlighters', 'highly', 'hilarious', 'hinder', 'hispanic', 'historians', 'historic', 'historical', 'historicall
y', 'history', 'hit', 'hockey', 'hokki', 'hold', 'holding', 'holds', 'hole', 'holes', 'holiday', 'holidays', 'holocaust', 'home',
'homeless', 'homelessness', 'homeroom', 'homes', 'homework', 'hone', 'honest', 'honestly', 'honor', 'honored', 'honors', 'hook',
'hooked', 'hooks', 'hoops', 'hop', 'hope', 'hopeful', 'hopefully', 'hopes', 'hoping', 'horizons', 'hospital', 'host', 'hot', 'hou
r', 'hours', 'house', 'housed', 'household', 'households', 'houses', 'housing', 'houston', 'how', 'however', 'hp', 'hub', 'hug',
'huge', 'hugs', 'hula', 'human', 'humans', 'humble', 'hundred', 'hundreds', 'hunger', 'hungry', 'hurdles', 'hurt', 'hydrated', 'hy
giene', 'hyperactivity', 'ib', 'ice', 'ict', 'idea', 'ideal', 'ideas', 'identification', 'identified', 'identify', 'identifying',
'identity', 'iep', 'ieps', 'if', 'ignite', 'ii', 'illinois', 'illustrate', 'illustrations', 'illustrators', 'image', 'images', 'im
agination', 'imaginations', 'imaginative', 'imagine', 'imagined', 'immediate', 'immediately', 'immense', 'immensely', 'immerse',
'immersed', 'immersion', 'immigrant', 'immigrants', 'immigrated', 'imovie', 'impact', 'impacted', 'impacts', 'impaired', 'impairme
nt', 'impairments', 'imperative', 'implement', 'implementation', 'implemented', 'implementing', 'importance', 'important', 'import
antly', 'impossible', 'impoverished', 'impress', 'impressed', 'improve', 'improved', 'improvement', 'improvements', 'improves', 'i
mproving', 'in', 'inability', 'incentive', 'incentives', 'include', 'included', 'includes', 'including', 'inclusion', 'inclusive',
'income', 'incomes', 'incoming', 'incorporate', 'incorporated', 'incorporates', 'incorporating', 'increase', 'increased', 'increas
es', 'increasing', 'increasingly', 'incredible', 'incredibly', 'independence', 'independent', 'independently', 'india', 'indian',
'indiana', 'indianapolis', 'individual', 'individuality', 'individualize', 'individualized', 'individually', 'individuals', 'indoo
r', 'indoors', 'industry', 'inexpensive', 'influence', 'information', 'informational', 'informative', 'informed', 'initial', 'init
iative', 'injury', 'ink', 'inner', 'innovate', 'innovation', 'innovative', 'innovators', 'input', 'inquire', 'inquiry', 'inquisiti
ve', 'insects', 'inside', 'insight', 'inspiration', 'inspirational', 'inspirations', 'inspire', 'inspired', 'inspires', 'inspirin
g', 'instance', 'instant', 'instantly', 'instead', 'instill', 'instilling', 'instruct', 'instruction', 'instructional', 'instructi
ons', 'instrument', 'instrumental', 'instruments', 'integral', 'integrate', 'integrated', 'integrates', 'integrating', 'integratio
n', 'integrity', 'intellectual', 'intellectually', 'intelligence', 'intelligent', 'intend', 'intended', 'intense', 'intensive', 'i
nteract', 'interacting', 'interaction', 'interactions', 'interactive', 'interdisciplinary', 'interest', 'interested', 'interestin
g', 'interests', 'intermediate', 'international', 'internet', 'interpret', 'intervention', 'interventions', 'into', 'intrigued',
'intrinsic', 'introduce', 'introduced', 'introducing', 'introduction', 'invaluable', 'invent', 'invention', 'inventions', 'invento
rs', 'inventory', 'invest', 'invested', 'investigate', 'investigating', 'investigation', 'investigations', 'investing', 'investmen
t', 'invite', 'invited', 'inviting', 'involve', 'involved', 'involvement', 'involves', 'involving', 'ipad', 'ipads', 'ipod', 'ipod
s', 'iready', 'is', 'island', 'issue', 'issues', 'it', 'item', 'items', 'its', 'ixl', 'jail', 'jazz', 'jersey', 'job', 'jobs', 'jo
hn', 'join', 'joining', 'journal', 'journalism', 'journals', 'journey', 'journeys', 'joy', 'joyful', 'jr', 'jump', 'jumping', 'jun
e', 'junior', 'juniors', 'just', 'justice', 'kahoot', 'kansas', 'keep', 'keeping', 'keeps', 'kentucky', 'kept', 'key', 'keyboard',
'keyboarding', 'keyboards', 'keys', 'khan', 'kick', 'kickball', 'kid', 'kiddos', 'kidney', 'kids', 'kind', 'kinder', 'kindergarte
n', 'kindergarteners', 'kindergartners', 'kinders', 'kindle', 'kindles', 'kindness', 'kinds', 'kinesthetic', 'kinetic', 'king', 'k
it', 'kitchen', 'kits', 'kneeling', 'knees', 'knew', 'knit', 'know', 'knowing', 'knowledge', 'knowledgeable', 'known', 'knows', 'k
ore', 'la', 'lab', 'label', 'labeled', 'labels', 'laboratory', 'labs', 'lack', 'lacking', 'lacks', 'ladies', 'lake', 'lakeshore',
'laminate', 'laminated', 'laminating', 'laminator', 'land', 'language', 'languages', 'lap', 'laptop', 'laptops', 'large', 'largel
y', 'larger', 'largest', 'las', 'last', 'lasting', 'lastly', 'late', 'later', 'latest', 'latin', 'latino', 'laugh', 'laughing', 'l
aughter', 'launch', 'laws', 'lawyers', 'lay', 'laying', 'lead', 'leader', 'leaders', 'leadership', 'leading', 'leads', 'league',
'lean', 'leaning', 'leap', 'learn', 'learned', 'learner', 'learners', 'learning', 'learns', 'least', 'leave', 'leaves', 'leaving',
'lecture', 'lectures', 'led', 'left', 'leg', 'lego', 'legos', 'legs', 'leisure', 'lend', 'length', 'lens', 'less', 'lesson', 'less
ons', 'let', 'lets', 'letter', 'letters', 'letting', 'level', 'leveled', 'levels', 'lexia', 'librarian', 'libraries', 'library',
'lie', 'lies', 'life', 'lifelong', 'lifestyle', 'lifestyles', 'lifetime', 'lift', 'light', 'lighting', 'lights', 'like', 'liked',
'likely', 'likes', 'limit', 'limitations', 'limited', 'limiting', 'limitless', 'limits', 'line', 'lined', 'lines', 'linguistic',
'linguistically', 'link', 'linked', 'links', 'list', 'listed', 'listen', 'listening', 'lists', 'literacy', 'literally', 'literar
y', 'literate', 'literature', 'little', 'littlebits', 'live', 'lived', 'lively', 'lives', 'living', 'load', 'local', 'locally', 'l
```

```
ocate', 'located', 'location', 'locations', 'lock', 'locks', 'log', 'logic', 'logical', 'long', 'longer', 'look', 'looked', 'looki
ng', 'looks', 'looping', 'loose', 'los', 'lose', 'losing', 'loss', 'lost', 'lot', 'lots', 'loud', 'louisiana', 'lovable', 'love',
'loved', 'lovely', 'lovers', 'loves', 'loving', 'low', 'lower', 'lowest', 'lucky', 'lunch', 'lunches', 'luxury', 'macbook', 'machi
ne', 'machines', 'made', 'magazine', 'magazines', 'magic', 'magical', 'magna', 'magnet', 'magnetic', 'magnets', 'mail', 'main', 'm
ainly', 'mainstream', 'maintain', 'maintaining', 'major', 'majority', 'make', 'maker', 'makers', 'makerspace', 'makes', 'makeup',
'makey', 'making', 'male', 'man', 'manage', 'management', 'managing', 'mandarin', 'mandated', 'mandela', 'manipulate', 'manipulati
ng', 'manipulative', 'manipulatives', 'manner', 'manners', 'many', 'map', 'maps', 'march', 'marching', 'mark', 'marker', 'marker
s', 'market', 'martin', 'massachusetts', 'master', 'mastered', 'mastering', 'masterpiece', 'masterpieces', 'masters', 'mastery',
'mat', 'match', 'matching', 'material', 'materials', 'math', 'mathematical', 'mathematicians', 'mathematics', 'mats', 'matter', 'm
atters', 'mature', 'max', 'maximize', 'maximum', 'may', 'maybe', 'me', 'meal', 'meals', 'mean', 'meaning', 'meaningful', 'means',
'meant', 'measure', 'measurement', 'measuring', 'mechanical', 'media', 'medical', 'medicine', 'meditation', 'medium', 'mediums',
'meet', 'meeting', 'meetings', 'meets', 'melting', 'member', 'members', 'memorable', 'memories', 'memory', 'men', 'mental', 'menta
lly', 'mention', 'mentioned', 'mentor', 'mentors', 'mess', 'message', 'messages', 'messy', 'met', 'metabolism', 'metal', 'method',
'methods', 'mexico', 'miami', 'mice', 'michigan', 'microphone', 'microphones', 'microscope', 'microscopes', 'microsoft', 'mid', 'm
iddle', 'might', 'mighty', 'migrant', 'mild', 'mile', 'miles', 'military', 'million', 'milwaukee', 'mind', 'minded', 'mindful', 'm
indfulness', 'minds', 'mindset', 'mine', 'minecraft', 'mini', 'minimal', 'minimize', 'minimum', 'minis', 'minnesota', 'minoritie
s', 'minority', 'minute', 'minutes', 'miss', 'missed', 'missing', 'mission', 'mississippi', 'missouri', 'mistakes', 'mix', 'mixe
d', 'mixing', 'mixture', 'mo', 'mobile', 'mobility', 'modalities', 'model', 'modeled', 'modeling', 'models', 'moderate', 'modern',
'modes', 'modifications', 'modified', 'mold', 'mom', 'moment', 'moments', 'momentum', 'monday', 'money', 'monitor', 'monitoring',
'monitors', 'montessori', 'month', 'monthly', 'months', 'mood', 'moon', 'more', 'moreover', 'morning', 'mornings', 'most', 'mostl
y', 'mother', 'motion', 'motivate', 'motivated', 'motivates', 'motivating', 'motivation', 'motivational', 'motivator', 'motor', 'm
otto', 'mountain', 'mountains', 'mouse', 'move', 'moved', 'movement', 'movements', 'movers', 'moves', 'movie', 'movies', 'moving',
'mr', 'mrs', 'ms', 'much', 'multi', 'multicultural', 'multimedia', 'multiple', 'multiplication', 'multitude', 'muscle', 'muscles',
'museum', 'museums', 'music', 'musical', 'musicians', 'must', 'my', 'myriad', 'mystery', 'name', 'named', 'names', 'nannan21st',
'nannana', 'nannanactive', 'nannanall', 'nannanalternative', 'nannanan', 'nannanart', 'nannanbook', 'nannanbooks', 'nannanbouncin
g', 'nannanbringing', 'nannanbuilding', 'nannancan', 'nannanchrome', 'nannanchromebook', 'nannanchromebooks', 'nannanclassroom',
'nannancoding', 'nannancozy', 'nannancreating', 'nannancreative', 'nannando', 'nannanempowering', 'nannanengaging', 'nannanexplori
ng', 'nannanextra', 'nannanfirst', 'nannanflexible', 'nannanfull', 'nannanfun', 'nannanfuture', 'nannanget', 'nannangetting', 'nan
nangrowing', 'nannanhands', 'nannanheadphones', 'nannanhealthy', 'nannanhelp', 'nannani', 'nannaninteractive', 'nannanipad', 'nann
anipads', 'nannanit', 'nannankeep', 'nannankeeping', 'nannankindergarten', 'nannanlearning', 'nannanlet', 'nannanlife', 'nannanlis
ten', 'nannanlistening', 'nannanliteracy', 'nannanlittle', 'nannanmake', 'nannanmaking', 'nannanmath', 'nannanmore', 'nannanmovin
g', 'nannanmrs', 'nannanmusic', 'nannanmy', 'nannannew', 'nannanoh', 'nannanon', 'nannanorganization', 'nannanour', 'nannanpleas
e', 'nannanproject', 'nannanread', 'nannanreading', 'nannanready', 'nannanschool', 'nannanscience', 'nannansensory', 'nannanspecia
l', 'nannanstand', 'nannansteam', 'nannanstem', 'nannanstudent', 'nannanstudents', 'nannansuper', 'nannansupplies', 'nannantablet
s', 'nannantake', 'nannanteaching', 'nannantech', 'nannantechnology', 'nannanthe', 'nannantime', 'nannanusing', 'nannanwe', 'nanna
nwhat', 'nannanwiggle', 'nannanwobble', 'nannanwriting', 'narrative', 'narratives', 'nation', 'national', 'nationalities', 'nativ
e', 'natives', 'natural', 'naturally', 'nature', 'navigate', 'navigating', 'nc', 'near', 'nearby', 'nearly', 'neat', 'necessaril
y', 'necessary', 'necessities', 'necessity', 'need', 'needed', 'needing', 'needless', 'needs', 'negative', 'neighborhood', 'neighb
orhoods', 'neighboring', 'neighbors', 'nelson', 'nervous', 'nestled', 'net', 'nets', 'nevada', 'never', 'new', 'newcomers', 'newe
r', 'newest', 'newly', 'news', 'newspaper', 'nex', 'next', 'ngss', 'nice', 'night', 'nights', 'nine', 'ninety', 'ninth', 'no', 'no
ise', 'noisy', 'non', 'none', 'nonfiction', 'nonverbal', 'noodle', 'nook', 'nor', 'norm', 'normal', 'normally', 'north', 'northeas
t', 'northern', 'northwest', 'not', 'note', 'notebook', 'notebooks', 'notes', 'nothing', 'notice', 'noticed', 'novel', 'novels',
'now', 'number', 'numbers', 'numerous', 'nurses', 'nurture', 'nurtured', 'nurturing', 'nutrient', 'nutrition', 'nutritional', 'nut
ritious', 'ny', 'nyc', 'oakland', 'obesity', 'object', 'objective', 'objectives', 'objects', 'observation', 'observations', 'obser
ve', 'observed', 'observing', 'obstacle', 'obstacles', 'obtain', 'obtaining', 'occasion', 'occupational', 'occur', 'occurs', 'ocea
n', 'odds', 'of', 'off', 'offer', 'offered', 'offering', 'offers', 'office', 'often', 'oftentimes', 'oh', 'ohio', 'oil', 'oils',
'ok', 'okay', 'oklahoma', 'old', 'older', 'olds', 'on', 'once', 'one', 'ones', 'ongoing', 'online', 'only', 'onto', 'open', 'opene
```

```
d', 'opening', 'opens', 'operate', 'operating', 'operations', 'opinion', 'opinions', 'opportunities', 'opportunity', 'opposed', 'o
ptimal', 'optimistic', 'optimize', 'option', 'options', 'or', 'oral', 'orchestra', 'order', 'ordered', 'ordinary', 'oregon', 'orf
f', 'org', 'organic', 'organisms', 'organization', 'organizational', 'organizations', 'organize', 'organized', 'organizer', 'organ
izers', 'organizing', 'oriented', 'original', 'osmo', 'osmos', 'other', 'others', 'otherwise', 'our', 'out', 'outcome', 'outcome
s', 'outdated', 'outdoor', 'outdoors', 'outgoing', 'outlet', 'outlets', 'outlook', 'outs', 'outside', 'outstanding', 'over', 'over
all', 'overcome', 'overcoming', 'overlooked', 'overwhelmed', 'overwhelming', 'owl', 'own', 'ownership', 'oxygen', 'ozobot', 'ozobo
ts', 'pace', 'paced', 'pacific', 'pack', 'packed', 'packs', 'pad', 'pads', 'page', 'pages', 'pain', 'paint', 'painted', 'paintin
g', 'paintings', 'paints', 'pair', 'paired', 'pairs', 'palsy', 'pantry', 'pants', 'paper', 'paperless', 'papers', 'parachute', 'pa
rent', 'parental', 'parents', 'park', 'parks', 'part', 'participants', 'participate', 'participated', 'participates', 'participati
ng', 'participation', 'particular', 'particularly', 'partitions', 'partner', 'partners', 'partnership', 'partnerships', 'parts',
'party', 'pass', 'passages', 'passed', 'passing', 'passion', 'passionate', 'passions', 'past', 'pastels', 'path', 'paths', 'pathwa
y', 'pathways', 'patience', 'pattern', 'patterns', 'pay', 'paying', 'pbis', 'pbl', 'pe', 'peace', 'peaceful', 'peak', 'pedal', 'pe
dometers', 'peer', 'peers', 'pen', 'pencil', 'pencils', 'pennsylvania', 'pens', 'people', 'per', 'percent', 'percentage', 'percuss
ion', 'perfect', 'perfectly', 'perform', 'performance', 'performances', 'performed', 'performing', 'perhaps', 'period', 'periods',
'permanent', 'perseverance', 'persevere', 'persistence', 'person', 'personal', 'personalities', 'personality', 'personalize', 'per
sonalized', 'personally', 'perspective', 'perspectives', 'pet', 'phenomenal', 'philadelphia', 'philosophy', 'phone', 'phonemic',
'phones', 'phonics', 'photo', 'photographs', 'photography', 'photos', 'phrases', 'physical', 'physically', 'physics', 'piano', 'pi
ck', 'picked', 'picking', 'picture', 'pictures', 'piece', 'pieces', 'pillow', 'pillows', 'pilot', 'pitch', 'pivotal', 'pizza', 'p
k', 'place', 'placed', 'placement', 'places', 'placing', 'plain', 'plan', 'planet', 'planned', 'planning', 'plans', 'plant', 'plan
ting', 'plants', 'plastic', 'plate', 'plates', 'platform', 'play', 'played', 'player', 'players', 'playful', 'playground', 'playin
g', 'plays', 'please', 'pleasure', 'plenty', 'plethora', 'plot', 'plus', 'pocket', 'pockets', 'poems', 'poetry', 'point', 'point
s', 'police', 'political', 'poor', 'poorest', 'pop', 'popcorn', 'popular', 'population', 'populations', 'portable', 'portfolio',
'portfolios', 'portion', 'poses', 'position', 'positions', 'positive', 'positively', 'possess', 'possibilities', 'possibility', 'p
ossible', 'possibly', 'post', 'poster', 'posters', 'posture', 'pot', 'potential', 'potentially', 'potter', 'pouches', 'poverty',
'power', 'powerful', 'powerpoint', 'practical', 'practice', 'practiced', 'practices', 'practicing', 'praise', 'pre', 'precious',
'predict', 'predictions', 'predominantly', 'predominately', 'prefer', 'preferences', 'preferred', 'prek', 'prep', 'preparation',
'prepare', 'prepared', 'prepares', 'preparing', 'preschool', 'preschoolers', 'presence', 'present', 'presentation', 'presentation
s', 'presented', 'presenting', 'presents', 'preserve', 'president', 'press', 'pressure', 'pretend', 'pretty', 'prevent', 'prevent
s', 'previous', 'previously', 'price', 'priced', 'priceless', 'prices', 'pride', 'primarily', 'primary', 'principal', 'principle
s', 'print', 'printed', 'printer', 'printers', 'printing', 'printmaking', 'prints', 'prior', 'priority', 'privacy', 'private', 'pr
ivilege', 'privileged', 'prize', 'prizes', 'pro', 'proactive', 'probably', 'problem', 'problems', 'procedures', 'process', 'proces
ses', 'processing', 'produce', 'produced', 'producing', 'product', 'production', 'productive', 'productivity', 'products', 'profes
sion', 'professional', 'professionals', 'proficiency', 'proficient', 'profound', 'program', 'programming', 'programs', 'progress',
'progresses', 'project', 'projected', 'projector', 'projects', 'promethean', 'promise', 'promote', 'promotes', 'promoting', 'promp
ts', 'propel', 'proper', 'properly', 'properties', 'proposal', 'props', 'protect', 'protected', 'protection', 'protective', 'prote
ctors', 'proud', 'proudly', 'prove', 'proven', 'proves', 'provide', 'provided', 'provides', 'providing', 'provoking', 'pta', 'publ
ic', 'publish', 'published', 'publishing', 'puerto', 'pull', 'pulled', 'pulling', 'pupils', 'puppet', 'puppets', 'purchase', 'purc
hased', 'purchasing', 'purpose', 'purposeful', 'purposes', 'pursue', 'pursuing', 'push', 'pushed', 'pushing', 'put', 'puts', 'putt
ing', 'puzzle', 'puzzles', 'qr', 'qualified', 'qualifies', 'qualify', 'qualifying', 'qualities', 'quality', 'quarter', 'quest', 'q
uestion', 'questioners', 'questioning', 'questions', 'quick', 'quicker', 'quickly', 'quiet', 'quietly', 'quite', 'quiz', 'quizze
s', 'quote', 'quotes', 'race', 'races', 'racial', 'racially', 'rack', 'rain', 'rainbow', 'rainy', 'raise', 'raised', 'raising', 'r
ange', 'ranges', 'ranging', 'rapidly', 'rare', 'rarely', 'rate', 'rates', 'rather', 'ratio', 'raz', 'reach', 'reached', 'reaches',
'reaching', 'reaction', 'reactions', 'read', 'reader', 'readers', 'readily', 'readiness', 'reading', 'readings', 'reads', 'ready',
'real', 'realistic', 'reality', 'realize', 'realized', 'realizing', 'really', 'reason', 'reasoning', 'reasons', 'rebuild', 'receiv
e', 'received', 'receives', 'receiving', 'recent', 'recently', 'receptive', 'recess', 'recipe', 'recipes', 'recognition', 'recogni
ze', 'recognized', 'recognizing', 'recommended', 'record', 'recorded', 'recorder', 'recorders', 'recording', 'recordings', 'recrea
te', 'recycle', 'recycled', 'recycling', 'red', 'redesign', 'reduce', 'reduced', 'reducing', 'reeds', 'refer', 'reference', 'refin
```

```
e', 'reflect', 'reflected', 'reflection', 'reflective', 'reflects', 'refrigerator', 'refugee', 'refugees', 'refuse', 'regarding',
'regardless', 'regards', 'region', 'regular', 'regularly', 'regulate', 'regulation', 'reinforce', 'reinforcement', 'reinforcing',
'relatable', 'relate', 'related', 'relates', 'relationship', 'relationships', 'relatively', 'relatives', 'relax', 'relaxed', 'rela
xing', 'release', 'relevant', 'reliable', 'relief', 'relieve', 'religions', 'reluctant', 'rely', 'remain', 'remainder', 'remainin
g', 'remains', 'remarkable', 'remediation', 'remember', 'remind', 'reminder', 'reminders', 'remote', 'remove', 'removed', 'renaiss
ance', 'rent', 'repair', 'repeat', 'repeated', 'repetition', 'replace', 'replaced', 'replacement', 'replacing', 'reply', 'report',
'reports', 'represent', 'representation', 'representations', 'represented', 'representing', 'represents', 'request', 'requested',
'requesting', 'requests', 'require', 'required', 'requirement', 'requirements', 'requires', 'requiring', 'rescue', 'research', 're
searched', 'researchers', 'researching', 'reservation', 'reside', 'residents', 'resilience', 'resilient', 'resistance', 'resourc
e', 'resourceful', 'resources', 'respect', 'respected', 'respectful', 'respond', 'responding', 'response', 'responses', 'responsib
ilities', 'responsibility', 'responsible', 'responsive', 'rest', 'restless', 'restricted', 'result', 'resulting', 'results', 'reta
in', 'retell', 'retelling', 'retention', 'return', 'returning', 'reusable', 'reuse', 'review', 'reviewing', 'reviews', 'revise',
'reward', 'rewarded', 'rewarding', 'rewards', 'rhyming', 'rhythm', 'rhythms', 'ribbon', 'rich', 'rid', 'ride', 'riding', 'right',
'rights', 'rigor', 'rigorous', 'rings', 'rise', 'rising', 'risk', 'risks', 'river', 'road', 'robot', 'robotic', 'robotics', 'robot
s', 'rock', 'rockets', 'rocking', 'rocks', 'role', 'roles', 'roll', 'rolling', 'roof', 'room', 'rooms', 'rope', 'ropes', 'rotate',
'rotating', 'rotation', 'rotations', 'rough', 'roughly', 'round', 'rounded', 'routine', 'routines', 'row', 'rows', 'rug', 'rugs',
'rules', 'run', 'running', 'runs', 'rural', 'rush', 'sacks', 'sad', 'sadly', 'safe', 'safely', 'safer', 'safety', 'said', 'salt',
'salvador', 'sample', 'samples', 'san', 'sand', 'sanitizer', 'sat', 'save', 'saved', 'saving', 'savvy', 'saw', 'say', 'saying', 's
ays', 'sc', 'scale', 'scan', 'scanner', 'scarce', 'scared', 'scary', 'scenarios', 'scene', 'scenes', 'schedule', 'scheduled', 'sch
edules', 'scholar', 'scholars', 'scholarships', 'scholastic', 'school', 'schoolers', 'schooling', 'schools', 'science', 'science
s', 'scientific', 'scientist', 'scientists', 'scissors', 'scope', 'score', 'scores', 'scratch', 'screen', 'screens', 'sculpture',
'sculptures', 'sea', 'search', 'searching', 'season', 'seasons', 'seat', 'seated', 'seating', 'seats', 'second', 'secondary', 'sec
ret', 'section', 'sections', 'secure', 'security', 'sedentary', 'see', 'seed', 'seeds', 'seeing', 'seek', 'seeking', 'seem', 'seem
s', 'seen', 'seesaw', 'select', 'selected', 'selecting', 'selection', 'selections', 'self', 'sell', 'selves', 'semester', 'send',
'sending', 'senior', 'seniors', 'sense', 'senses', 'sensitive', 'sensory', 'sent', 'sentence', 'sentences', 'separate', 'septembe
r', 'sequence', 'sequencing', 'series', 'serious', 'seriously', 'serve', 'served', 'serves', 'service', 'services', 'serving', 'se
s', 'session', 'sessions', 'set', 'setbacks', 'sets', 'setting', 'settings', 'settle', 'setup', 'seuss', 'seven', 'seventh', 'seve
nty', 'several', 'severe', 'severely', 'sewing', 'shake', 'shakespeare', 'shape', 'shaped', 'shapes', 'share', 'shared', 'sharin
g', 'sharp', 'sharpen', 'sharpened', 'sharpener', 'sharpeners', 'she', 'sheet', 'sheets', 'shelf', 'shelter', 'shelters', 'shelve
s', 'shift', 'shine', 'shirt', 'shirts', 'shoes', 'shoot', 'shop', 'shopping', 'short', 'shortage', 'show', 'showcase', 'showed',
'showing', 'shown', 'shows', 'shy', 'siblings', 'sick', 'side', 'sides', 'sight', 'sign', 'significant', 'significantly', 'signs',
'silent', 'silly', 'similar', 'simple', 'simply', 'simulations', 'simultaneously', 'since', 'sing', 'singers', 'singing', 'singl
e', 'sink', 'sit', 'site', 'sites', 'sits', 'sitting', 'situated', 'situation', 'situations', 'six', 'sixth', 'sixty', 'size', 'si
zed', 'sizes', 'skill', 'skills', 'skin', 'sky', 'sleep', 'slide', 'slides', 'slightly', 'slow', 'slowly', 'small', 'smaller', 'sm
allest', 'smart', 'smartboard', 'smarter', 'smell', 'smile', 'smiles', 'smiling', 'smooth', 'smoothly', 'snack', 'snacks', 'snap',
'snow', 'so', 'soak', 'soap', 'soar', 'soccer', 'social', 'socialization', 'socialize', 'socially', 'society', 'socio', 'socioecon
omic', 'socioeconomically', 'socks', 'soft', 'softball', 'software', 'soil', 'solar', 'solely', 'solid', 'solidify', 'solution',
'solutions', 'solve', 'solved', 'solvers', 'solving', 'some', 'someone', 'something', 'sometimes', 'somewhere', 'song', 'songs',
'soon', 'sort', 'sorting', 'sorts', 'soul', 'sound', 'sounds', 'source', 'sources', 'south', 'southern', 'southwest', 'space', 'sp
aces', 'span', 'spanish', 'spans', 'spark', 'sparks', 'spatial', 'speak', 'speaker', 'speakers', 'speaking', 'speaks', 'special',
'specialist', 'specialized', 'specially', 'specials', 'species', 'specific', 'specifically', 'spectrum', 'speech', 'speed', 'spel
l', 'spelling', 'spend', 'spending', 'spent', 'sphero', 'spin', 'spirit', 'spirited', 'spirits', 'spite', 'split', 'spoken', 'spon
ges', 'spontaneous', 'sport', 'sports', 'sportsmanship', 'spot', 'spots', 'spread', 'spring', 'square', 'squares', 'st', 'stabilit
y', 'stable', 'stacked', 'staff', 'stage', 'stages', 'stakes', 'stamina', 'stamp', 'stamps', 'stand', 'standard', 'standardized',
'standards', 'standing', 'stands', 'star', 'starbucks', 'stars', 'start', 'started', 'starter', 'starting', 'starts', 'state', 'st
ated', 'statement', 'states', 'station', 'stationary', 'stations', 'statistics', 'status', 'statuses', 'stay', 'staying', 'stays',
'steady', 'steam', 'stem', 'step', 'stepping', 'steps', 'stick', 'stickers', 'sticks', 'sticky', 'still', 'stimulate', 'stimulate
```

```
d', 'stimulating', 'stimulation', 'stock', 'stocked', 'stomach', 'stone', 'stool', 'stools', 'stop', 'stopped', 'storage', 'stor
e', 'stored', 'stores', 'stories', 'story', 'storytelling', 'storyworks', 'straight', 'strategic', 'strategies', 'strategy', 'stre
am', 'street', 'streets', 'strength', 'strengthen', 'strengthening', 'strengths', 'stress', 'stressed', 'stressful', 'stretch', 's
tretching', 'stricken', 'strides', 'string', 'strings', 'strips', 'strive', 'strives', 'striving', 'strong', 'stronger', 'strongl
y', 'structure', 'structured', 'structures', 'struggle', 'struggled', 'struggles', 'struggling', 'stuck', 'student', 'students',
'studied', 'studies', 'studio', 'study', 'studying', 'stuff', 'sturdy', 'style', 'styles', 'subject', 'subjects', 'submit', 'subsc
ription', 'subscriptions', 'subtract', 'subtracting', 'subtraction', 'suburb', 'suburban', 'succeed', 'succeeding', 'success', 'su
ccesses', 'successful', 'successfully', 'such', 'suffer', 'sufficient', 'sugar', 'suggested', 'suggestions', 'suit', 'suitable',
'suited', 'suits', 'summer', 'sun', 'sunshine', 'super', 'superheroes', 'supplement', 'supplemental', 'supplied', 'supplies', 'sup
ply', 'supplying', 'support', 'supported', 'supporting', 'supportive', 'supports', 'supposed', 'sure', 'surely', 'surface', 'surpr
ise', 'surprised', 'surrounded', 'surrounding', 'surroundings', 'surrounds', 'survival', 'survive', 'sustain', 'sustainable', 'sus
tained', 'sweet', 'sweetest', 'swing', 'switch', 'symbols', 'syndrome', 'system', 'systems', 'table', 'tables', 'tablet', 'tablet
s', 'tackle', 'tactile', 'tag', 'tags', 'tailored', 'take', 'taken', 'takers', 'takes', 'taking', 'tale', 'talent', 'talented', 't
alents', 'tales', 'talk', 'talked', 'talkers', 'talking', 'talks', 'tall', 'tangible', 'tap', 'tape', 'tapping', 'target', 'target
ed', 'targets', 'task', 'tasks', 'taste', 'taught', 'teach', 'teacher', 'teachers', 'teaches', 'teaching', 'team', 'teams', 'teamw
ork', 'tear', 'tears', 'tech', 'technical', 'technique', 'techniques', 'technological', 'technologically', 'technologies', 'techno
logy', 'teen', 'teenage', 'teenagers', 'teens', 'teeth', 'television', 'tell', 'telling', 'tells', 'temperature', 'temporary', 'te
n', 'tend', 'tends', 'tennessee', 'tennis', 'term', 'terms', 'terrific', 'test', 'tested', 'testing', 'tests', 'texas', 'text', 't
extbook', 'textbooks', 'texts', 'thank', 'thankful', 'thanks', 'that', 'the', 'theater', 'theatre', 'their', 'them', 'theme', 'the
med', 'themes', 'then', 'theory', 'therapeutic', 'therapist', 'therapy', 'there', 'thereby', 'therefore', 'these', 'they', 'thin',
'thing', 'things', 'think', 'thinkers', 'thinking', 'third', 'thirst', 'thirsty', 'thirty', 'this', 'thoroughly', 'those', 'thoug
h', 'thought', 'thoughtful', 'thoughts', 'thousand', 'thousands', 'three', 'thrilled', 'thrive', 'thriving', 'through', 'throughou
t', 'throw', 'throwing', 'thus', 'ti', 'tie', 'tied', 'ties', 'tight', 'tile', 'tiles', 'time', 'timely', 'timer', 'timers', 'time
s', 'tinker', 'tiny', 'tip', 'tips', 'tired', 'tirelessly', 'tissue', 'tissues', 'title', 'titles', 'tk', 'to', 'today', 'toes',
'together', 'told', 'tolerance', 'tomorrow', 'ton', 'tone', 'toner', 'tons', 'too', 'took', 'tool', 'tools', 'toon', 'top', 'topi
c', 'topics', 'torn', 'toss', 'total', 'totally', 'touch', 'touches', 'touching', 'tough', 'toward', 'towards', 'town', 'towns',
'toy', 'toys', 'track', 'trackers', 'tracking', 'tracks', 'trade', 'tradition', 'traditional', 'traditionally', 'traditions', 'tra
in', 'trained', 'training', 'traits', 'transfer', 'transferred', 'transform', 'transformation', 'transformed', 'transforming', 'tr
ansient', 'transition', 'transitional', 'transitioning', 'transitions', 'translate', 'transport', 'transportation', 'trash', 'trau
ma', 'traumatic', 'travel', 'traveled', 'travelers', 'traveling', 'trays', 'treasure', 'treat', 'treated', 'treats', 'tree', 'tree
s', 'tremendous', 'tremendously', 'trend', 'trial', 'trials', 'tricky', 'tried', 'tries', 'trip', 'trips', 'trouble', 'true', 'tru
ly', 'trust', 'truth', 'try', 'trying', 'tubes', 'tubs', 'tune', 'turn', 'turned', 'turning', 'turns', 'tutorials', 'tutoring', 't
v', 'twelve', 'twenty', 'twice', 'twist', 'two', 'type', 'types', 'typical', 'typically', 'typing', 'ukulele', 'ukuleles', 'ultima
te', 'ultimately', 'unable', 'uncomfortable', 'uncommon', 'underfunded', 'underprivileged', 'understand', 'understanding', 'unders
tandings', 'understood', 'unfamiliar', 'unfortunate', 'unfortunately', 'unhealthy', 'uniform', 'uniforms', 'unique', 'uniquely',
'uniqueness', 'unit', 'united', 'units', 'universal', 'universe', 'university', 'unknown', 'unless', 'unlike', 'unlimited', 'unloc
k', 'unprepared', 'unstable', 'up', 'upcoming', 'update', 'updated', 'upgrade', 'upload', 'upon', 'upper', 'ups', 'upset', 'urba
n', 'us', 'usage', 'use', 'used', 'useful', 'user', 'uses', 'using', 'usually', 'utah', 'utensils', 'utilize', 'utilized', 'utiliz
ing', 'utmost', 'vacuum', 'valley', 'valuable', 'value', 'valued', 'values', 'varied', 'varies', 'variety', 'various', 'vary', 'va
rying', 'vast', 'vastly', 'vegas', 'vegetable', 'vegetables', 'vehicle', 'velcro', 'venture', 'verbal', 'verbally', 'versatile',
'version', 'versions', 'versus', 'very', 'via', 'vibrant', 'video', 'videos', 'vietnam', 'vietnamese', 'view', 'viewing', 'views',
'village', 'violence', 'violin', 'violins', 'virginia', 'virtual', 'virtually', 'visible', 'vision', 'visit', 'visiting', 'visit
s', 'visual', 'visualize', 'visually', 'visuals', 'vital', 'vivid', 'vocabulary', 'vocal', 'vocational', 'voice', 'voices', 'volle
yball', 'volleyballs', 'volume', 'volunteer', 'volunteers', 'voracious', 'wait', 'waiting', 'wake', 'walk', 'walked', 'walking',
'walks', 'wall', 'walls', 'want', 'wanted', 'wanting', 'wants', 'war', 'warm', 'was', 'washington', 'waste', 'wasted', 'watch', 'w
atched', 'watching', 'water', 'watercolor', 'waves', 'way', 'ways', 'we', 'weaknesses', 'wealth', 'wealthy', 'weapon', 'wear', 'we
aring', 'weather', 'web', 'website', 'websites', 'week', 'weekend', 'weekends', 'weekly', 'weeks', 'weight', 'weighted', 'weight
```

```
s', 'welcome', 'welcomed', 'welcoming', 'well', 'wellness', 'went', 'west', 'western', 'wet', 'what', 'whatever', 'wheel', 'wheel
s', 'when', 'whenever', 'where', 'wherever', 'whether', 'which', 'while', 'whisper', 'white', 'whiteboard', 'whiteboards', 'who',
'whole', 'whose', 'why', 'wide', 'wider', 'wiggle', 'wiggles', 'wiggling', 'wiggly', 'wild', 'will', 'willing', 'willingness', 'wi
n', 'wind', 'window', 'windows', 'winning', 'winter', 'wipe', 'wipes', 'wire', 'wireless', 'wires', 'wise', 'wish', 'with', 'withi
n', 'without', 'witness', 'witnessed', 'wobble', 'wobbly', 'women', 'wonder', 'wonderful', 'wonderfully', 'wondering', 'wonders',
'wood', 'wooden', 'word', 'words', 'work', 'workbooks', 'worked', 'worker', 'workers', 'workforce', 'working', 'workout', 'workpla
ce', 'works', 'worksheet', 'worksheets', 'workshop', 'workshops', 'workspace', 'workstations', 'world', 'worlds', 'worms', 'worn',
'worried', 'worries', 'worry', 'worrying', 'worse', 'worth', 'worthwhile', 'worthy', 'would', 'wow', 'write', 'writer', 'writers',
'writing', 'written', 'wrong', 'wrote', 'www', 'xylophones', 'yard', 'year', 'yearbook', 'yearn', 'yearning', 'years', 'yes', 'yes
terday', 'yet', 'yoga', 'york', 'you', 'young', 'younger', 'youngest', 'your', 'youth', 'youtube', 'zest', 'zip', 'zone', 'zones',
'zoo']
==========================================================================================
```

IDF and wrod (Features name) for AVG W2V

Taking words, so later we can use it for deriving vectorize of both essay and title.

In [66]:

```
print(Tfidf_vectorizer.idf_)

df_idf = pd.DataFrame(Tfidf_vectorizer.idf_, index=tf,columns=["tf_idf_weights"])
df_idf_sort_desc=df_idf.sort_values(by=["tf_idf_weights"],ascending=False)
#df_idf_sort_desc=df_idf_sort_desc
df_idf_sort_desc_2k=df_idf_sort_desc[:2000]
df_idf_sort_desc_2k#
```

```
[7.26044748 5.99879557 4.52668893 ... 6.40205073 7.56918297 7.27619584]
```

Out[66]:

	tf_idf_weights
archery	8.474892
german	8.328288
violins	8.241277
dell	8.161234
golf	8.018133
bot	8.018133
waves	7.985343
violin	7.985343
echo	7.953595

	tf_idf_weights
hockey	7.953595
minecraft	7.922823
drones	7.922823
calculus	7.863983
chicken	7.863983
oils	7.835812
superheroes	7.835812
fluorescent	7.835812
dojo	7.808413
scanner	7.808413
volleyballs	7.781744
bots	7.781744
whisper	7.781744
printmaking	7.781744
reeds	7.755769
partitions	7.755769
easels	7.755769
bees	7.755769
cricut	7.730451
holocaust	7.730451
drawer	7.730451
...	...
carts	6.567300
split	6.567300

	tf_idf_weights
photo	6.567300
workout	6.567300
mentioned	6.567300
sportsmanship	6.567300
avoid	6.567300
incentives	6.567300
finger	6.567300
fairly	6.567300
published	6.567300
dramatically	6.567300
weaknesses	6.559518
700	6.559518
nannanart	6.559518
grows	6.559518
island	6.559518
hinder	6.559518
keyboarding	6.559518
97	6.559518
fairy	6.559518
exam	6.559518
tales	6.551796
hug	6.551796
existing	6.551796
passing	6.551796

	tf_idf_weights
strongly	6.551796
mention	6.551796
jumping	6.551796
150	6.551796

2000 rows × 1 columns

In [67]:
```
print(type(Tfidf_vectorizer.idf_))
print(Tfidf_vectorizer.idf_)

# argsort() will return the indices of values from low to high.
# When you print feature names of these indices, these indices will return you the feature names with low probability.
# So, please reverse the indices after argsort()

tf_sorted_Asc=Tfidf_vectorizer.idf_.argsort()
print(tf_sorted_Asc)
tf_sorted_desc=tf_sorted_Asc[::-1]
print(tf_sorted_desc)

# https://cmdlinetips.com/2018/01/how-to-create-pandas-dataframe-from-multiple-lists/
TFIDF_Feature_IDX_dataFrame=pd.DataFrame({'Feature_Word': tf,'Feature_index' : tf_sorted_desc})

# https://cmdlinetips.#com/2018/02/how-to-sort-pandas-dataframe-by-columns-and-row/
TFIDF_Feature_IDX_dataFrame_sorted=TFIDF_Feature_IDX_dataFrame.sort_values('Feature_index',ascending=False)
TFIDF_Feature_IDX_dataFrame_sorted.head(11)

TFIDF_Feature_2K=TFIDF_Feature_IDX_dataFrame_sorted[:2000]
print(type(TFIDF_Feature_2K))
TFIDF_Feature_2K

#TFIDF_Feature_40=TFIDF_Feature_IDX_dataFrame_sorted[:40]
#print(type(TFIDF_Feature_40))
#TFIDF_Feature_40
```

```
<class 'numpy.ndarray'>
[7.26044748 5.99879557 4.52668893 ... 6.40205073 7.56918297 7.27619584]
[4357 3962 2953 ... 4798 1988  323]
[ 323 1988 4798 ... 2953 3962 4357]
```

```
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.frame.DataFrame'>
```

Out[67]:

	Feature_Word	Feature_index
499	belief	4999
77	95	4998
2233	hungry	4997
982	consist	4996
832	clipboards	4995
1219	determination	4994
2725	majority	4993
4611	touches	4992
2170	historical	4991
3614	puppet	4990
4801	virtually	4989
4486	teen	4988
3577	promethean	4987
3982	searching	4986
4600	tool	4985
376	assure	4984
3076	needed	4983
4851	we	4982
841	clothes	4981
1325	documenting	4980
292	ap	4979
4975	www	4978
276	animation	4977

	Feature_Word	Feature_index
50	400	4976
403	autistic	4975
1598	etc	4974
2165	hilarious	4973
4027	separate	4972
4864	weekends	4971
4192	soon	4970
2605	let	4969
4712	unfortunate	4968
1188	depend	4967
4971	writing	4966
919	competitive	4965
568	boring	4964
3160	obstacle	4963
327	area	4962
1761	fiction	4961
3633	qualified	4960

In [68]:
```
#TFIDF_Feature_40=TFIDF_Feature_IDX_dataFrame_sorted[:40]
#print(type(TFIDF_Feature_40))
#TFIDF_Feature_40
#
#TFIDF_Feature_EssTitle_40=TFIDF_Feature_40.drop(columns="Feature_index")
#print(type(TFIDF_Feature_EssTitle_40))
#print(TFIDF_Feature_EssTitle_40)
```

```
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.frame.DataFrame'>
       Feature_Word
499          belief
```

```
77               95
2233         hungry
982         consist
832      clipboards
1219  determination
2725       majority
4611        touches
2170     historical
3614         puppet
4801      virtually
4486           teen
3577     promethean
3982      searching
4600           tool
376          assure
3076         needed
4851             we
841         clothes
1325    documenting
292              ap
4975            www
276       animation
50              400
403        autistic
1598            etc
2165      hilarious
4027       separate
4864       weekends
4192           soon
2605            let
4712    unfortunate
1188         depend
4971        writing
919     competitive
568          boring
3160       obstacle
327            area
1761        fiction
3633      qualified
```

In [69]:

```
TFIDF_Feature_EssTitle=TFIDF_Feature_2K.drop(columns="Feature_index")
print(type(TFIDF_Feature_EssTitle))
print(TFIDF_Feature_EssTitle)
```

```
<class 'pandas.core.frame.DataFrame'>
       Feature_Word
499          belief
```

```
77              95
2233        hungry
982        consist
832     clipboards
1219  determination
2725      majority
4611       touches
2170    historical
3614        puppet
4801     virtually
4486          teen
3577    promethean
3982     searching
4600          tool
376         assure
3076        needed
4851            we
841        clothes
1325   documenting
292             ap
4975           www
276      animation
50             400
403       autistic
1598           etc
2165     hilarious
4027      separate
4864      weekends
4192          soon
...            ...
1399      dynamics
3513         press
2222           hub
533           bins
315    approaching
2272     immersion
986     consisting
601         breaks
415      awareness
2164        highly
1617       evident
793         chosen
400         author
1888          four
973    connections
2881         model
3128     notebooks
```

```
1735        farther
975         conquer
501         believe
1565       entirely
1445  educationally
448           based
3905          rough
3843           rest
1531        engages
1278    disciplines
1626         exceed
2093       hallways
2519           know

[2000 rows x 1 columns]
```

In [70]:
```
#TFIDF_Feature_EssTitle.Feature_Word.value_counts()
```

Out[70]:
```
mentors          1
fluorescent      1
clay             1
who              1
are              1
magazine         1
creators         1
account          1
contributions    1
syndrome         1
brainstorm       1
everyday         1
pen              1
emphasis         1
newcomers        1
other            1
wanting          1
reason           1
deficit          1
helped           1
individually     1
increasing       1
roll             1
supplied         1
responding       1
dire             1
calming          1
effective        1
age              1
```

```
enriching        1
                ..
dad              1
although         1
raz              1
violin           1
pushing          1
schools          1
clean            1
pupils           1
dynamics         1
missed           1
detroit          1
normally         1
very             1
wrong            1
strategic        1
several          1
crime            1
freely           1
distance         1
interaction      1
economic         1
eliminate        1
describe         1
letting          1
moved            1
mine             1
how              1
familiar         1
110              1
passions         1
Name: Feature_Word, Length: 2000, dtype: int64
```

2.2 Computing Co-occurance matrix

In [71]:

```
# please write all the code with proper documentation, and proper titles for each subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debugging your code
# make sure you featurize train and test data separatly

# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the reader
    # b. Legends if needed
```

```
    # c. X-axis label
    # d. Y-axis label
```

In [72]:
```
## https://stackoverflow.com/questions/55148750/converting-pandas-dataframe-to-numpy-array
#TFIDF_EssTitle_npAarr = TFIDF_Feature_EssTitle.Feature_Word.values
#print(type(TFIDF_EssTitle_npAarr))
#TFIDF_EssTitle_npAarr#
```

```
<class 'numpy.ndarray'>
```

Out[72]:
```
array(['belief', '95', 'hungry', ..., 'exceed', 'hallways', 'know'],
      dtype=object)
```

In [73]:
```
## https://stackoverflow.com/questions/55148750/converting-pandas-dataframe-to-numpy-array
#TFIDF_EssTitle_40_npAarr = TFIDF_Feature_EssTitle_40.Feature_Word.values
#print(type(TFIDF_EssTitle_40_npAarr))
#TFIDF_EssTitle_40_npAarr
```

```
<class 'numpy.ndarray'>
```

Out[73]:
```
array(['belief', '95', 'hungry', 'consist', 'clipboards', 'determination',
       'majority', 'touches', 'historical', 'puppet', 'virtually', 'teen',
       'promethean', 'searching', 'tool', 'assure', 'needed', 'we',
       'clothes', 'documenting', 'ap', 'www', 'animation', '400',
       'autistic', 'etc', 'hilarious', 'separate', 'weekends', 'soon',
       'let', 'unfortunate', 'depend', 'writing', 'competitive', 'boring',
       'obstacle', 'area', 'fiction', 'qualified'], dtype=object)
```

In [74]:
```
## https://stackoverflow.com/questions/41661801/python-calculate-the-co-occurrence-matrix
#length=2000
#CoMatrix = np.zeros([length,length]) # n is the count of all words
#print(CoMatrix[1,2])
#print(CoMatrix.shape)
#print(type(CoMatrix))
##def cal_occ(TFIDF_EssTitle_npAarr, listofSentence,CoMatrix):
#def cal_occ(TFIDF_EssTitle_npAarr, CoMatrix):
#    # https://www.geeksforgeeks.org/enumerate-in-python/
#    #print(listofSentence)
#    #for i,word in enumerate(listofSentence):
#    for i,word in enumerate(TFIDF_EssTitle_npAarr):
#        #print(i,word)
#        # if i = 3; max(3-5,0) and min(3+5,2000)    ----+----+----+
#        #           max(0) and min(8)               0 3    8
#        # if i = 100; max(100-5,0) and min(100+5,2000)    ----+--100----+----+
#        #           max(95) and min(105)                      95    |  105
#        # if i = 1998; max(1998-5,0) and min(1998+5,2000)    +----1998-+
```

```
#        #           max(1993) and min(105)
#        for j in range(max(i-window,0),min(i+window,length)):
#            print(word,i,TFIDF_EssTitle_npAarr[j],j)
#            if (i==j):
#                continue
#            elif (word==TFIDF_EssTitle_npAarr[j]):
#                print("Hello")
#                CoMatrix[word,TFIDF_EssTitle_npAarr[j]]+=1
#                CoMatrix[TFIDF_EssTitle_npAarr[j],word]=CoMatrix[word,TFIDF_EssTitle_npAarr[j]]
#
#
#window=5
#for sentence in tqdm(TFIDF_EssTitle_npAarr):
#    # https://developers.google.com/edu/python/lists
#    cal_occ(TFIDF_EssTitle_npAarr, CoMatrix)
```

In [75]:

```
## https://stackoverflow.com/questions/41661801/python-calculate-the-co-occurrence-matrix
#length=40
#CoMatrix = np.zeros([length,length]) # n is the count of all words
#print(CoMatrix[1,2])
#print(CoMatrix.shape)
#print(type(CoMatrix))
##def cal_occ(TFIDF_EssTitle_npAarr, listofSentence,CoMatrix):
#def cal_occ(TFIDF_EssTitle_40_npAarr, CoMatrix):
#    # https://www.geeksforgeeks.org/enumerate-in-python/
#    #print(listofSentence)
#    #for i,word in enumerate(listofSentence):
#    for i,word in tqdm(enumerate(TFIDF_EssTitle_40_npAarr)):
#        #print(i,word)
#        # if i = 3; max(3-5,0) and min(3+5,2000)    ----+----+----+
#        #           max(0) and min(8)               0 3    8
#        # if i = 100; max(100-5,0) and min(100+5,2000)    ----+--100----+----+
#        #           max(95) and min(105)                      95    |  105
#        # if i = 1998; max(1998-5,0) and min(1998+5,2000)    +----1998-+
#        #           max(1993) and min(105)
#        for j in range(max(i-window,0),min(i+window,length)):
#            if (i==j):
#                continue
#            elif (word==TFIDF_EssTitle_40_npAarr[j]):
#                CoMatrix[word,TFIDF_EssTitle_40_npAarr[j]]+=1
#                CoMatrix[TFIDF_EssTitle_40_npAarr[j],word]=CoMatrix[word,TFIDF_EssTitle_40_npAarr[j]]
#
#
#window=2
```

```
#for sentence in tqdm(TFIDF_EssTitle_40_npAarr):
#    # https://developers.google.com/edu/python/lists
#    cal_occ(TFIDF_EssTitle_npAarr, CoMatrix)
```

In [76]:

```
#print(CoMatrix[1,2])
#print(CoMatrix.shape)
#print(type(CoMatrix))
#print(CoMatrix)
```

In [77]:

```
# https://stackoverflow.com/questions/41661801/python-calculate-the-co-occurrence-matrix
length=6
CoMatrix = np.zeros([length,length]) # n is the count of all words

print(CoMatrix[1,2])
print(CoMatrix.shape)
print(type(CoMatrix))
#def cal_occ(TFIDF_EssTitle_npAarr, listofSentence,CoMatrix):
def cal_occ(CoMatDF,CorpusList):
    # https://www.geeksforgeeks.org/enumerate-in-python/
    for i,word in enumerate(CorpusList):
        #print(i,word)
        # if i = 3; max(3-5,0) and min(3+5,2000)    ----+----+----+
        #           max(0) and min(8)               0 3    8
        # if i = 100; max(100-5,0) and min(100+5,2000)    ----+--100----+----+
        #           max(95) and min(105)                     95    |  105
        # if i = 1998; max(1998-5,0) and min(1998+5,2000)    +----1998-+
        #           max(1993) and min(105)
        for j in range(max(i-window,0),min(i+window,rangeLength)+1): # adding 1, coz loop won't execute till last interation.
            #print(word,i,CorpusList[j],j)
            #print("Range:",max(i-window,0),min(i+window,rangeLength))
            if (i==j):
                continue #print("---diagonal---")
            else: #if (word==Corpus[j]):
                #print("-------------incrementby1")
                CoMatDF.loc[word,CorpusList[j]]+=1
                #print(CoMatDF)
                #CoMatrix[Corpus[j],word]=CoMatrix[word,Corpus[j]]


window=2
Corpus = "He is not lazy He is intelligent He is smart"
CorpusList=[]
CorpusList=list(Corpus.split(" "))
```

```
print(CorpusList)
# ['He', 'is', 'not', 'lazy', 'He', 'is', 'intelligent', 'He', 'is', 'smart']
# --0-----1------2------3------4------5---------6---------7------8------9---

#rangeLength=length+1 #because range func do not include the last interation.
rangeLength=len(CorpusList)-1
print("rangeLength:",rangeLength)
MatColumns=['He', 'is', 'not', 'lazy', 'intelligent', 'smart']

CoMatDF=pd.DataFrame(data=CoMatrix,index=MatColumns,columns=MatColumns)
print(CoMatDF)
#for sentence in tqdm(CorpusList):
    #print("----------------------STARTING--------------------")
    # https://developers.google.com/edu/python/lists
cal_occ(CoMatDF,CorpusList)
print(CoMatDF)
```

```
0.0
(6, 6)
<class 'numpy.ndarray'>
['He', 'is', 'not', 'lazy', 'He', 'is', 'intelligent', 'He', 'is', 'smart']
rangeLength: 9
              He   is  not  lazy  intelligent  smart
He           0.0  0.0  0.0   0.0          0.0    0.0
is           0.0  0.0  0.0   0.0          0.0    0.0
not          0.0  0.0  0.0   0.0          0.0    0.0
lazy         0.0  0.0  0.0   0.0          0.0    0.0
intelligent  0.0  0.0  0.0   0.0          0.0    0.0
smart        0.0  0.0  0.0   0.0          0.0    0.0
              He   is  not  lazy  intelligent  smart
He           0.0  4.0  2.0   1.0          2.0    1.0
is           4.0  0.0  1.0   2.0          2.0    1.0
not          2.0  1.0  0.0   1.0          0.0    0.0
lazy         1.0  2.0  1.0   0.0          0.0    0.0
intelligent  2.0  2.0  0.0   0.0          0.0    0.0
smart        1.0  1.0  0.0   0.0          0.0    0.0
```

In [78]:

```
X_train["EssayTitle"].head(5)
```

Out[78]:

```
9115     liberty elementary title 1 school large percen...
13389    my students come diverse backgrounds they come...
13827    environment shapes experience no less true cla...
7263     my students diverse group ambitious enthusiast...
45303    my students diverse ethnicity also abilities t...
Name: EssayTitle, dtype: object
```

In [79]:
```
#TFIDF_Feature_EssTitle
#TFIDF_Feature_EssTitle_40
```

Out[79]:

	Feature_Word
499	belief
77	95
2233	hungry
982	consist
832	clipboards
1219	determination
2725	majority
4611	touches
2170	historical
3614	puppet
4801	virtually
4486	teen
3577	promethean
3982	searching
4600	tool
376	assure
3076	needed
4851	we
841	clothes
1325	documenting
292	ap
4975	www

	Feature_Word
276	animation
50	400
403	autistic
1598	etc
2165	hilarious
4027	separate
4864	weekends
4192	soon
2605	let
4712	unfortunate
1188	depend
4971	writing
919	competitive
568	boring
3160	obstacle
327	area
1761	fiction
3633	qualified

In [80]:
```
TFIDF_Feature_EssTitle
```

Out[80]:

	Feature_Word
499	belief
77	95
2233	hungry

	Feature_Word
982	consist
832	clipboards
1219	determination
2725	majority
4611	touches
2170	historical
3614	puppet
4801	virtually
4486	teen
3577	promethean
3982	searching
4600	tool
376	assure
3076	needed
4851	we
841	clothes
1325	documenting
292	ap
4975	www
276	animation
50	400
403	autistic
1598	etc
2165	hilarious

	Feature_Word
4027	separate
4864	weekends
4192	soon
...	...
1399	dynamics
3513	press
2222	hub
533	bins
315	approaching
2272	immersion
986	consisting
601	breaks
415	awareness
2164	highly
1617	evident
793	chosen
400	author
1888	four
973	connections
2881	model
3128	notebooks
1735	farther
975	conquer
501	believe

	Feature_Word
1565	entirely
1445	educationally
448	based
3905	rough
3843	rest
1531	engages
1278	disciplines
1626	exceed
2093	hallways
2519	know

2000 rows × 1 columns

In [81]:
```
#X_EssayTitle_200=X_train["EssayTitle"][:200]
#X_EssayTitle_200
#
#X_EssayTitle_2=X_train["EssayTitle"][:2]
#X_EssayTitle_2
```

Out[81]:
```
9115     liberty elementary title 1 school large percen...
13389    my students come diverse backgrounds they come...
Name: EssayTitle, dtype: object
```

In [82]:
```
#type(TFIDF_Feature_EssTitle_40)
```

Out[82]:
```
pandas.core.frame.DataFrame
```

In [83]:
```
def chk_with_Key_feature_list(text):
    wlist=[]
    #print(text)
    #print(type(text))
    wlist=list(text.split(sep=None))
    # https://stackoverflow.com/questions/14769162/find-matching-words-in-a-list-and-a-string
    if set(wlist).intersection(Key_feature_list):
```

```
        return True
    return False
```

2.3 Applying TruncatedSVD and Calculating Vectors for $essay$ and $project_t it \leq$

In [86]:
```
# please write all the code with proper documentation, and proper titles for each subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debugging your code
# make sure you featurize train and test data separatly

# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the reader
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

- 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)

In [87]:
```
type(CoMatDF)
CoMatDF.shape
```

Out[87]:
```
(2000, 2000)
```

In [88]:
```
(CoMatDF != 0).sum(1).sum()
```

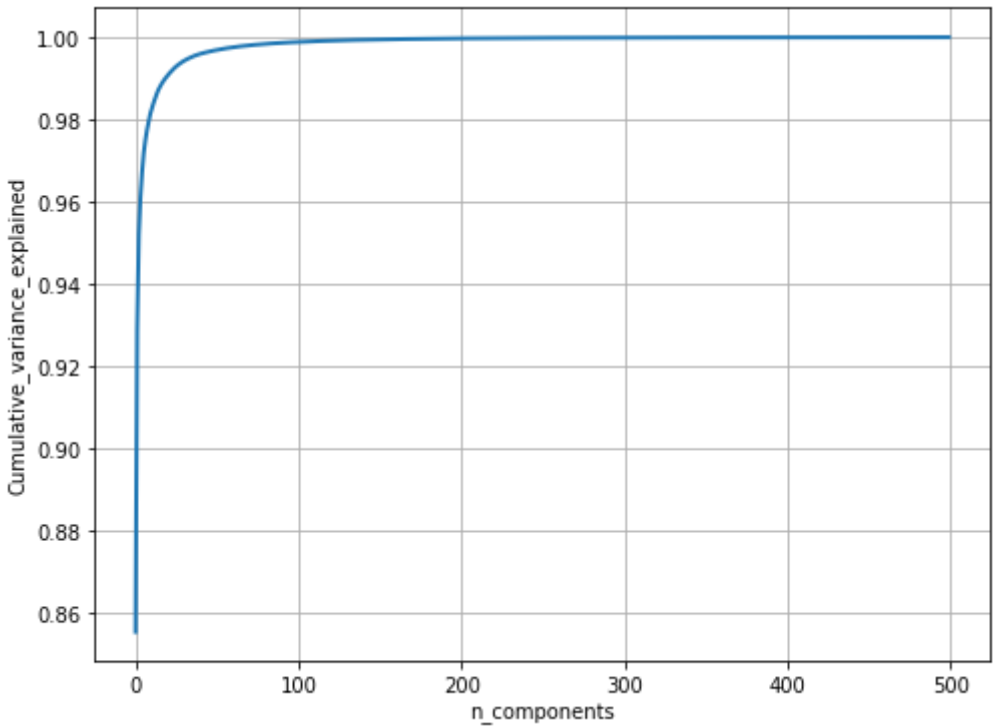
Out[88]:
```
799416
```

In [89]:
```
np.count_nonzero(CoMatDF)
```

Out[89]:
```
799416
```

In [90]:
```
from sklearn.decomposition import TruncatedSVD
```

```
svd = TruncatedSVD(n_components = 500)
model = svd.fit_transform(CoMatDF)

percentage_var_explained = svd.explained_variance_ / np.sum(svd.explained_variance_);
cum_var_explained = np.cumsum(percentage_var_explained)

# cumulative explained variance vs n_components
plt.figure(figsize=(8, 6))
plt.plot(cum_var_explained, linewidth=2)
plt.axis()
plt.grid(True)
plt.xlabel('n_components')
plt.ylabel('Cumulative_variance_explained')
plt.show()
```



N=100 covers most of the data variance

In [101…

```
svd = TruncatedSVD(n_components=100)
result_train=svd.fit_transform(CoMatDF)
```

```
print(result_train.shape)
result_train
```

```
(2000, 100)
```

Out[101…

```
array([[ 8.65975518e+01, -2.49066971e+01, -3.10935040e+00, ...,
         9.86882871e-01,  1.54994059e+00, -1.71166994e+00],
       [ 2.76688889e+02, -6.70872185e+01,  1.26368687e+02, ...,
        -2.09911203e+00,  9.34264311e+00, -1.65988069e+00],
       [ 6.25623395e+02, -1.48470007e+02,  9.01505226e+01, ...,
        -1.01651050e+01,  1.14323711e+01, -3.07212495e+00],
       ...,
       [ 1.05075709e+02, -4.08002191e+01, -1.06634487e+00, ...,
        -8.18547436e+00,  2.15838542e+00, -2.02250520e+00],
       [ 5.27617420e+01, -5.20953656e+00,  1.52674437e+01, ...,
         7.05827172e-01, -5.99726051e-01, -9.53859538e-01],
       [ 4.11328688e+03, -1.04290408e+03, -4.74685879e+02, ...,
        -3.29641916e+01, -1.01006062e+01, -2.70600036e+01]])
```

> - 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)

In [113…

```
model = result_train
glove_words = set(df_idf_sort_desc_2k.index)
keys={}
for i,j in enumerate(glove_words):
    #print(i,j)
    keys[j]=i
keys
```

Out[113…

```
{'keys': 0,
 'severely': 1,
 'predominately': 2,
 'scenes': 3,
 'palsy': 4,
 'association': 5,
 'enabling': 6,
 'versions': 7,
 'fly': 8,
 'librarian': 9,
 'schedules': 10,
 'hearted': 11,
 'believer': 12,
 'containers': 13,
 'makeup': 14,
```

```
                 'nannanstudent': 15,
                 'happened': 16,
                 'climb': 17,
                 'chess': 18,
                 'traumatic': 19,
                 'signs': 20,
                 'eyed': 21,
                 'hoops': 22,
                 'invest': 23,
                 'fewer': 24,
                 'bill': 25,
                 'confined': 26,
                 'luxury': 27,
                 'amazes': 28,
                 'packed': 29,
                 'nannancreative': 30,
                 'concerned': 31,
                 'immigrated': 32,
                 'thin': 33,
                 'unprepared': 34,
                 'reuse': 35,
                 'joyful': 36,
                 'basically': 37,
                 '94': 38,
                 'saved': 39,
                 'recorded': 40,
                 'christmas': 41,
                 'facilities': 42,
                 'breathe': 43,
                 'diet': 44,
                 'housed': 45,
                 'approaches': 46,
                 'graduates': 47,
                 'understandings': 48,
                 'rare': 49,
                 'myriad': 50,
                 'wet': 51,
                 'closing': 52,
                 'yearning': 53,
                 'windows': 54,
                 'brainpop': 55,
                 'reside': 56,
                 'refrigerator': 57,
                 'miami': 58,
                 'bonus': 59,
                 'demonstrated': 60,
                 'ti': 61,
```

```
                 'worse': 62,
                 'bases': 63,
                 'completion': 64,
                 'pulled': 65,
                 'placement': 66,
                 'soap': 67,
                 'transferred': 68,
                 'softball': 69,
                 'pet': 70,
                 'ocean': 71,
                 'lens': 72,
                 'architecture': 73,
                 'hardship': 74,
                 'movers': 75,
                 '2015': 76,
                 'facility': 77,
                 'remote': 78,
                 'detail': 79,
                 'stamp': 80,
                 'virginia': 81,
                 '1000': 82,
                 'cardstock': 83,
                 'coaching': 84,
                 '96': 85,
                 'transport': 86,
                 'organizations': 87,
                 'las': 88,
                 'sequence': 89,
                 'montessori': 90,
                 'clues': 91,
                 'popcorn': 92,
                 'followed': 93,
                 'oakland': 94,
                 'harvest': 95,
                 'stream': 96,
                 'gathering': 97,
                 'wasted': 98,
                 'ambassadors': 99,
                 'them': 100,
                 'suburb': 101,
                 'embraces': 102,
                 'balancing': 103,
                 'entry': 104,
                 'detailed': 105,
                 'lock': 106,
                 'nannanstudents': 107,
                 'upgrade': 108,
```

```
                 'addressed': 109,
                 'colleagues': 110,
                 'french': 111,
                 'lies': 112,
                 'grows': 113,
                 'happiness': 114,
                 'surely': 115,
                 'clutter': 116,
                 'passing': 117,
                 'invite': 118,
                 'before': 119,
                 'buzz': 120,
                 'rhythms': 121,
                 'upset': 122,
                 'supposed': 123,
                 'stacked': 124,
                 '29': 125,
                 'links': 126,
                 'proactive': 127,
                 'nannanschool': 128,
                 'accompany': 129,
                 'resourceful': 130,
                 'scholarships': 131,
                 'earbuds': 132,
                 'baskets': 133,
                 'empathetic': 134,
                 'dvd': 135,
                 'visits': 136,
                 'displaying': 137,
                 'crafts': 138,
                 'helpers': 139,
                 'hug': 140,
                 'touches': 141,
                 'exchange': 142,
                 'hydrated': 143,
                 'subtracting': 144,
                 'spirited': 145,
                 'comments': 146,
                 'fix': 147,
                 'nannanart': 148,
                 'traditions': 149,
                 'costs': 150,
                 'capability': 151,
                 'storytelling': 152,
                 'jumping': 153,
                 'adapted': 154,
                 'worthy': 155,
```

```
 'theory': 156,
 'consisting': 157,
 'liked': 158,
 'truth': 159,
 'bikes': 160,
 'laughing': 161,
 'aspiring': 162,
 'ozobots': 163,
 'ecosystem': 164,
 'mother': 165,
 'mandated': 166,
 'dances': 167,
 'couch': 168,
 'fairly': 169,
 'strides': 170,
 'regarding': 171,
 'employment': 172,
 'socks': 173,
 'nannanfun': 174,
 'tells': 175,
 'invent': 176,
 'broad': 177,
 'unknown': 178,
 'shakespeare': 179,
 'trees': 180,
 'blending': 181,
 'labels': 182,
 'hanging': 183,
 'fashion': 184,
 'nannanwhat': 185,
 'drum': 186,
 'workplace': 187,
 'wise': 188,
 'september': 189,
 'caused': 190,
 'extras': 191,
 'clip': 192,
 'river': 193,
 'manners': 194,
 'invited': 195,
 'figuring': 196,
 'eighty': 197,
 'thousand': 198,
 'button': 199,
 'inventions': 200,
 'appeal': 201,
 'advances': 202,
```

```
 'hate': 203,
 'requirement': 204,
 'window': 205,
 'gang': 206,
 'repeat': 207,
 'einstein': 208,
 'ongoing': 209,
 'forces': 210,
 'suited': 211,
 'powerpoint': 212,
 'enriched': 213,
 'tirelessly': 214,
 'reusable': 215,
 'diego': 216,
 'article': 217,
 'witness': 218,
 'individualize': 219,
 'cones': 220,
 'technique': 221,
 'figures': 222,
 'easels': 223,
 'civil': 224,
 'crowded': 225,
 'colleges': 226,
 'passed': 227,
 'genuinely': 228,
 'soul': 229,
 'roughly': 230,
 'plates': 231,
 'repeated': 232,
 'sized': 233,
 'costly': 234,
 'somewhere': 235,
 'scan': 236,
 'emerging': 237,
 'length': 238,
 'dell': 239,
 'weaknesses': 240,
 'extensive': 241,
 'joining': 242,
 'update': 243,
 'dark': 244,
 'user': 245,
 'minorities': 246,
 'ohio': 247,
 'behaved': 248,
 'cafeteria': 249,
```

```
 'indian': 250,
 'hyperactivity': 251,
 'damaged': 252,
 'dvds': 253,
 'beg': 254,
 'worthwhile': 255,
 'behaviorally': 256,
 'worksheet': 257,
 'repair': 258,
 'nannanmrs': 259,
 '48': 260,
 'mississippi': 261,
 'recognizing': 262,
 'credit': 263,
 'tons': 264,
 'beautifully': 265,
 'teen': 266,
 'witnessed': 267,
 'television': 268,
 'worries': 269,
 'motivator': 270,
 'concert': 271,
 'resistance': 272,
 'blow': 273,
 'log': 274,
 'bookshelf': 275,
 'drone': 276,
 'grip': 277,
 'tracks': 278,
 'divided': 279,
 'riding': 280,
 'worried': 281,
 'monitoring': 282,
 'foundations': 283,
 'flowing': 284,
 'essay': 285,
 'occasion': 286,
 'handful': 287,
 'statistics': 288,
 'plate': 289,
 'listed': 290,
 'interpret': 291,
 'talks': 292,
 'steady': 293,
 'wealthy': 294,
 'cords': 295,
 'homeroom': 296,
```

```
 'nannanbouncing': 297,
 'cds': 298,
 'ears': 299,
 'nex': 300,
 'president': 301,
 'producing': 302,
 'disruption': 303,
 'laminated': 304,
 'appalachian': 305,
 'nannanlistening': 306,
 'themed': 307,
 'holes': 308,
 'firm': 309,
 'inquire': 310,
 'thriving': 311,
 'surrounds': 312,
 'investing': 313,
 'brave': 314,
 'june': 315,
 'bed': 316,
 'paperless': 317,
 'theatre': 318,
 'nannanliteracy': 319,
 'certified': 320,
 'dominican': 321,
 'impress': 322,
 'dense': 323,
 'nannanorganization': 324,
 'apparent': 325,
 'builders': 326,
 'resilience': 327,
 'masters': 328,
 'contributions': 329,
 'firsthand': 330,
 'sticky': 331,
 'rooms': 332,
 'keyboarding': 333,
 'scheduled': 334,
 'holidays': 335,
 'greeting': 336,
 'supplied': 337,
 'smooth': 338,
 'supplemental': 339,
 'concentrating': 340,
 'goggles': 341,
 'laminator': 342,
 'bellies': 343,
```

```
 'hockey': 344,
 'rope': 345,
 'tricky': 346,
 'nearby': 347,
 'nannanoh': 348,
 'elmo': 349,
 'tears': 350,
 'historically': 351,
 'agricultural': 352,
 'torn': 353,
 'counselor': 354,
 'bridges': 355,
 'encompasses': 356,
 'catching': 357,
 'gloves': 358,
 'setup': 359,
 'lectures': 360,
 'sanitizer': 361,
 'virtually': 362,
 'ribbon': 363,
 'qualities': 364,
 'breakout': 365,
 'agriculture': 366,
 'overwhelmed': 367,
 'laminating': 368,
 'george': 369,
 'statement': 370,
 'ranges': 371,
 'recycled': 372,
 'museum': 373,
 'sick': 374,
 'digitally': 375,
 'nannanheadphones': 376,
 'swing': 377,
 'mandela': 378,
 'borrowed': 379,
 'streets': 380,
 'uncommon': 381,
 'tutorials': 382,
 'monthly': 383,
 'bones': 384,
 'imagined': 385,
 'constructive': 386,
 'revise': 387,
 'excelling': 388,
 'cricut': 389,
 'designers': 390,
```

```
            'relaxing': 391,
            'clock': 392,
            'worker': 393,
            'yesterday': 394,
            'bricks': 395,
            'versus': 396,
            'established': 397,
            'yard': 398,
            'benches': 399,
            'initial': 400,
            'prevents': 401,
            'sustain': 402,
            'exhibit': 403,
            'which': 404,
            'reflective': 405,
            'delivery': 406,
            'footballs': 407,
            'heritage': 408,
            '92': 409,
            'nannanall': 410,
            'integrity': 411,
            'rising': 412,
            'mat': 413,
            'category': 414,
            'belongings': 415,
            'flip': 416,
            'dallas': 417,
            'correlate': 418,
            'instilling': 419,
            'smartboard': 420,
            'participates': 421,
            'pathway': 422,
            'temperature': 423,
            'drastically': 424,
            'redesign': 425,
            'butterflies': 426,
            'holiday': 427,
            'bugs': 428,
            'graduating': 429,
            'calendar': 430,
            'suggestions': 431,
            'puppet': 432,
            'answering': 433,
            'films': 434,
            'chaos': 435,
            'locate': 436,
            'falls': 437,
```

```
 'relieve': 438,
 'limiting': 439,
 'pattern': 440,
 'preferred': 441,
 'reservation': 442,
 '110': 443,
 'largely': 444,
 'persistence': 445,
 'advantages': 446,
 'equity': 447,
 'treasure': 448,
 'debate': 449,
 'symbols': 450,
 'el': 451,
 'launch': 452,
 'pushed': 453,
 'expanded': 454,
 'leaning': 455,
 'climbing': 456,
 'village': 457,
 'oil': 458,
 'inspirations': 459,
 'quiz': 460,
 'entirely': 461,
 'tubes': 462,
 'competent': 463,
 'dolls': 464,
 'shortage': 465,
 'max': 466,
 'battery': 467,
 'vastly': 468,
 'inexpensive': 469,
 'covering': 470,
 'except': 471,
 'bouncing': 472,
 'terrific': 473,
 'prints': 474,
 'nannancan': 475,
 'celebration': 476,
 'utensils': 477,
 'thoroughly': 478,
 'bass': 479,
 'nonverbal': 480,
 'golf': 481,
 'outgoing': 482,
 'identity': 483,
 'studied': 484,
```

```
 'plain': 485,
 'soil': 486,
 'anger': 487,
 '55': 488,
 'equitable': 489,
 'scary': 490,
 'ieps': 491,
 'dollars': 492,
 'diagrams': 493,
 'citizen': 494,
 'seed': 495,
 'frames': 496,
 'youtube': 497,
 'constraints': 498,
 'bubbles': 499,
 'wheel': 500,
 'farther': 501,
 'realistic': 502,
 'neighbors': 503,
 'ordered': 504,
 'leap': 505,
 'operating': 506,
 'crew': 507,
 'germs': 508,
 'beliefs': 509,
 'spatial': 510,
 'de': 511,
 '34': 512,
 'pbl': 513,
 'representing': 514,
 'hidden': 515,
 'counters': 516,
 'taste': 517,
 'kidney': 518,
 'jr': 519,
 'replaced': 520,
 'safer': 521,
 'tale': 522,
 'stocked': 523,
 'linguistic': 524,
 'comic': 525,
 'salt': 526,
 'alternatives': 527,
 'questioning': 528,
 'host': 529,
 'decorate': 530,
 'sheet': 531,
```

```
 'xylophones': 532,
 'inventors': 533,
 'conflict': 534,
 'nannankeep': 535,
 'conclusions': 536,
 'nationalities': 537,
 'drawers': 538,
 'illustrate': 539,
 'tile': 540,
 'showed': 541,
 'mr': 542,
 'lexia': 543,
 'economics': 544,
 'dramatically': 545,
 'looping': 546,
 'bin': 547,
 'sun': 548,
 'bottle': 549,
 'differentiating': 550,
 'achievers': 551,
 'rain': 552,
 'trials': 553,
 'fallen': 554,
 'tag': 555,
 'condition': 556,
 'nervous': 557,
 'peace': 558,
 'carrying': 559,
 'investigation': 560,
 'optimistic': 561,
 'off': 562,
 'replacing': 563,
 'explorations': 564,
 'knees': 565,
 'messages': 566,
 'recipe': 567,
 'strategic': 568,
 'additions': 569,
 'hop': 570,
 'twelve': 571,
 'stomach': 572,
 'charged': 573,
 'million': 574,
 'foldables': 575,
 'gears': 576,
 'newcomers': 577,
 'tie': 578,
```

```
 'floors': 579,
 'stepping': 580,
 'illustrators': 581,
 'brighten': 582,
 'gardens': 583,
 'flooding': 584,
 'guiding': 585,
 'ok': 586,
 'checking': 587,
 'mold': 588,
 'ipods': 589,
 'retelling': 590,
 'sat': 591,
 'citizenship': 592,
 'packs': 593,
 'integrates': 594,
 'vivid': 595,
 'conquer': 596,
 'nannanlittle': 597,
 'amazingly': 598,
 'chips': 599,
 'calculator': 600,
 'procedures': 601,
 'potentially': 602,
 'remediation': 603,
 'newest': 604,
 'ipod': 605,
 'fostered': 606,
 'man': 607,
 'pantry': 608,
 'qualified': 609,
 'driving': 610,
 'accustomed': 611,
 'planting': 612,
 'ict': 613,
 'gangs': 614,
 'suitable': 615,
 'microsoft': 616,
 'formation': 617,
 'tips': 618,
 'mid': 619,
 'hero': 620,
 'dividers': 621,
 'requiring': 622,
 'harness': 623,
 'wood': 624,
 'hp': 625,
```

```
 'collected': 626,
 'trade': 627,
 'velcro': 628,
 'press': 629,
 'globally': 630,
 'frog': 631,
 'drawn': 632,
 'division': 633,
 'northwest': 634,
 'violins': 635,
 'restless': 636,
 'cues': 637,
 'surprised': 638,
 'solidify': 639,
 'journalism': 640,
 'touching': 641,
 'binder': 642,
 'fridays': 643,
 'microphones': 644,
 'bears': 645,
 'diagnosis': 646,
 'zest': 647,
 'invention': 648,
 'freshmen': 649,
 'prizes': 650,
 'consuming': 651,
 'belonging': 652,
 'ninth': 653,
 'zones': 654,
 'ups': 655,
 'files': 656,
 'ladies': 657,
 'mixing': 658,
 'francisco': 659,
 'alaska': 660,
 'breathing': 661,
 'dictionary': 662,
 'partitions': 663,
 'snap': 664,
 'nannansuper': 665,
 'harry': 666,
 'nannantime': 667,
 'toy': 668,
 'extension': 669,
 'mo': 670,
 'locally': 671,
 'suits': 672,
```

```
 'male': 673,
 'suit': 674,
 'osmos': 675,
 'fed': 676,
 'circuit': 677,
 'mountain': 678,
 'hole': 679,
 'remove': 680,
 'lined': 681,
 'scared': 682,
 'dad': 683,
 'nights': 684,
 'painted': 685,
 'violin': 686,
 'tissue': 687,
 'oregon': 688,
 'wake': 689,
 'missed': 690,
 'burden': 691,
 'treats': 692,
 'brown': 693,
 'baltimore': 694,
 'scene': 695,
 'distance': 696,
 'fictional': 697,
 'gotten': 698,
 'ton': 699,
 'picking': 700,
 'delayed': 701,
 'debates': 702,
 'perfectly': 703,
 'ukuleles': 704,
 'nestled': 705,
 'drums': 706,
 'beads': 707,
 '800': 708,
 'mile': 709,
 'medicine': 710,
 'canvas': 711,
 'disturbing': 712,
 'hi': 713,
 'awarded': 714,
 'nelson': 715,
 'nannanextra': 716,
 'consistency': 717,
 'props': 718,
 'butterfly': 719,
```

```
 'link': 720,
 'operate': 721,
 'owl': 722,
 'david': 723,
 'adaptive': 724,
 'forgotten': 725,
 'mindfulness': 726,
 'uniforms': 727,
 'blog': 728,
 'tolerance': 729,
 'la': 730,
 'coffee': 731,
 'describes': 732,
 'peaceful': 733,
 'favorites': 734,
 'published': 735,
 '150': 736,
 'amazon': 737,
 'screens': 738,
 'sits': 739,
 'arise': 740,
 'instruct': 741,
 'reflected': 742,
 'situated': 743,
 'forming': 744,
 'jersey': 745,
 'paints': 746,
 'explained': 747,
 'matching': 748,
 'generational': 749,
 'conducting': 750,
 'vocal': 751,
 'masterpiece': 752,
 'solar': 753,
 'fixed': 754,
 'lean': 755,
 'nannanmusic': 756,
 'honest': 757,
 'parachute': 758,
 'sea': 759,
 'weights': 760,
 'specials': 761,
 'session': 762,
 'excuses': 763,
 '31': 764,
 'nurtured': 765,
 'fitbit': 766,
```

```
 'gel': 767,
 'conjunction': 768,
 'metal': 769,
 'tip': 770,
 'smallest': 771,
 'combat': 772,
 'scholar': 773,
 'ordinary': 774,
 'gardening': 775,
 'versatile': 776,
 'net': 777,
 'pain': 778,
 'ib': 779,
 'rugs': 780,
 'recommended': 781,
 'counseling': 782,
 'baby': 783,
 'smarter': 784,
 'endurance': 785,
 'reminders': 786,
 'prices': 787,
 'voracious': 788,
 'beloved': 789,
 'bike': 790,
 'incomes': 791,
 'stopped': 792,
 'avenue': 793,
 'piano': 794,
 'ixl': 795,
 'nutritional': 796,
 'justice': 797,
 'impaired': 798,
 'athletics': 799,
 'nannanusing': 800,
 'availability': 801,
 'named': 802,
 'blend': 803,
 'pathways': 804,
 'crisis': 805,
 'combining': 806,
 '97': 807,
 'nannantablets': 808,
 'mornings': 809,
 'responsive': 810,
 'vegetable': 811,
 'holocaust': 812,
 'protection': 813,
```

```
 'giant': 814,
 'solely': 815,
 'suffer': 816,
 'mark': 817,
 'everybody': 818,
 'transitions': 819,
 'greek': 820,
 'inventory': 821,
 'masterpieces': 822,
 'translate': 823,
 'hospital': 824,
 'grants': 825,
 'rush': 826,
 'minimize': 827,
 'rhyming': 828,
 'wind': 829,
 'percussion': 830,
 'journeys': 831,
 '700': 832,
 'qr': 833,
 'nannanalternative': 834,
 'nannanteaching': 835,
 'necessarily': 836,
 'shirt': 837,
 'girl': 838,
 'demographic': 839,
 'kinetic': 840,
 'grouped': 841,
 'orchestra': 842,
 'intend': 843,
 'slide': 844,
 'microscopes': 845,
 'peak': 846,
 'flex': 847,
 'survival': 848,
 'mediums': 849,
 'chemical': 850,
 'samples': 851,
 'wheels': 852,
 'mighty': 853,
 'potter': 854,
 'narrative': 855,
 'roof': 856,
 'tracking': 857,
 'nannanipad': 858,
 'nannanfuture': 859,
 'hinder': 860,
```

```
 'friendship': 861,
 'compliment': 862,
 'toner': 863,
 'organizer': 864,
 'assisting': 865,
 'administrators': 866,
 'mentioned': 867,
 'calculus': 868,
 'whisper': 869,
 'lift': 870,
 'weighted': 871,
 'costumes': 872,
 'accounts': 873,
 'organic': 874,
 'eleven': 875,
 'earning': 876,
 'heads': 877,
 '3doodler': 878,
 'endure': 879,
 'chaotic': 880,
 'greeted': 881,
 'archery': 882,
 'battle': 883,
 'strips': 884,
 'tk': 885,
 'accidents': 886,
 'attached': 887,
 'collage': 888,
 '450': 889,
 'discouraged': 890,
 'she': 891,
 'beings': 892,
 'headphone': 893,
 'benjamin': 894,
 'contact': 895,
 'wider': 896,
 'reply': 897,
 'twist': 898,
 'locations': 899,
 'smell': 900,
 'investigating': 901,
 'relief': 902,
 'monitors': 903,
 'microscope': 904,
 'modes': 905,
 'flooded': 906,
 'hallways': 907,
```

```
 'gateway': 908,
 'strongly': 909,
 'kansas': 910,
 'disciplines': 911,
 'wire': 912,
 'intermediate': 913,
 'fundraisers': 914,
 'minnesota': 915,
 'predict': 916,
 'opposed': 917,
 'bond': 918,
 'nannanempowering': 919,
 'southwest': 920,
 'transformed': 921,
 'connects': 922,
 'nannantake': 923,
 'counts': 924,
 'newer': 925,
 'lovers': 926,
 'pulling': 927,
 'lakeshore': 928,
 'khan': 929,
 'drinks': 930,
 'bay': 931,
 'remarkable': 932,
 'compost': 933,
 'cheap': 934,
 'arizona': 935,
 'ecosystems': 936,
 'thousands': 937,
 'tear': 938,
 'cushion': 939,
 'makey': 940,
 'instance': 941,
 'universe': 942,
 'downs': 943,
 'rights': 944,
 'tutoring': 945,
 'paintings': 946,
 'selves': 947,
 'downloaded': 948,
 'expectation': 949,
 'keyboards': 950,
 'selections': 951,
 'evident': 952,
 'exams': 953,
 'prize': 954,
```

```
 'league': 955,
 'bluetooth': 956,
 'label': 957,
 'cubes': 958,
 'classics': 959,
 'participated': 960,
 'sewing': 961,
 'views': 962,
 'avoid': 963,
 'controlled': 964,
 'mess': 965,
 'nannanan': 966,
 'electricity': 967,
 'modifications': 968,
 'failures': 969,
 'dialogue': 970,
 'passages': 971,
 'desires': 972,
 'intense': 973,
 'spin': 974,
 '36': 975,
 'nannanfull': 976,
 'retell': 977,
 'applies': 978,
 'recreate': 979,
 'wires': 980,
 'performed': 981,
 'pursuing': 982,
 'hilarious': 983,
 'clipboard': 984,
 'assure': 985,
 'sharpeners': 986,
 'insects': 987,
 'volleyball': 988,
 'nannancoding': 989,
 'shopping': 990,
 'advancing': 991,
 'grew': 992,
 'brushes': 993,
 'cups': 994,
 'dull': 995,
 'profound': 996,
 'informative': 997,
 'approaching': 998,
 'chicken': 999,
 ...}
```

Make Data Model Ready: project_essay | AVG W2V

In [114…

```
# average Word2Vec for Train Essay
# compute average word2vec for each review.
X_train_essay_avg_w2v = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_train['essay'].values): # for each review/sentence
    vector = np.zeros(100) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words: #glove_word is a set
            vector += model[keys[word]]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    X_train_essay_avg_w2v.append(vector)

print(len(X_train_essay_avg_w2v))
print(len(X_train_essay_avg_w2v[0]))
```

```
100%|██████████████████████████████████████████████████████████████████████████| 33500/33500 [00:01<00:00, 26634.93it/s]
33500
100
```

In [116…

```
# average Word2Vec for Test Essay
# compute average word2vec for each review.
X_test_essay_avg_w2v = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_test['essay'].values): # for each review/sentence
    vector = np.zeros(100) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[keys[word]]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    X_test_essay_avg_w2v.append(vector)

print(len(X_test_essay_avg_w2v))
print(len(X_test_essay_avg_w2v[0]))
```

```
100%|██████████████████████████████████████████████████████████████████████████| 16500/16500 [00:00<00:00, 27144.56it/s]
16500
100
```

Make Data Model Ready: project_title | AVG W2V

In [118…

```
# average Word2Vec for Train Title
# compute average word2vec for each review.
X_train_title_avg_w2v = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_train['project_title'].values): # for each review/sentence
    vector = np.zeros(100) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[keys[word]]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    X_train_title_avg_w2v.append(vector)

print(len(X_train_title_avg_w2v))
print(len(X_train_title_avg_w2v[0]))
```

```
100%|██████████████████████████████████████████████████████████████████████████| 33500/33500 [00:00<00:00, 270856.90it/s]
33500
100
```

In [119…

```
# average Word2Vec for Test Essay
# compute average word2vec for each review.
X_test_title_avg_w2v = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_test['project_title'].values): # for each review/sentence
    vector = np.zeros(100) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[keys[word]]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    X_test_title_avg_w2v.append(vector)

print(len(X_test_title_avg_w2v))
print(len(X_test_title_avg_w2v[0]))
```

```
100%|██████████████████████████████████████████████████████████████████████████| 16500/16500 [00:00<00:00, 280415.63it/s]
16500
100
```

2.4 Merge the features from step 3 and step 4

In [97]:
```
# please write all the code with proper documentation, and proper titles for each subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debugging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the reader
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

In [120…
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr_set5_GBT = hstack((X_train_state_ohe, X_train_clean_ohe, X_train_cleanSub_ohe, X_train_grade_ohe,X_train_teacher_ohe, X_train
X_te_set5_GBT = hstack((X_test_state_ohe, X_test_clean_ohe,X_test_cleanSub_ohe, X_test_grade_ohe,X_test_teacher_ohe, X_test_quanti

print("Final Data matrix | XGBOOST")
print(X_tr_set5_GBT.shape, y_train.shape)
print(X_te_set5_GBT.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix | XGBOOST
(33500, 308) (33500,)
(16500, 308) (16500,)
====================================================================================================
```

2.5 Apply XGBoost on the Final Features from the above section

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

In [99]:
```
# No need to split the data into train and test(cv)
# use the Dmatrix and apply xgboost on the whole data
# please check the Quora case study notebook as reference

# please write all the code with proper documentation, and proper titles for each subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debugging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the reader
    # b. Legends if needed
```

```
    # c. X-axis label
    # d. Y-axis label
```

In [121…

```
import sys
import math

import numpy as np
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import roc_auc_score

# you might need to install this one
import xgboost as xgb

class XGBoostClassifier():
    def __init__(self, num_boost_round=10, **params):
        self.clf = None
        self.num_boost_round = num_boost_round
        self.params = params
        self.params.update({'objective': 'multi:softprob'})

    def fit(self, X, y, num_boost_round=None):
        num_boost_round = num_boost_round or self.num_boost_round
        self.label2num = {label: i for i, label in enumerate(sorted(set(y)))}
        dtrain = xgb.DMatrix(X, label=[self.label2num[label] for label in y])
        self.clf = xgb.train(params=self.params, dtrain=dtrain, num_boost_round=num_boost_round, verbose_eval=1)

    def predict(self, X):
        num2label = {i: label for label, i in self.label2num.items()}
        Y = self.predict_proba(X)
        y = np.argmax(Y, axis=1)
        return np.array([num2label[i] for i in y])

    def predict_proba(self, X):
        dtest = xgb.DMatrix(X)
        return self.clf.predict(dtest)

    def score(self, X, y):
        Y = self.predict_proba(X)[:,1]
        return roc_auc_score(y, Y)

    def get_params(self, deep=True):
        return self.params

    def set_params(self, **params):
```

```
        if 'num_boost_round' in params:
            self.num_boost_round = params.pop('num_boost_round')
        if 'objective' in params:
            del params['objective']
        self.params.update(params)
        return self
```

In [144…

```
XGclf = XGBoostClassifier(eval_metric = 'auc', num_class = 2, nthread = 4)
#####################################################################
#                Change from here                                   #
#####################################################################
parameters = {
    'num_boost_round': [5,11,15,21,25], #[100, 250, 500],
    'eta': [0.05, 0.1, 0.3],
    'max_depth': [2,3,5,7,10],  #[6, 9, 12],
    'subsample': [0.9, 1.0],
    'colsample_bytree': [0.9, 1.0],
}

clf = GridSearchCV(XGclf, parameters,cv=3, scoring='roc_auc', return_train_score=True)
# return_train_score : boolean, default=False
# If False, the cv_results_ attribute will not include training scores. Computing training scores is used to
# get insights on how different parameter settings impact the overfitting/underfitting trade-off. However computing
# the scores on the training set can be computationally expensive and is not strictly required to select the parameters
# that yield the best generalization performance.


#X = np.array([[1,2], [3,4], [2,1], [4,3], [1,0], [4,5]])
#Y = np.array([0, 1, 0, 1, 0, 1])
clf.fit(X_tr_set5_GBT, y_train)
```

Out[144…

```
GridSearchCV(cv=3, error_score='raise-deprecating',
             estimator=<__main__.XGBoostClassifier object at 0x000001FA524BEA58>,
             iid='warn', n_jobs=None,
             param_grid={'colsample_bytree': [0.9, 1.0],
                         'eta': [0.05, 0.1, 0.3], 'max_depth': [2, 3, 5, 7, 10],
                         'num_boost_round': [5, 11, 15, 21, 25],
                         'subsample': [0.9, 1.0]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
             scoring='roc_auc', verbose=0)
```

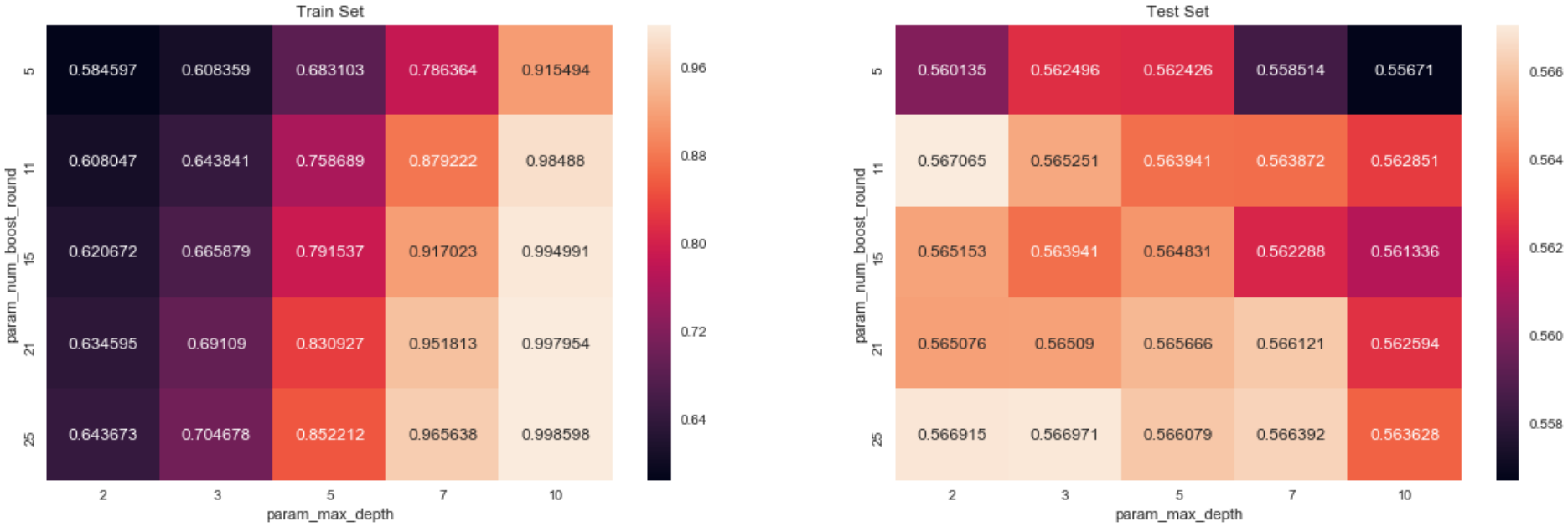
In [146…

```
print(clf.best_estimator_)
print(clf.best_params_)
print(clf.score(X_te_set5_GBT, y_test))
```

```
<__main__.XGBoostClassifier object at 0x000001FA52EFA9E8>
{'colsample_bytree': 1.0, 'eta': 0.3, 'max_depth': 2, 'num_boost_round': 11, 'subsample': 0.9}
0.5589668798423189
```

In [147…

```
# from Assignment 8_DonorsChoose_DT

# https://seaborn.pydata.org/generated/seaborn.heatmap.html
import seaborn as sns; sns.set()
max_scores1=pd.DataFrame(clf.cv_results_).groupby(['param_num_boost_round','param_max_depth']).max().unstack()[['mean_test_score',

fig,ax=plt.subplots(1,2,figsize=(20,6))

sns.heatmap(max_scores1.mean_train_score,annot=True,fmt='4g',ax=ax[0])
sns.heatmap(max_scores1.mean_test_score,annot=True,fmt='4g',ax=ax[1])

ax[0].set_title('Train Set')
ax[1].set_title('Test Set')

plt.show()
```



In [148…

```
print(clf.score(X_tr_set5_GBT,y_train))
print(clf.score(X_te_set5_GBT,y_test))
print(clf.best_params_)
print(clf.best_score_)
```

```
0.5930462801691487
0.5589668798423189
{'colsample_bytree': 1.0, 'eta': 0.3, 'max_depth': 2, 'num_boost_round': 11, 'subsample': 0.9}
0.567064911747751
```

Best Parameter

'max_depth': 2, 'num_boost_round': 11

In [149…

```
#Best tune parameters
param_grid = {'max_depth': [2],
              'num_boost_round': [11]
             }
```

In [150…

```
#code source: http://occam.olin.edu/sites/default/files/DataScienceMaterials/machine_learning_lecture_2/Machine%20Learning%20Lectu
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.datasets import *
import xgboost as xgb

#Using GridSearchCV
# https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
# XGBoostError: value 0 for Parameter num_class should be greater equal to 1
modelbestXBb = GridSearchCV(XGBoostClassifier(num_class = 2),param_grid)
modelbestXBb.fit(X_tr_set5_GBT, y_train)

print(modelbestXBb.best_score_)
print(modelbestXBb.score(X_te_set5_GBT, y_test))
```

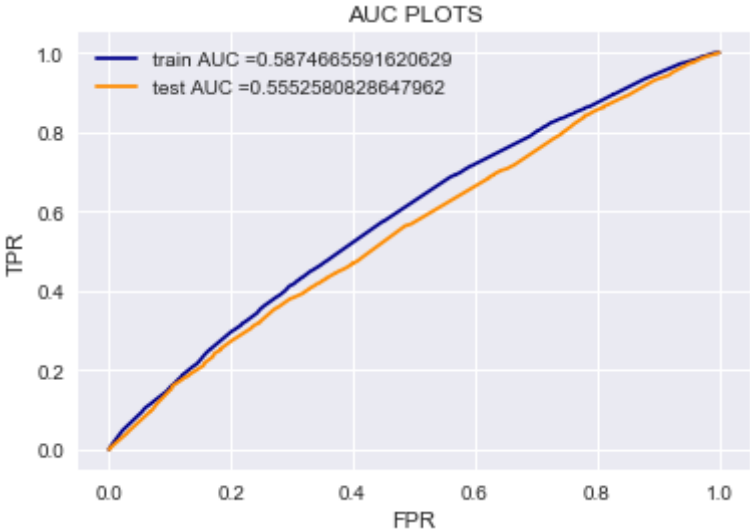
```
0.559504953417887
0.5552580828647962
```

In [152…

```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve
from sklearn.metrics import roc_curve, auc

y_train_XB_pred = modelbestXBb.predict_proba(X_tr_set5_GBT)[:,1]
y_test_XB_pred = modelbestXBb.predict_proba(X_te_set5_GBT)[:,1]

print(modelbestXBb.best_params_)
print(modelbestXBb.score(X_te_set5_GBT, y_test))

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_XB_pred)
```

```
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_XB_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)),color='darkblue')
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)),color='darkorange')
plt.legend()
plt.xlabel("FPR")
plt.ylabel("TPR")
plt.title("AUC PLOTS")
plt.grid(True)
plt.show()
```

```
{'max_depth': 2, 'num_boost_round': 11}
0.5552580828647962
```



In [153…

```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def predict(proba, threshould, fpr, tpr):

    t = threshould[np.argmax(tpr*(1-fpr))]

    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very high

    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshold", np.round(t,3))
    predictions = []
    for i in proba:
        if i>=t:
            predictions.append(1)
```

```
        else:
            predictions.append(0)
    return predictions
```

In [154…

```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
print("="*100)
from sklearn.metrics import confusion_matrix
print("Train confusion matrix")
print(confusion_matrix(y_train, predict(y_train_XB_pred, tr_thresholds, train_fpr, train_tpr)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict(y_test_XB_pred, te_thresholds, test_fpr, test_tpr)))
```

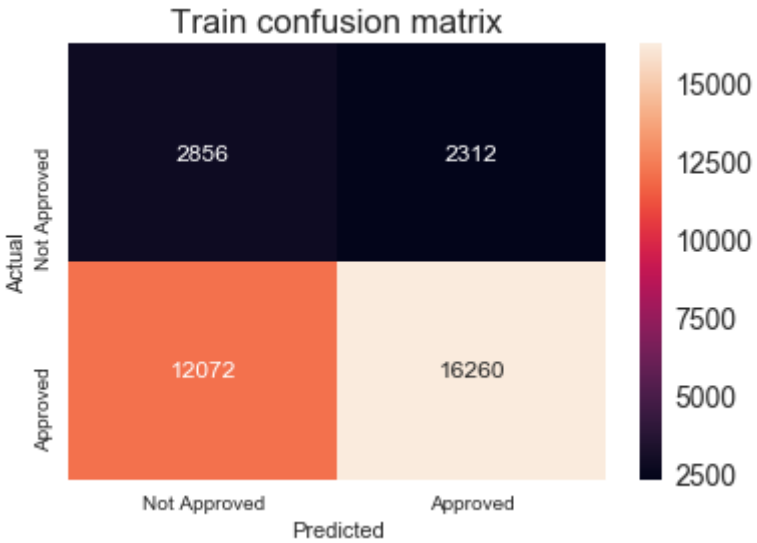
```
====================================================================================================
Train confusion matrix
the maximum value of tpr*(1-fpr) 0.31716043603290306 for threshold 0.836
[[ 2856  2312]
 [12072 16260]]
Test confusion matrix
the maximum value of tpr*(1-fpr) 0.2900099541519036 for threshold 0.837
[[1310 1236]
 [6089 7865]]
```

In [155…

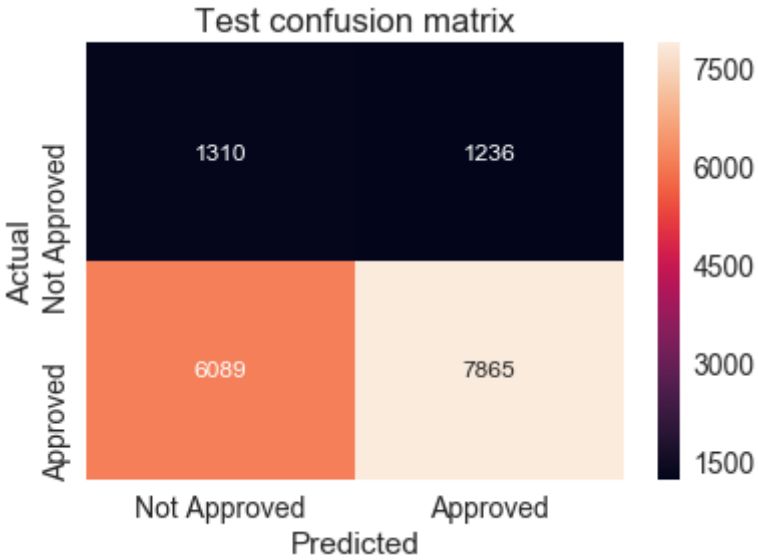
```
import seaborn as snTr
import seaborn as snTe
import pandas as pdH
import matplotlib.pyplot as pltTr
import matplotlib.pyplot as pltTe

# https://stackoverflow.com/questions/35572000/how-can-i-plot-a-confusion-matrix
arrayTr=confusion_matrix(y_train, predict(y_train_XB_pred, tr_thresholds, train_fpr, train_tpr))
df_cmTr = pdH.DataFrame(arrayTr,range(2),range(2))
#print(arrayTr)
# https://stackoverflow.com/questions/32723798/how-do-i-add-a-title-to-seaborn-heatmap
axTr = pltTr.axes()

snTr.set(font_scale=1.4)#for label size

# https://seaborn.pydata.org/generated/seaborn.heatmap.html

snTr.heatmap(df_cmTr, annot=True,annot_kws={"size": 12},fmt="d",ax=axTr)# font size, format in digit

labels=['Not Approved','Approved']
axTr.set_xticklabels(labels)
axTr.set_yticklabels(labels)
#Suggestion 4.Label confusion matrix heatmap with actual and predicted labels.
```

```
pltTr.title("Train confusion matrix")
pltTr.xlabel("Predicted")
pltTr.ylabel("Actual")
pltTr.show()

# https://stackoverflow.com/questions/50947776/plot-two-seaborn-heatmap-graphs-side-by-side
#fig, ax =plt.subplots(1,1)

# https://stackoverflow.com/questions/35572000/how-can-i-plot-a-confusion-matrix
arrayTe=confusion_matrix(y_test, predict(y_test_XB_pred, te_thresholds, test_fpr, test_tpr))
df_cmTe = pdH.DataFrame(arrayTe,range(2),range(2))

axTe = pltTe.axes()

snTe.set(font_scale=1.4)#for label size

# https://seaborn.pydata.org/generated/seaborn.heatmap.html
snTe.heatmap(df_cmTe, annot=True,annot_kws={"size": 12},fmt="d",ax=axTe)# font size, format in digit

#Suggestion 4.Label confusion matrix heatmap with actual and predicted labels.
axTe.set_xticklabels(labels)
axTe.set_yticklabels(labels)
pltTe.title("Test confusion matrix")
pltTe.xlabel("Predicted")
pltTe.ylabel("Actual")
pltTe.show()
```

```
the maximum value of tpr*(1-fpr) 0.31716043603290306 for threshold 0.836
```




```
the maximum value of tpr*(1-fpr) 0.2900099541519036 for threshold 0.837
```



3. Conclusion

In [156…

```
# Please write down few lines about what you observed from this assignment.
# Please compare all your models using Prettytable library
from prettytable import PrettyTable
```

```
x = PrettyTable()

x.field_names = ["Vectorizer", "Model", "max_depth", "num_boost_round","AUC"]

x.add_row(["wordtovec", "XgBoost  ", 2, 11, 0.5552580828647962 ])

print(x)
```

```
+------------+-----------+-----------+-----------------+--------------------+
| Vectorizer |   Model   | max_depth | num_boost_round |        AUC         |
+------------+-----------+-----------+-----------------+--------------------+
| wordtovec  | XgBoost   |     2     |        11       | 0.5552580828647962 |
+------------+-----------+-----------+-----------------+--------------------+
```