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	
`project_id`	A unique identifier for the proposed project. **Examp
`project_title`	Title of the proje • `Art Will Make` • `Firs`
`project_grade_category`	Grade level of students for which the project is targeted. One enum • `Gra` • • • `(`
`project_subject_categories`	One or more (comma-separated) subject categories for the p following enumerate • `Applie` • `Ca` • `Healt` • `Histo` • `Literacy` • `Math` • `Music` • `Spe` • • `Music` • `Literacy & Language, Math`
`school_state`	State where school is located ([Two-letter U (https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations#F
`project_subject_subcategories`	One or more (comma-separated) subject subcategories • • `Literature & Writing, Socia`
`project_resource_summary`	An explanation of the resources needed for the proj • `My students need hands on literacy material sens`
`project_essay_1`	First ap
`project_essay_2`	Second ap
`project_essay_3`	Third ap
`project_essay_4`	Fourth ap
`project_submitted_datetime`	Datetime when project application was submitted. **Example:** `1`
`teacher_id`	A unique identifier for the teacher of the proposed pro `bdf8baa8fedef6bfeec7ae`

Feature	
`teacher_prefix`	Teacher's title. One of the following enum • • • • • •
`teacher_number_of_previously_posted_projects`	Number of project applications previously submitted by the

[*] 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

Notes on the Essay Data

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

- __project_essay_1:__ "Introduce us to your classroom"
- __project_essay_2:__ "Tell us more about your students"
- __project_essay_3:__ "Describe how your students will use the materials you're requesting"
- __project_essay_4:__ "Close by sharing why your project will make a difference"

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

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

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

In [185]:

```
%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 [186]:

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

In [187]:

```
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_pr
efix' '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_approve
d']
```

In [188]:

```
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[188]:

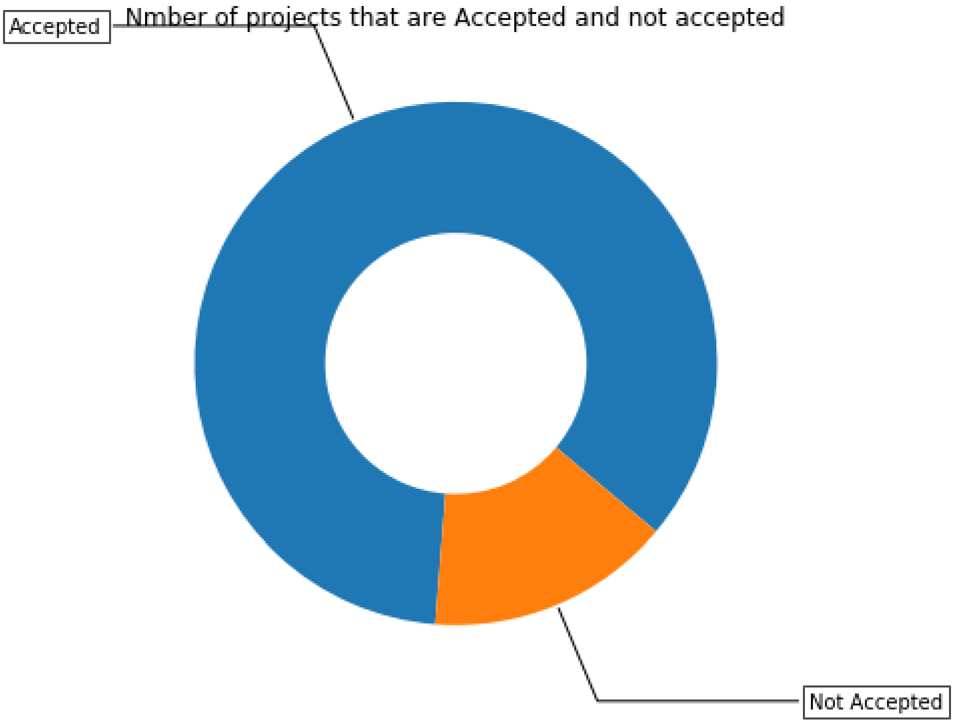
	id	description	quantity	price
0	p233245	LC652 - Lakeshore Double-Space Mobile Drying Rack	1	149.00
1	p069063	Bouncy Bands for Desks (Blue support pipes)	3	14.95

1.2 Data Analysis

In [189]:

```
# PROVIDE CITATIONS TO YOUR CODE IF YOU TAKE IT FROM ANOTHER WEBSITE.
# https://matplotlib.org/gallery/pie_and_polar_charts/pie_and_donut_labels.html#
sphx-glr-gallery-pie-and-polar-charts-pie-and-donut-labels-py


y_value_counts = project_data['project_is_approved'].value_counts()
print("Number of projects thar are approved for funding ", y_value_counts[1], ",
 (", (y_value_counts[1]/(y_value_counts[1]+y_value_counts[0]))*100,"%)")
print("Number of projects thar are not approved for funding ", y_value_counts[0
], ", (", (y_value_counts[0]/(y_value_counts[1]+y_value_counts[0]))*100,"%)")

fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(aspect="equal"))
recipe = ["Accepted", "Not Accepted"]

data = [y_value_counts[1], y_value_counts[0]]

wedges, texts = ax.pie(data, wedgeprops=dict(width=0.5), startangle=-40)

bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
kw = dict(xycoords='data', textcoords='data', arrowprops=dict(arrowstyle="-"),
          bbox=bbox_props, zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1)/2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = "angle,angleA=0,angleB={}".format(ang)
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(recipe[i], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                horizontalalignment=horizontalalignment, **kw)

ax.set_title("Nmber of projects that are Accepted and not accepted")

plt.show()
```

```
Number of projects thar are approved for funding  92706 , ( 84.85830
404217927 %)
Number of projects thar are not approved for funding  16542 , ( 15.1
41695957820739 %)
```



Observations

- This dataset looks imbalanced.
- The majority of the projects (~85%) are accepted and (~15%) are not accepted.

1.2.1 Univariate Analysis: School State

In [190]:

```
# Pandas dataframe groupby count, mean: https://stackoverflow.com/a/19385591/408
4039

temp = pd.DataFrame(project_data.groupby("school_state")["project_is_approved"].
apply(np.mean)).reset_index()
# if you have data which contain only 0 and 1, then the mean = percentage (think
 about it)
temp.columns = ['state_code', 'num_proposals']

'''# How to plot US state heatmap: https://datascience.stackexchange.com/a/9620

scl = [[0.0, 'rgb(242,240,247)'],[0.2, 'rgb(218,218,235)'],[0.4, 'rgb(188,189,22
0)'],\
            [0.6, 'rgb(158,154,200)'],[0.8, 'rgb(117,107,177)'],[1.0, 'rgb(84,3
9,143)']]

data = [ dict(
        type='choropleth',
        colorscale = scl,
        autocolorscale = False,
        locations = temp['state_code'],
        z = temp['num_proposals'].astype(float),
        locationmode = 'USA-states',
        text = temp['state_code'],
        marker = dict(line = dict (color = 'rgb(255,255,255)',width = 2)),
        colorbar = dict(title = "% of pro")
    ) ]

layout = dict(
        title = 'Project Proposals % of Acceptance Rate by US States',
        geo = dict(
            scope='usa',
            projection=dict( type='albers usa' ),
            showlakes = True,
            lakecolor = 'rgb(255, 255, 255)',
        ),
    )

fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='us-map-heat-map')
'''
```

Out[190]:

```
'# How to plot US state heatmap: https://datascience.stackexchange.c
om/a/9620\n\nscl = [[0.0, \'rgb(242,240,247)\'],[0.2, \'rgb(218,218,
235)\'],[0.4, \'rgb(188,189,220)\'],            [0.6, \'rgb(158,154,
200)\'],[0.8, \'rgb(117,107,177)\'],[1.0, \'rgb(84,39,143)\']]\n\nda
ta = [ dict(\n        type=\'choropleth\',\n        colorscale = sc
l,\n        autocolorscale = False,\n        locations = temp[\'stat
e_code\'],\n        z = temp[\'num_proposals\'].astype(float),\n
    locationmode = \'USA-states\',\n        text = temp[\'state_code
\'],\n        marker = dict(line = dict (color = \'rgb(255,255,255)
\',width = 2)),\n        colorbar = dict(title = "% of pro")\n    )
]\n\nlayout = dict(\n        title = \'Project Proposals % of Accept
ance Rate by US States\',\n        geo = dict(\n            scope=
\'usa\',\n            projection=dict( type=\'albers usa\' ),\n
       showlakes = True,\n            lakecolor = \'rgb(255, 255, 25
5)\',\n        ),\n    )\n\nfig = go.Figure(data=data, layout=layou
t)\noffline.iplot(fig, filename=\'us-map-heat-map\')\n'
```

In [191]:

```python
# https://www.csi.cuny.edu/sites/default/files/pdf/administration/ops/2lettersta
bbrev.pdf
temp.sort_values(by=['num_proposals'], inplace=True)
print("States with lowest % approvals")
print(temp.head(5))
print('='*50)
print("States with highest % approvals")
print(temp.tail(5))
```

```
States with lowest % approvals
   state_code  num_proposals
46         VT       0.800000
7          DC       0.802326
43         TX       0.813142
26         MT       0.816327
18         LA       0.831245
==================================================
States with highest % approvals
   state_code  num_proposals
30         NH       0.873563
35         OH       0.875152
47         WA       0.876178
28         ND       0.888112
8          DE       0.897959
```

In [192]:

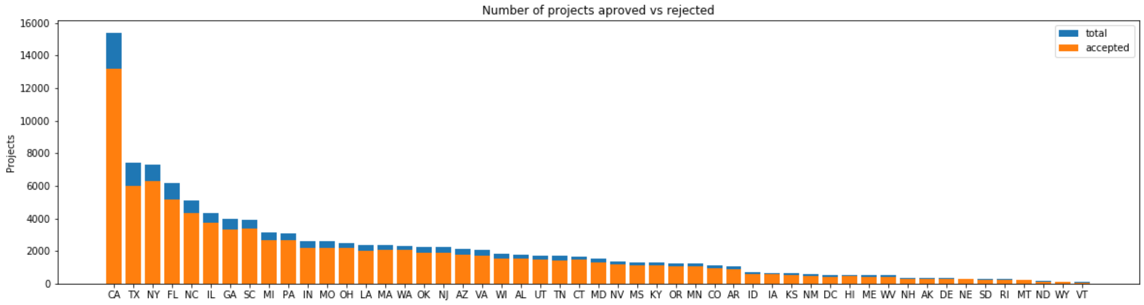
```
#stacked bar plots matplotlib: https://matplotlib.org/gallery/lines_bars_and_markers/bar_stacked.html
def stack_plot(data, xtick, col2='project_is_approved', col3='total'):
    ind = np.arange(data.shape[0])

    plt.figure(figsize=(20,5))
    p1 = plt.bar(ind, data[col3].values)
    p2 = plt.bar(ind, data[col2].values)

    plt.ylabel('Projects')
    plt.title('Number of projects aproved vs rejected')
    plt.xticks(ind, list(data[xtick].values))
    plt.legend((p1[0], p2[0]), ('total', 'accepted'))
    plt.show()
```

In [193]:

```
def univariate_barplots(data, col1, col2='project_is_approved', top=False):
    # Count number of zeros in dataframe python: https://stackoverflow.com/a/51540521/4084039
    temp = pd.DataFrame(project_data.groupby(col1)[col2].agg(lambda x: x.eq(1).sum())).reset_index()

    # Pandas dataframe grouby count: https://stackoverflow.com/a/19385591/4084039
    temp['total'] = pd.DataFrame(project_data.groupby(col1)[col2].agg({'total':'count'})).reset_index()['total']
    temp['Avg'] = pd.DataFrame(project_data.groupby(col1)[col2].agg({'Avg':'mean'})).reset_index()['Avg']

    temp.sort_values(by=['total'],inplace=True, ascending=False)

    if top:
        temp = temp[0:top]

    stack_plot(temp, xtick=col1, col2=col2, col3='total')
    print(temp.head(5))
    print("="*50)
    print(temp.tail(5))
```

In [194]:

```
univariate_barplots(project_data, 'school_state', 'project_is_approved', False)
```



```
   school_state  project_is_approved  total       Avg
4            CA                13205  15388  0.858136
43           TX                 6014   7396  0.813142
34           NY                 6291   7318  0.859661
9            FL                 5144   6185  0.831690
27           NC                 4353   5091  0.855038
==================================================
   school_state  project_is_approved  total       Avg
39           RI                  243    285  0.852632
26           MT                  200    245  0.816327
28           ND                  127    143  0.888112
50           WY                   82     98  0.836735
46           VT                   64     80  0.800000
```

SUMMARY: Every state has greater than 80% success rate in approval

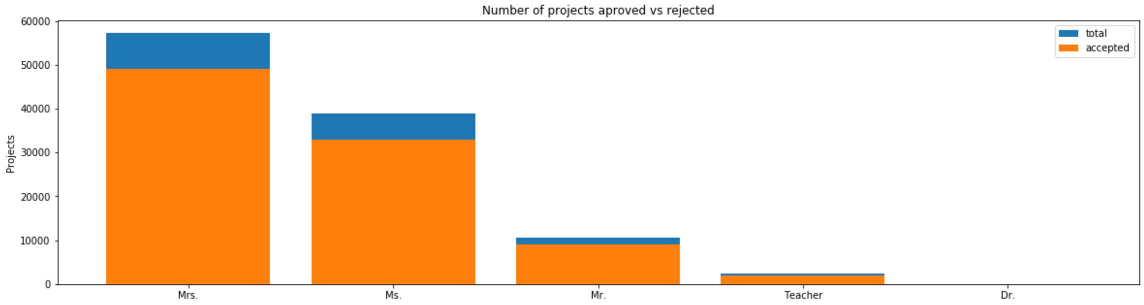
Observations:

- California state seems to have the highest number of project submissions.
- Vermont state has the lowest number of project submissions.

1.2.2 Univariate Analysis: teacher_prefix

In [195]:

```
univariate_barplots(project_data, 'teacher_prefix', 'project_is_approved' , top=
False)
```



```
  teacher_prefix  project_is_approved  total       Avg
2           Mrs.                48997  57269  0.855559
3            Ms.                32860  38955  0.843537
1            Mr.                 8960  10648  0.841473
4        Teacher                 1877   2360  0.795339
0            Dr.                    9     13  0.692308
==================================================
```

```
  teacher_prefix  project_is_approved  total       Avg
2           Mrs.                48997  57269  0.855559
3            Ms.                32860  38955  0.843537
1            Mr.                 8960  10648  0.841473
4        Teacher                 1877   2360  0.795339
0            Dr.                    9     13  0.692308
```

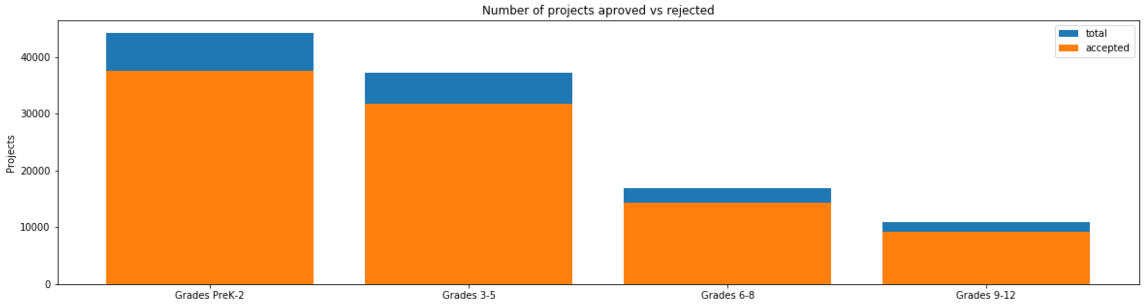
Observations:

- Most of the people have Mrs. as the teacher prefix value (more than 50%).

1.2.3 Univariate Analysis: project_grade_category

In [196]:

```
univariate_barplots(project_data, 'project_grade_category', 'project_is_approve
d', top=False)
```



```
  project_grade_category  project_is_approved  total       Avg
3          Grades PreK-2                37536  44225  0.848751
0             Grades 3-5                31729  37137  0.854377
1             Grades 6-8                14258  16923  0.842522
2            Grades 9-12                 9183  10963  0.837636
==================================================
  project_grade_category  project_is_approved  total       Avg
3          Grades PreK-2                37536  44225  0.848751
0             Grades 3-5                31729  37137  0.854377
1             Grades 6-8                14258  16923  0.842522
2            Grades 9-12                 9183  10963  0.837636
```

Observations

- Most of the project submissions (40000+) are done for the pre kindergarden students.
- Teachers tend to submit more number of projects for kids and less number of projects for older students.

1.2.4 Univariate Analysis: project_subject_categories

In [197]:

```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflow.c
om/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 s
pace "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to r
eplace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empt
y) ex:"Math & Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trail
ing spaces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())
```

In [198]:

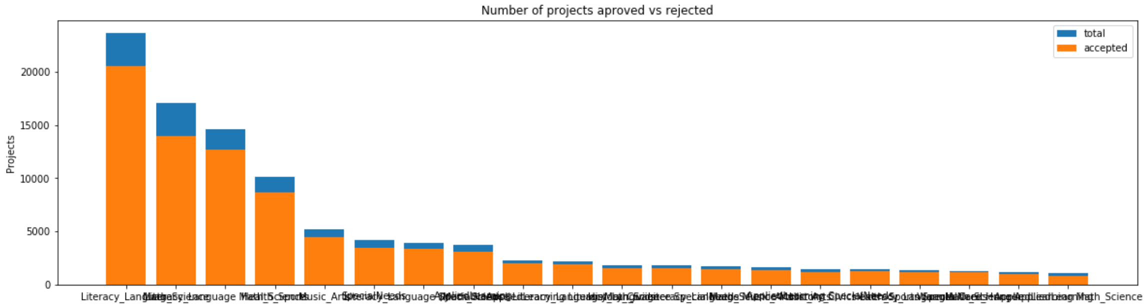
```
project_data['clean_categories'] = cat_list
project_data.drop(['project_subject_categories'], axis=1, inplace=True)
project_data.head(2)
```

Out[198]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	p
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	

In [199]:

```
univariate_barplots(project_data, 'clean_categories', 'project_is_approved', top
=20)
```



```
                  clean_categories  project_is_approved  total
Avg
24               Literacy_Language                20520  23655  0.86
7470
32                    Math_Science                13991  17072  0.81
9529
28  Literacy_Language Math_Science                12725  14636  0.86
9432
8                    Health_Sports                 8640  10177  0.84
8973
40                      Music_Arts                 4429   5180  0.85
5019
==================================================
                    clean_categories  project_is_approved  total
   Avg
19  History_Civics Literacy_Language                 1271   1421  0.
894441
14        Health_Sports SpecialNeeds                 1215   1391  0.
873472
50                Warmth Care_Hunger                 1212   1309  0.
925898
33      Math_Science AppliedLearning                 1019   1220  0.
835246
4       AppliedLearning Math_Science                  855   1052  0.
812738
```

Observations

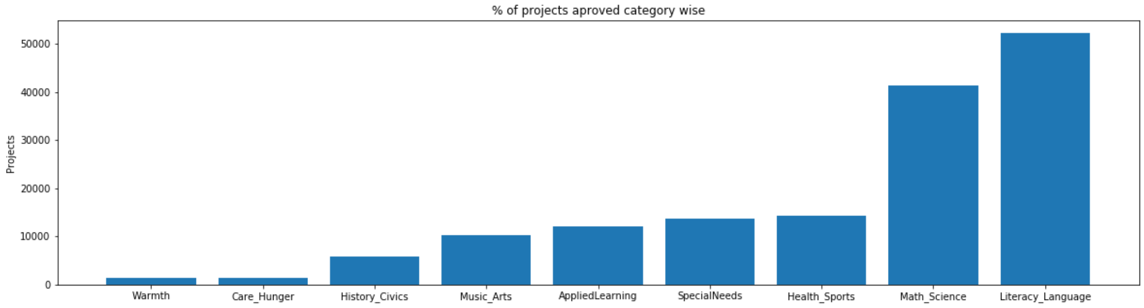
- Most of the project submissions(23655) are done for literacy language.
- Math/science projects seem to need less donations than languages.

In [200]:

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

In [201]:

```
# dict sort by value python: https://stackoverflow.com/a/613218/4084039
cat_dict = dict(my_counter)
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))


ind = np.arange(len(sorted_cat_dict))
plt.figure(figsize=(20,5))
p1 = plt.bar(ind, list(sorted_cat_dict.values()))

plt.ylabel('Projects')
plt.title('% of projects aproved category wise')
plt.xticks(ind, list(sorted_cat_dict.keys()))
plt.show()
```



Observations

- Literacy language projects are more likely to get approved.
- warmth / hunger-care projects are lessl likely to get approved.

In [202]:

```
for i, j in sorted_cat_dict.items():
    print("{:20} :{:10}".format(i,j))
```

```
Warmth               :      1388
Care_Hunger          :      1388
History_Civics       :      5914
Music_Arts           :     10293
AppliedLearning      :     12135
SpecialNeeds         :     13642
Health_Sports        :     14223
Math_Science         :     41421
Literacy_Language    :     52239
```

1.2.5 Univariate Analysis: project_subject_subcategories

In [203]:

```
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())
```

In [204]:

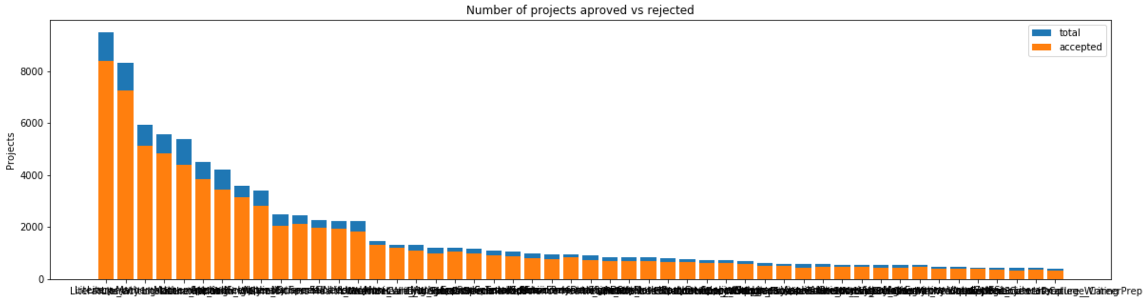
```
project_data['clean_subcategories'] = sub_cat_list
project_data.drop(['project_subject_subcategories'], axis=1, inplace=True)
project_data.head(2)
```

Out[204]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	p
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	

In [205]:

```
univariate_barplots(project_data, 'clean_subcategories', 'project_is_approved',
top=50)
```



```
                clean_subcategories  project_is_approved  total
  Avg
317                        Literacy                 8371   9486  0.8
82458
319            Literacy Mathematics                 7260   8325  0.8
72072
331  Literature_Writing Mathematics                 5140   5923  0.8
67803
318     Literacy Literature_Writing                 4823   5571  0.8
65733
342                     Mathematics                 4385   5379  0.8
15207
==================================================
                    clean_subcategories  project_is_approved  total
      Avg
196     EnvironmentalScience Literacy                   389    444
0.876126
127                                 ESL                   349    421
0.828979
79                   College_CareerPrep                   343    421
0.814727
17   AppliedSciences Literature_Writing                   361    420
0.859524
3    AppliedSciences College_CareerPrep                   330    405
0.814815
```

Observations:

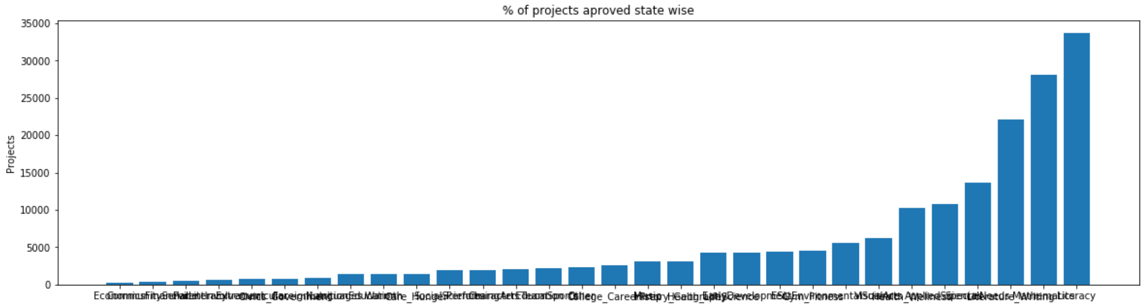
- Projects on rare categories tend to have a better chance (~90%+) of project approvals.
- Teachers are more interested in submitting Literacy & Maths subjects's projects.
- Applied sciences and college preparation projects are the least in number.

In [206]:

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

In [207]:

```
# dict sort by value python: https://stackoverflow.com/a/613218/4084039
sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv: kv[1]))


ind = np.arange(len(sorted_sub_cat_dict))
plt.figure(figsize=(20,5))
p1 = plt.bar(ind, list(sorted_sub_cat_dict.values()))

plt.ylabel('Projects')
plt.title('% of projects aproved state wise')
plt.xticks(ind, list(sorted_sub_cat_dict.keys()))
plt.show()
```



Observations

- Statewise economics seems to have less number project approvals
- Literacy, mathematics, literature writing and applied sciences are people's favorite topics for donation.

In [208]:

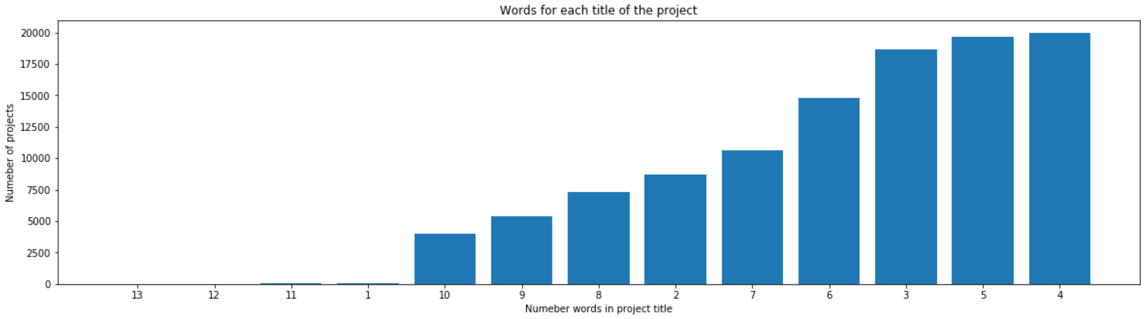
```
for i, j in sorted_sub_cat_dict.items():
    print("{:20} :{:10}".format(i,j))
```

```
Economics            :       269
CommunityService     :       441
FinancialLiteracy    :       568
ParentInvolvement    :       677
Extracurricular      :       810
Civics_Government    :       815
ForeignLanguages     :       890
NutritionEducation   :      1355
Warmth               :      1388
Care_Hunger          :      1388
SocialSciences       :      1920
PerformingArts       :      1961
CharacterEducation   :      2065
TeamSports           :      2192
Other                :      2372
College_CareerPrep   :      2568
Music                :      3145
History_Geography    :      3171
Health_LifeScience   :      4235
EarlyDevelopment     :      4254
ESL                  :      4367
Gym_Fitness          :      4509
EnvironmentalScience :      5591
VisualArts           :      6278
Health_Wellness      :     10234
AppliedSciences      :     10816
SpecialNeeds         :     13642
Literature_Writing   :     22179
Mathematics          :     28074
Literacy             :     33700
```

1.2.6 Univariate Analysis: Text features (Title)

In [209]:

```
#How to calculate number of words in a string in DataFrame: https://stackoverflow.com/a/37483537/4084039
word_count = project_data['project_title'].str.split().apply(len).value_counts()
word_dict = dict(word_count)
word_dict = dict(sorted(word_dict.items(), key=lambda kv: kv[1]))


ind = np.arange(len(word_dict))
plt.figure(figsize=(20,5))
p1 = plt.bar(ind, list(word_dict.values()))

plt.ylabel('Numeber of projects')
plt.xlabel('Numeber words in project title')
plt.title('Words for each title of the project')
plt.xticks(ind, list(word_dict.keys()))
plt.show()
```



Observations

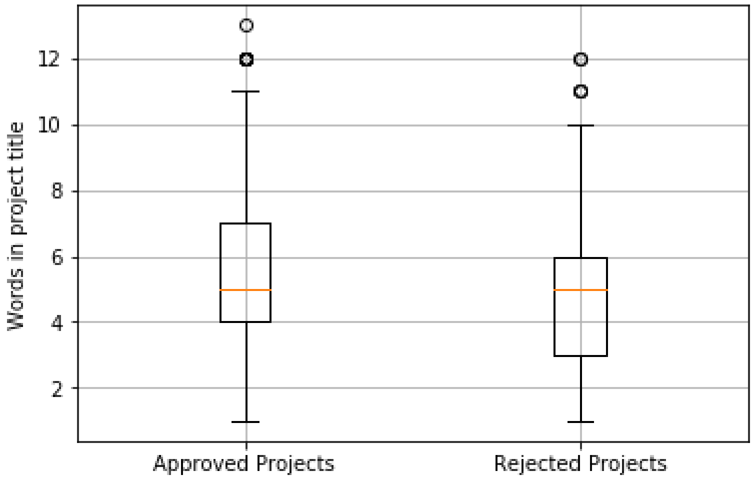
- Most of the projects (~17500 - 20000) have 3 to 5 words in title of the project.
- Very few titles have 1 to 13 words in the title.

In [210]:

```
approved_title_word_count = project_data[project_data['project_is_approved']==1]['project_title'].str.split().apply(len)
approved_title_word_count = approved_title_word_count.values

rejected_title_word_count = project_data[project_data['project_is_approved']==0]['project_title'].str.split().apply(len)
rejected_title_word_count = rejected_title_word_count.values
```

In [211]:

```
# https://glowingpython.blogspot.com/2012/09/boxplot-with-matplotlib.html
plt.boxplot([approved_title_word_count, rejected_title_word_count])
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Words in project title')
plt.grid()
plt.show()
```

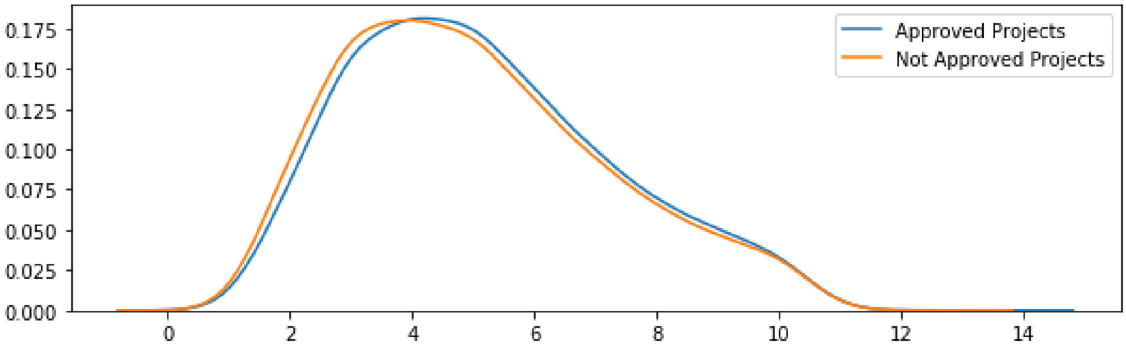


Observations

- Both approved and rejected projects tend to have similar number of words (median is 5) in the title.

In [212]:

```
plt.figure(figsize=(10,3))
sns.kdeplot(approved_title_word_count,label="Approved Projects", bw=0.6)
sns.kdeplot(rejected_title_word_count,label="Not Approved Projects", bw=0.6)
plt.legend()
plt.show()
```



Observations:

- Approved and rejected title projects' word count looks very similar
- Word count doesn't seem to have real effect on project approvals.

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

In [213]:

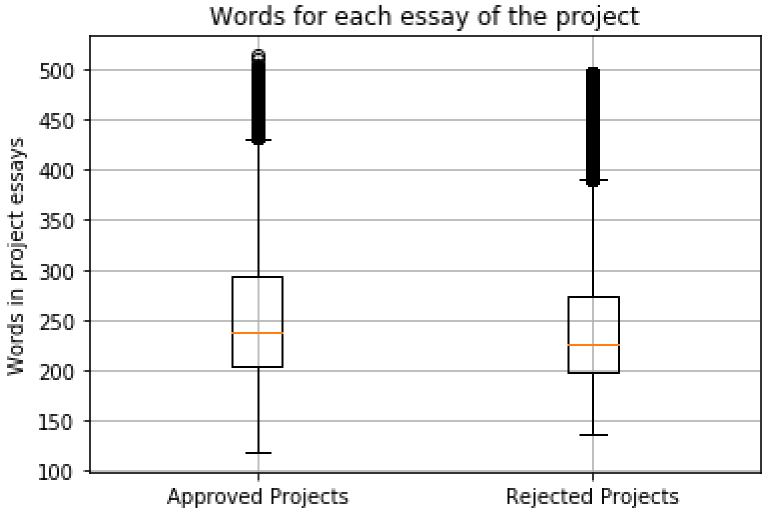
```
# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) +\
                        project_data["project_essay_2"].map(str) + \
                        project_data["project_essay_3"].map(str) + \
                        project_data["project_essay_4"].map(str)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

In [214]:

```
approved_word_count = project_data[project_data['project_is_approved']==1]['essay'].str.split().apply(len)
approved_word_count = approved_word_count.values

rejected_word_count = project_data[project_data['project_is_approved']==0]['essay'].str.split().apply(len)
rejected_word_count = rejected_word_count.values
```

In [215]:

```
# https://glowingpython.blogspot.com/2012/09/boxplot-with-matplotlib.html
plt.boxplot([approved_word_count, rejected_word_count])
plt.title('Words for each essay of the project')
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Words in project essays')
plt.grid()
plt.show()
```

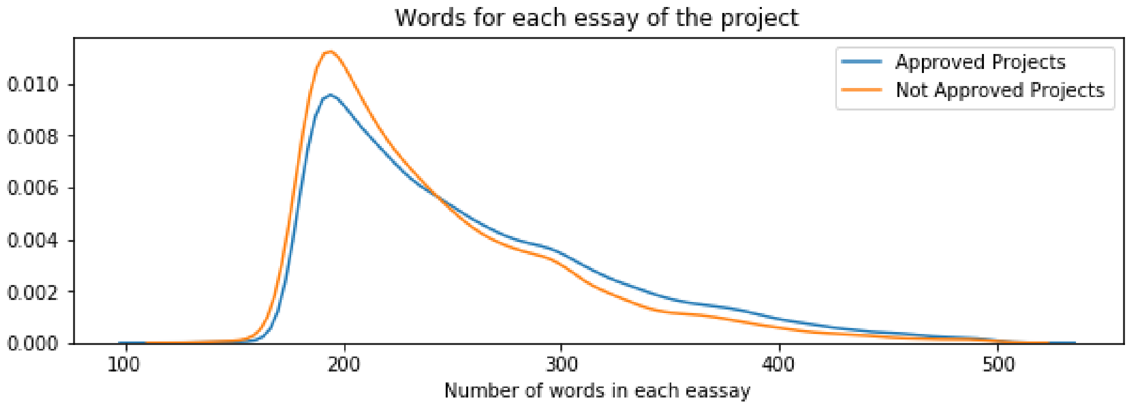


Observations:

- Most of the approved projects have slightly more number of words in the project essay (425+) as compared to rejected projects(380+).

In [216]:

```
plt.figure(figsize=(10,3))
sns.distplot(approved_word_count, hist=False, label="Approved Projects")
sns.distplot(rejected_word_count, hist=False, label="Not Approved Projects")
plt.title('Words for each essay of the project')
plt.xlabel('Number of words in each eassay')
plt.legend()
plt.show()
```



Observations

- Rejected projects' curve seems to fall faster than approved projects' curve. That means approved projects have slightly more words than rejected projects.

1.2.8 Univariate Analysis: Cost per project

In [217]:

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

Out[217]:

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

In [218]:

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

Out[218]:

	id	price	quantity
0	p000001	459.56	7
1	p000002	515.89	21

In [219]:

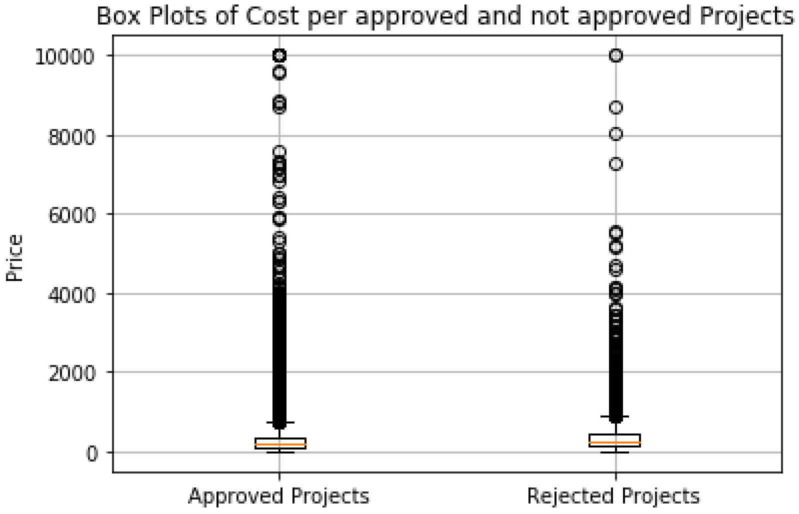
```
# join two dataframes in python:
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

In [220]:

```
approved_price = project_data[project_data['project_is_approved']==1]['price'].v
alues

rejected_price = project_data[project_data['project_is_approved']==0]['price'].v
alues
```

In [221]:

```
# https://glowingpython.blogspot.com/2012/09/boxplot-with-matplotlib.html
plt.boxplot([approved_price, rejected_price])
plt.title('Box Plots of Cost per approved and not approved Projects')
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Price')
plt.grid()
plt.show()
```

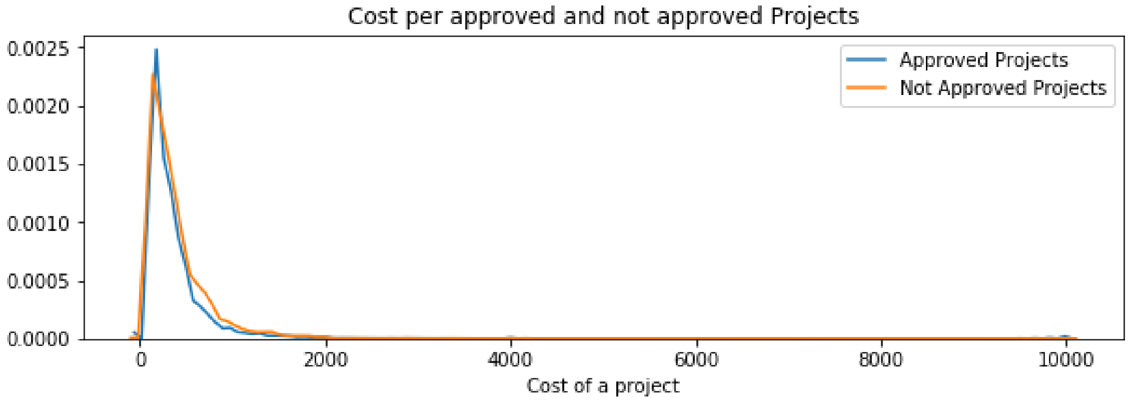


Observations:

- Some of the approved projects have high cost.
- Very few high cost projects seem to have been rejected. So project acceptance/rejection doesn't usually depend on project cost.

In [222]:

```
plt.figure(figsize=(10,3))
sns.distplot(approved_price, hist=False, label="Approved Projects")
sns.distplot(rejected_price, hist=False, label="Not Approved Projects")
plt.title('Cost per approved and not approved Projects')
plt.xlabel('Cost of a project')
plt.legend()
plt.show()
```



Observations:

- Approved projects' curve falls slightly faster than non approved projects. So we can say that approved projects have a little bit low cost for the projects costing between 1000$.

In [223]:

```python
# http://zetcode.com/python/prettytable/
from prettytable import PrettyTable

#If you get a ModuleNotFoundError error , install prettytable using: pip3 install prettytable

x = PrettyTable()
x.field_names = ["Percentile", "Approved Projects", "Not Approved Projects"]

for i in range(0,101,5):
    x.add_row([i,np.round(np.percentile(approved_price,i), 3), np.round(np.percentile(rejected_price,i), 3)])
print(x)
```

```
+------------+-------------------+----------------------+
| Percentile | Approved Projects | Not Approved Projects |
+------------+-------------------+----------------------+
|     0      |       0.66        |         1.97          |
|     5      |      13.59        |         41.9          |
|    10      |      33.88        |        73.67          |
|    15      |       58.0        |        99.109         |
|    20      |      77.38        |        118.56         |
|    25      |      99.95        |       140.892         |
|    30      |      116.68       |        162.23         |
|    35      |     137.232       |       184.014         |
|    40      |      157.0        |       208.632         |
|    45      |     178.265       |       235.106         |
|    50      |      198.99       |       263.145         |
|    55      |      223.99       |        292.61         |
|    60      |      255.63       |       325.144         |
|    65      |     285.412       |        362.39         |
|    70      |     321.225       |        399.99         |
|    75      |     366.075       |       449.945         |
|    80      |      411.67       |       519.282         |
|    85      |      479.0        |       618.276         |
|    90      |      593.11       |       739.356         |
|    95      |     801.598       |       992.486         |
|   100      |      9999.0       |        9999.0         |
+------------+-------------------+----------------------+
```

1.2.9 Univariate Analysis: teacher_number_of_previously_posted_projects

In [224]:

```
# Check values of sample data to understand type of data

project_data.head(2)
```

Out[224]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	proj
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	

Observations

- Looks like the data is numerical

In [225]:

```
# Looking for a sample teacher's previously posted projeccts

project_data[project_data['teacher_id'] == '897464ce9ddc600bced1151f324dd63a']
```

Out[225]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL
9474	157048	p164768	897464ce9ddc600bced1151f324dd63a	Mr.	FL
57583	103042	p094220	897464ce9ddc600bced1151f324dd63a	Mr.	FL
80440	32971	p035710	897464ce9ddc600bced1151f324dd63a	Mr.	FL

Observations:

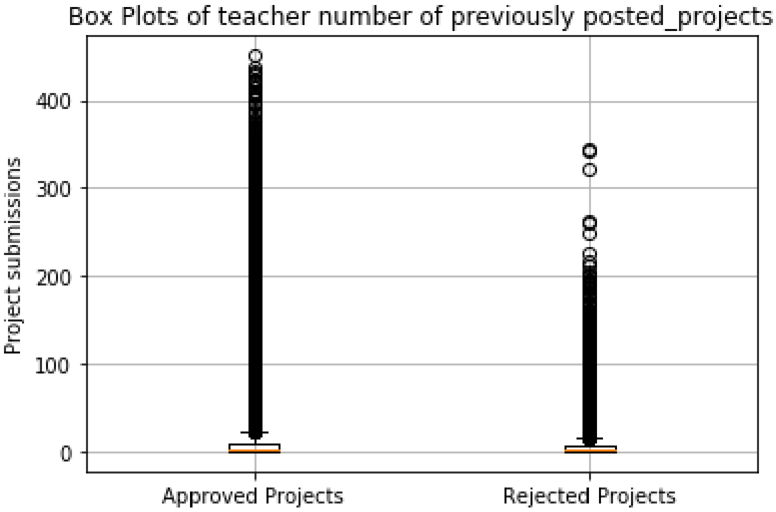
- There are different number of previously posted projects (7 & 12 in our case).
- We can say that teachers have submitted multiple projects in the past, so value of teacher_number_of_previously_posted_projects can be different for the same teacher (depending on time of posting)

In [226]:

```
prev_projects_per_teacher = project_data[['teacher_number_of_previously_posted_p
rojects', 'project_is_approved']]

# Separating approved and non_approved data
approved_projects = prev_projects_per_teacher[prev_projects_per_teacher.project_
is_approved == 1]['teacher_number_of_previously_posted_projects']
rejected_projects = prev_projects_per_teacher[prev_projects_per_teacher.project_
is_approved == 0]['teacher_number_of_previously_posted_projects']
```

In [227]:

```
# https://glowingpython.blogspot.com/2012/09/boxplot-with-matplotlib.html
# Referred earlier example

plt.boxplot([approved_projects, rejected_projects])
plt.title('Box Plots of teacher number of previously posted_projects')
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Project submissions')
plt.grid()
plt.show()
```

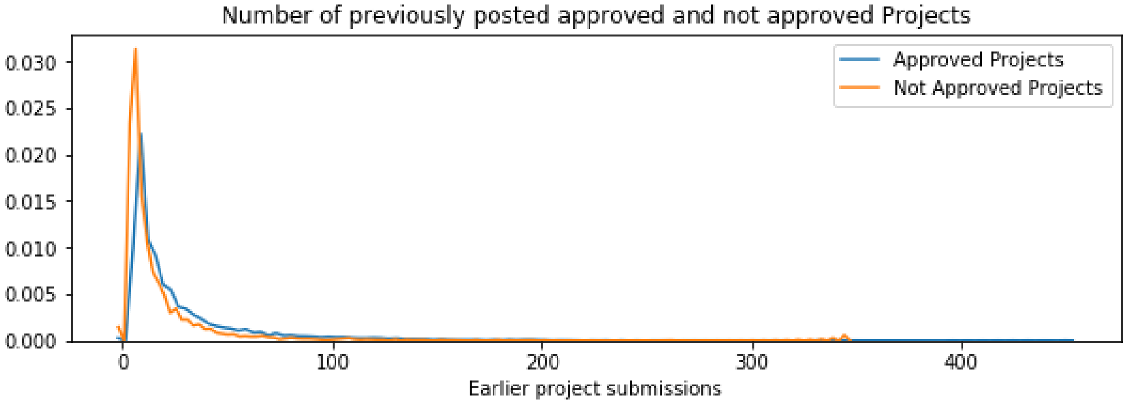


Observations:

- The teachers who had posted many projects earlier seem to have gotten a higher chance of approvals than the ones who didn't.

In [228]:

```
# Referred earlier example

plt.figure(figsize=(10,3))
sns.distplot(approved_projects, hist=False, label="Approved Projects")
sns.distplot(rejected_projects, hist=False, label="Not Approved Projects")
plt.title('Number of previously posted approved and not approved Projects')
plt.xlabel('Earlier project submissions')
plt.legend()
plt.show()
```



Observations

- A lot of the teachers who hadn't posted any projects earlier also got their projects approved.
- If a teacher had posted more than ~350 projects, he had almost 100% chance of getting the projects approved.

In [229]:

```
# http://zetcode.com/python/prettytable/
# Referred earlier example

x = PrettyTable()
x.field_names = ["Percentile", "Approved Projects", "Not Approved Projects"]

for i in range(0,101,5):
    x.add_row([i,np.round(np.percentile(approved_projects,i), 3), np.round(np.pe
rcentile(rejected_projects,i), 3)])
print(x)
```

```
+------------+-------------------+-----------------------+
| Percentile | Approved Projects | Not Approved Projects |
+------------+-------------------+-----------------------+
|     0      |        0.0        |          0.0          |
|     5      |        0.0        |          0.0          |
|     10     |        0.0        |          0.0          |
|     15     |        0.0        |          0.0          |
|     20     |        0.0        |          0.0          |
|     25     |        0.0        |          0.0          |
|     30     |        1.0        |          0.0          |
|     35     |        1.0        |          1.0          |
|     40     |        1.0        |          1.0          |
|     45     |        2.0        |          1.0          |
|     50     |        2.0        |          2.0          |
|     55     |        3.0        |          2.0          |
|     60     |        4.0        |          3.0          |
|     65     |        5.0        |          3.0          |
|     70     |        7.0        |          4.0          |
|     75     |        9.0        |          6.0          |
|     80     |        13.0       |          8.0          |
|     85     |        19.0       |          11.0         |
|     90     |        30.0       |          17.0         |
|     95     |        57.0       |          31.0         |
|    100     |       451.0       |         345.0         |
+------------+-------------------+-----------------------+
```

Observations

- Earlier project submissions have significant impact on approvals.
- More than 80% of the projects got approved for teachers who had posted 13+ projects earlier.
- The teacher having 451 previous project submissions got his all projects approved.
- Project rejections usually happens due to low number of previously posted projects.

1.2.10 Univariate Analysis: project_resource_summary

- Please do this on your own based on the data analysis that was done in the above cells
- Check if the `presence of the numerical digits` in the `project_resource_summary` effects the acceptance of the project or not. If you observe that `presence of the numerical digits` is helpful in the classification, please include it for further process or you can ignore it.

In [230]:

```
# Checking head of file to understand the type of data
project_data.head(2)
```

Out[230]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	p
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	

Observations

- Looks like project_resource_summary field has textual data

In [231]:

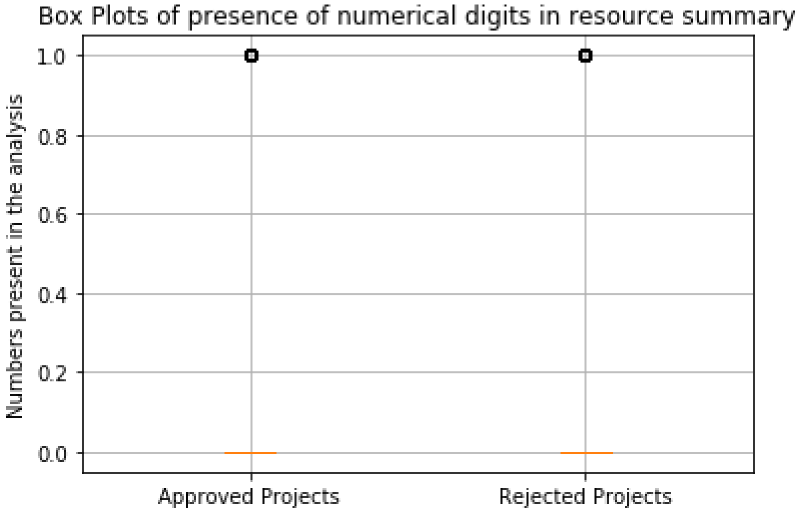
```
# find numbers in text python https://stackoverflow.com/a/4289348/2515354
# presence of regex https://stackoverflow.com/a/9012064/2515354

resource_summary = project_data[["project_resource_summary", "project_is_approve
d"]]
# Adding a new column to indicate the presence of numerical digits (1 - digits a
re present, 0 = digits are absent)
resource_summary["presence_of_the_numerical_digits"] = [1 if re.compile(r'\d+').
search(summary) else 0 for summary in resource_summary["project_resource_summar
y"]]
resource_summary.head()
```

Out[231]:

	project_resource_summary	project_is_approved	presence_of_the_numerical_digits
0	My students need opportunities to practice beg...	0	0
1	My students need a projector to help with view...	1	0
2	My students need shine guards, athletic socks,...	0	0
3	My students need to engage in Reading and Math...	1	0
4	My students need hands on practice in mathemat...	1	0

In [232]:

```
# https://glowingpython.blogspot.com/2012/09/boxplot-with-matplotlib.html
# Referred earlier example for Box plot

approved_projects = resource_summary[resource_summary.project_is_approved == 1][
'presence_of_the_numerical_digits']
rejected_projects = resource_summary[resource_summary.project_is_approved == 0][
'presence_of_the_numerical_digits']

plt.boxplot([approved_projects, rejected_projects])
plt.title('Box Plots of presence of numerical digits in resource summary')
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Numbers present in the analysis')
plt.grid()
plt.show()
```

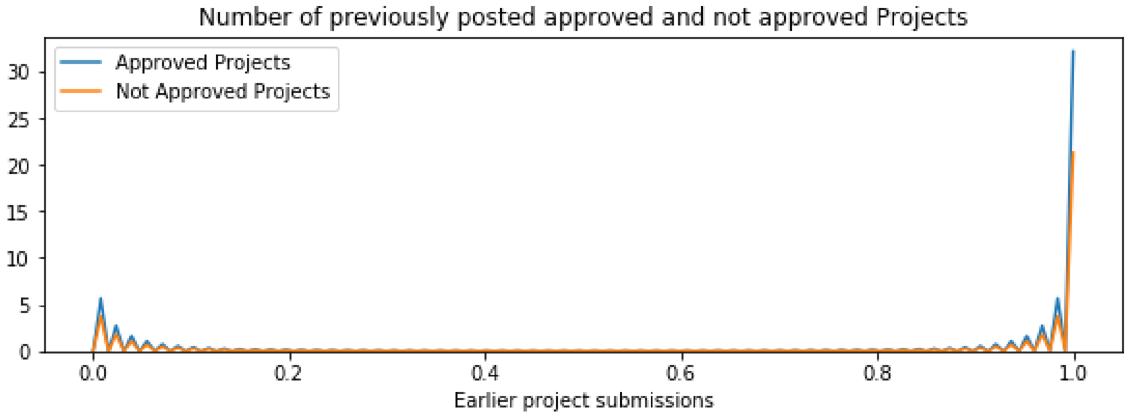


Observations:

- As the values are discrete, above plot doesn't yield any useful information

In [233]:

```
# Referred earlier example

plt.figure(figsize=(10,3))
sns.distplot(approved_projects, hist=False, label="Approved Projects")
sns.distplot(rejected_projects, hist=False, label="Not Approved Projects")
plt.title('Number of previously posted approved and not approved Projects')
plt.xlabel('Earlier project submissions')
plt.legend()
plt.show()
```



In [234]:

```
# http://zetcode.com/python/prettytable/
# Referred earlier example

x = PrettyTable()
x.field_names = ["Percentile", "Approved Projects", "Rejected Projects"]

for i in range(0,101,5):
    x.add_row([i,np.round(np.percentile(approved_projects,i), 3), np.round(np.pe
rcentile(rejected_projects,i), 3)])
print(x)
```

```
+------------+-------------------+-------------------+
| Percentile | Approved Projects | Rejected Projects |
+------------+-------------------+-------------------+
|     0      |        0.0        |        0.0        |
|     5      |        0.0        |        0.0        |
|     10     |        0.0        |        0.0        |
|     15     |        0.0        |        0.0        |
|     20     |        0.0        |        0.0        |
|     25     |        0.0        |        0.0        |
|     30     |        0.0        |        0.0        |
|     35     |        0.0        |        0.0        |
|     40     |        0.0        |        0.0        |
|     45     |        0.0        |        0.0        |
|     50     |        0.0        |        0.0        |
|     55     |        0.0        |        0.0        |
|     60     |        0.0        |        0.0        |
|     65     |        0.0        |        0.0        |
|     70     |        0.0        |        0.0        |
|     75     |        0.0        |        0.0        |
|     80     |        0.0        |        0.0        |
|     85     |        1.0        |        0.0        |
|     90     |        1.0        |        1.0        |
|     95     |        1.0        |        1.0        |
|    100     |        1.0        |        1.0        |
+------------+-------------------+-------------------+
```

Observations

- We're getting pretty similar, as roughly 80% approved projects don't have digits in project resource summary whereas 85% rejected projected don't have digits in project resource summary as well.

1.3 Text preprocessing

1.3.1 Essay Text

In [235]:

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

Out[235]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	proj
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	

In [236]:

```
# printing some random essays.
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 thei
r second or third languages. We are a melting pot of refugees, immig
rants, and native-born Americans bringing the gift of language to ou
r school. \r\n\r\n We have over 24 languages represented in our Engl
ish Learner program with students at every level of mastery.  We als
o have over 40 countries represented with the families within our sc
hool.  Each student brings a wealth of knowledge and experiences to 
us that open our eyes to new cultures, beliefs, and respect.\"The li
mits of your language are the limits of your world.\"-Ludwig Wittgen
stein  Our English learner's have a strong support system at home th
at begs for more resources.  Many times our parents are learning to 
read and speak English along side of their children.  Sometimes this
creates barriers for parents to be able to help their child learn ph
onetics, letter recognition, and other reading skills.\r\n\r\nBy pro
viding these dvd's and players, students are able to continue their 
mastery of the English language even if no one at home is able to as
sist.  All families with students within the Level 1 proficiency sta
tus, will be a offered to be a part of this program.  These educatio
nal videos will be specially chosen by the English Learner Teacher a
nd will be sent home regularly to watch.  The videos are to help the
child develop early reading skills.\r\n\r\nParents that do not have 
access to a dvd player will have the opportunity to check out a dvd 
player to use for the year.  The plan is to use these videos and edu
cational dvd's for the years to come for other EL students.\r\nnanna
n
==================================================
The 51 fifth grade students that will cycle through my classroom thi
s year all love learning, at least most of the time. At our school, 
97.3% of the students receive free or reduced price lunch. Of the 56
0 students, 97.3% are minority students. \r\nThe school has a vibran
t community that loves to get together and celebrate. Around Hallowe
en 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 c
rafts made by the students, dances, and games. At the end of the yea
r the school hosts a carnival to celebrate the hard work put in duri
ng the school year, with a dunk tank being the most popular activit
y.My students will use these five brightly colored Hokki stools in p
lace of regular, stationary, 4-legged chairs. As I will only have a 
total of ten in the classroom and not enough for each student to hav
e an individual one, they will be used in a variety of ways. During 
independent reading time they will be used as special chairs student
s will each use on occasion. I will utilize them in place of chairs 
at my small group tables during math and reading times. The rest of 
the day they will be used by the students who need the highest amoun
t of movement in their life in order to stay focused on school.\r\n
\r\nWhenever asked what the classroom is missing, my students always
say more Hokki Stools. They can't get their fill of the 5 stools we 
already have. When the students are sitting in group with me on the 
Hokki Stools, they are always moving, but at the same time doing the
ir work. Anytime the students get to pick where they can sit, the Ho
kki Stools are the first to be taken. There are always students who 
head over to the kidney table to get one of the stools who are disap
pointed as there are not enough of them. \r\n\r\nWe ask a lot of stu
dents to sit for 7 hours a day. The Hokki stools will be a compromis
e that allow my students to do desk work and move at the same time. 
These stools will help students to meet their 60 minutes a day of mo
vement by allowing them to activate their core muscles for balance w
hile they sit. For many of my students, these chairs will take away 
the barrier that exists in schools for a child who can't sit still.n
annan
==================================================
How do you remember your days of school? Was it in a sterile environ
ment with plain walls, rows of desks, and a teacher in front of 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 c
oming to each day.\r\n\r\nMy class is made up of 28 wonderfully uniq
ue boys and girls of mixed races in Arkansas.\r\nThey 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 c
lassrooms. These 9 and 10 year-old students 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 able to help create the mood in our classroom s
etting to be one of a themed nautical environment. Creating a classr
oom environment is very important in the success in each and every c
hild's education. The nautical photo props will be used with each ch
ild as they step foot into our classroom for the first time on Meet
the Teacher evening. I'll take pictures of each child with them, hav
e them developed, and then hung in our classroom ready for their fir
st day of 4th grade.  This kind gesture will set the tone before eve
n the first day of school! The nautical thank you cards will be used
throughout the year by the students as they create thank you cards t
o their team groups.\r\n\r\nYour generous donations will help me to
help make our classroom a fun, inviting, learning environment from d
ay one.\r\n\r\nIt costs lost of money out of my own pocket on resour
ces to get our classroom ready. Please consider helping with this pr
oject to make our new school year a very successful one. Thank you!n
annan
==================================================
My kindergarten students have varied disabilities ranging from speec
h and language delays, cognitive delays, gross/fine motor delays, to
autism. They are eager beavers and always strive to work their harde
st working past their limitations. \r\n\r\nThe materials we have are
the ones I seek out for my students. I teach in a Title I school whe
re most of the students receive free or reduced price lunch.  Despit
e their disabilities and limitations, my students love coming to sch
ool 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 answ
er and I love then because they develop their core, which enhances g
ross motor and in Turn fine motor skills. \r\nThey also want to lear
n through games, my kids don't want to sit and do worksheets. They w
ant 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 ju
st have the 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-Am
erican, making up the largest segment of the student body. 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 do
ctors, lawyers, or engineers children from rich backgrounds or neigh
borhoods. As an educator I am inspiring minds of young children and
we focus not only on academics but one smart, effective, efficient,
and disciplined students with good character.In our classroom we can
utilize the Bluetooth for swift transitions during class. I use a sp
eaker which doesn't amplify the sound enough to receive the message.
```

```
Due to the volume of my speaker my students can't hear videos or boo
ks clearly and it isn't making the lessons as meaningful. But with t
he bluetooth speaker my students will be able to hear and I can sto
p, pause and replay it at any time.\r\nThe cart will allow me to hav
e more room for storage of things that are needed for the day and ha
s an extra part to it I can use.  The table top chart has all of the
letter, words and pictures for students to learn about different let
ters and it is more accessible.nannan
==================================================
```

In [237]:

```
# 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 [238]:

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

```
My kindergarten students have varied disabilities ranging from speec
h and language delays, cognitive delays, gross/fine motor delays, to
autism. They are eager beavers and always strive to work their harde
st working past their limitations. \r\n\r\nThe materials we have are
the ones I seek out for my students. I teach in a Title I school whe
re most of the students receive free or reduced price lunch.  Despit
e their disabilities and limitations, my students love coming to sch
ool 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 answ
er and I love then because they develop their core, which enhances g
ross motor and in Turn fine motor skills. \r\nThey also want to lear
n through games, my kids do not want to sit and do worksheets. They
want to learn to count by jumping and playing. Physical engagement i
s the key to our success. The number toss and color and shape mats c
an make that happen. My students will forget they are doing work and
just have the fun a 6 year old deserves.nannan
==================================================
```

In [239]:

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

```
My kindergarten students have varied disabilities ranging from speec
h and language delays, cognitive delays, gross/fine motor delays, to
autism. They are eager beavers and always strive to work their harde
st 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 m
ost of the students receive free or reduced price lunch.  Despite th
eir 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 abl
e to move as they learn or so they say.Wobble chairs are the answer
and I love then because they develop their core, which enhances gros
s motor and in Turn fine motor skills.   They also want to learn thr
ough 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 color and shape mats can mak
e that happen. My students will forget they are doing work and just
have the fun a 6 year old deserves.nannan
```

In [240]:

```
#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 speec
h and language delays cognitive delays gross fine motor delays to au
tism 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 t
he students receive free or reduced price lunch Despite their disabi
lities and limitations my students love coming to school and come ea
ger to learn and explore Have you ever felt like you had ants in you
r pants and you needed to groove and move as you were in a meeting T
his 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 th
en because they develop their core 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 Th
e number toss and color and shape mats can make that happen My stude
nts will forget they are doing work and just have the fun a 6 year o
ld deserves nannan
```

In [241]:

```
# 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', 'itse
lf', 'they', 'them', 'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'tha
t', "that'll", 'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'ha
s', 'had', 'having', 'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'becaus
e', 'as', 'until', 'while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 't
hrough', 'during', 'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'of
f', 'over', 'under', 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'al
l', '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'v
e", 'now', 'd', 'll', 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "d
idn'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', "should
n't", 'wasn', "wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [242]:

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

```
100%|██████████| 109248/109248 [00:56<00:00, 1924.81it/s]
```

In [243]:

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

Out[243]:

```
'my kindergarten students varied disabilities ranging speech languag
e delays cognitive delays gross fine motor delays autism they eager 
beavers always strive work hardest working past limitations the mate
rials ones i seek students i teach title i school students receive f
ree reduced price lunch despite disabilities limitations students lo
ve coming school come eager learn explore have ever felt like ants p
ants needed groove move meeting this kids feel time the want able mo
ve learn say wobble chairs answer i love develop core enhances gross
motor turn fine motor skills they also want learn games kids not wan
t sit worksheets they want learn count jumping playing physical enga
gement key success the number toss color shape mats make happen my s
tudents forget work fun 6 year old deserves nannan'
```

1.3.2 Project title Text

In [244]:

```
# remove unicode characters python https://stackoverflow.com/a/48900582/2515354
# https://gist.github.com/sebleier/554280
# Referred decontracted function from above sample code

cleaned_titles = []

for title in tqdm(project_data['project_title']):
    clean_title = decontracted(title)
    clean_title = clean_title.strip()
    clean_title = clean_title.replace('\\r', '')
    clean_title = clean_title.replace('\\n', '')
    clean_title = clean_title.replace('&', 'and')
    clean_title = re.sub(r'[\\\/\(\)\[\],:=.\-_\!\"\'\?\*\+]*', '', clean_title)

    clean_title = clean_title.lower()
    clean_title = (clean_title.encode('ascii', 'ignore')).decode("utf-8")
    clean_title = ' '.join(e for e in clean_title.split() if e not in stopwords)
    cleaned_titles.append(clean_title)

cleaned_titles
```

```
100%|██████████| 109248/109248 [00:03<00:00, 35842.43it/s]
```

Out[244]:

```
['educational support english learners home',
 'wanted projector hungry learners',
 'soccer equipment awesome middle school students',
 'techie kindergarteners',
 'interactive math tools',
 'flexible seating mrs jarvis terrific third graders',
 'chromebooks special education reading program',
 '21st century',
 'targeting success class',
 'love readingpure pleasure',
 'reading changes lives',
 'elevating academics parent rapports technology',
 'building life science experiences',
 'everyone deserves heard',
 'tablets show us world',
 'making recess active',
 'making great leap leapfrog',
 'technology teaches tomorrow talents today',
 'test time',
 'wiggling way success',
 'magic carpet ride library',
 'sitting standing classroom',
 'books budding intellectuals',
 'instrumental power conquering steam',
 'steam challengesscience technology engineering art math',
 'math masters',
 'techy teaching',
 '4th grade french immersion class ipads',
 'handson language literacy',
 'basic classroom supplies needed',
 '2nd grade explores world charlotte web',
 'inclusive learning space',
 'learning facts fiction',
 'computing way financial literacy part 2',
 'ball',
 'put coach',
 'inquiry based discovery laptop learning',
 'target kids printer ink',
 'kinders inspired target fitness part one',
 'engaging students technology',
 'leveling books multiage class',
 'twist writing traits first graders',
 'need nonfiction',
 'hands tech',
 'pressing mastery flood',
 'chromebooks create intrigue motivation filling gaps',
 'paper',
 'keep closet open',
 'chromebook gold',
 'rainy day run around',
 'active energized',
 'great books clean organized filing cabinets successful students',
 'stand learn',
 'reading together',
 'swim life ymca',
 'stem need capitalize technology',
 'first coffee',
 'mouse hunt',
 'awesome authors need terrific table',
 'interactive assessments',
 'picnic table make us able',
 'forming magnificent minds',
 'adding interactive technology young writers toolbox',
 'need paper ink new year',
 'read art',
 'keep computer lab nice',
 'science technology engineering math oh kinder stem',
 'magic reading',
 'stem made simple sensible integrated meaningful purposeful learnin
g engaging',
 'chrome class',
 'immersion trip outdoor gear',
 'magnets electricity live',
 'gotta catch chromebook',
 'college signing day rally prizes deserving students',
 'student seating paradise',
 'jump music',
 'comfortable place learn',
 'growth mindset future',
 'standup desks mrs brown class',
 'make music make year',
 'crazy computers',
 'need seat',
 'fall love reading',
 'classroom books inspire',
 'planes trains andsteam',
 'technology bust',
 'targeting love baseball hitting bullseye',
 'exploring graphic novels',
 'read understand like everyone else',
 'education technology',
 'publishers need printer',
 'chicka chicka boom boom help us cool classroom',
 'help keep us motivated',
 'new music stands benton middlehigh school band',
 'new literacy unit books',
 'ilearn igrow isucceed ipads',
 'leveled books everyone',
 'drseuss others help us read',
 'buttons bulldogs',
 'teaching math manipulatives',
 '21st century learners 21st century technology',
 'fun physically fit',
 'project light',
 'buzzing books',
 'pehealth technology',
 'ceramics history clay sculpture',
 'louisiana flooded classroom',
 'paper pencils markers oh',
 'easy 123',
 'start year strong',
 'super supplies',
 '1st graders reaching stars reading writing',
 'writer workshop 1st grade authors',
 'technology classroom',
 'write way',
 'paperpaperpaper please',
 'diminish digitial divide',
 'global learners taking lead',
 'magical morphing exploring wondrous life cycle butterflies',
 'use marbles stimulate brain',
 'flexible seating',
 'creating sense community',
 'help us ball pe',
 'book em danno',
 'easy eyes nose',
 'student materials needed',
 'ipad supplies room',
 'gone chopin bach five',
 'without string world silent',
 'bring projects life color',
 'learning words listening weekly',
 'not love lego books',
 'deep heart texas',
 'make move',
 'music please motivation needed vigorous physical activity',
 'art 21st century',
 'sand water word work fun',
 'not wander lostjrrtolkien',
 'kindergarten stem stations',
 'first grade cool coding',
 'classroom library exceptional students',
 'learning historyholocaust',
 'want shake wobble bounce',
 'ready go macbook pro',
 'book bins',
 'spreading love one card time',
 'ditch white board get boogie board',
 'needs chromebookwe',
 'steam technology inquiry based learning',
 'flexible seating',
 'movement hokki stools',
 'need kleen slate',
 '6th graders think like engineers',
 'shhhhh working independently learning center',
 'national parks outer space',
 'flexible seating optimal learning',
 'getting move',
 'safely searching critters',
 'oceanbots deep sea explorers',
 'lego robotics programming resources digital media mics',
 'help us organize classroom library',
 'history comes life mrs butler 5th grade class',
 'healthy bodies dirty hands let kids kids',
 'hear help students hear heard',
 'hovering hovergraft',
 'pedaling way day',
 'team practices make perfect teams',
 'students crazy apple watches',
 'growing future programmers',
 'scientific calculators',
 'alternative seating comfy classrooms',
 'tablets rescue',
 'colorful learning environment',
 'tech knowledge',
 'wobble work',
 'reading together fun',
 'magic steam prek',
 'education seating choice flexible seating classroom',
 'globe gazing students want see world',
 'learning technology',
 'pitter ipadder',
 'teachers love video games',
 'testing point',
 'calculators kids',
 'building math life skills future',
 'becoming architects engineers builders age 6',
 'flexible seating third',
 'ineeds support steam',
 'endless possibilities',
 'building grade level chromebooks',
 'seating active learners',
 'providing learning environment kids need',
 'headphones kiddos',
 'operation 60',
 'projecting way future',
 'amazing anchor charts',
 'behavior technology match made heaven',
 'sitting pretty science lab',
 'foundations prek writing',
 'write author',
 'sensory toys make sense world',
 'ring bells hear',
 'chromebooks stem',
 'balls bubbles birthday books',
 'flexible seating flexible classroom',
 'safe books bouncy balls',
 'like hot like cold',
 '5th grade life science owl pellet dissection project',
 'perfect position 5th grade orchestra',
 'urban prek5 technology classroom looking 3d printer',
 'learning fun rewarding',
 'music knowledge go hand hand part 2',
 'help feed newington elementary school students',
 'classroom projection 21st century way',
 'calculating success',
 'scope help us become wellrounded readers',
 'savvy stem startup using robust robotics',
 '21st century technology needed',
 'help us bring music home part 2',
 'robots would lives like without',
 'sew happy',
 'chevron solve create ipads',
 'taking hyper hyperactivity',
 'students need supplies',
 'science success',
 'little wiggle room',
 'wiggle seating kinder kids',
 '7th grade action researchers',
 'pick us play tune',
 'current events classroom',
 'open window technology',
 'help us develop reader workshop',
 'ipad learning',
 'rollin river',
 'fidget toys chairs middle school kids',
 'guided reading resources',
 'fueling bodies minds right start',
 'enhancing minds research inquiry',
 'amplifying student learning one ipad time',
 'flexible seating',
 'listening center little learners',
 '21st century students 21st century classroom',
 'art enhances learning',
 'future bright technology',
 'technology environment responsibility',
 'wiggle work',
 'get recharged',
 'let children play',
 'life essentials',
 'cul es la ecuacin de esta lnea',
 'classroom carpet',
 'flexible seating creating 21st century learning environment',
 'robotics futureinteractive minds create',
 'board not bored',
 'bearcat chemistry',
 'autodesk inventor comes alive 8gb memory computers ohigh',
 'writing towards success',
 'see learning',
 'help us hokki pokie',
 'journey new exciting places',
 'ray cleanliness',
 'sharpen pencils',
 'becoming literate citizens',
 'reading classics class disappear forever',
 'go go google gadgets',
 'lifting weights lifting spirits',
 'book month program',
 'technology makes learning meaningful',
 'staying date',
 'amazing student work binders',
 'document camera present dissections projects diagrams lessons',
 'books books books',
 'fire learning amazon fire tablet',
 'handson learning stems',
 'classroom pets fish tadpoles turtles chameleons hermit crabs',
 'healthy snack attacks',
 'flexible seating fun',
 'stay fit exercise spark 2',
 'optimizing reading growth accelerated reader',
 'super star second graders',
 'multiplying efforts flood',
 'like moveit moveit',
 'oysters looking pearls',
 'classroom essentials',
 'prototyping help others',
 'super scholars accelerating towards excellence',
 'audio books students visual impairments',
 'furniture firsties',
 'oodles outdoor fun',
 'learning shred',
 'ap literature success new novels',
 'social studies first',
 'chromebooks fantastic 1st graders',
 'gamification learning',
 'kindergarten stem grow',
 'guitar tuners',
 'band basics create music',
 'mindfulness essential oils',
 'reading learning friendship',
 'tablets inspire middle school math minds',
 'biology interactive learning log',
 'littlebits big learning',
 'organization express train',
 'ordinary finds extraordinary minds',
 'special supplies bilingual students',
 'creativity intelligence fun bobcats steam ahead',
 'active recesslet get moving',
 'wobble work',
 'ya make 3d objects',
 'ap chemistry prep books',
 'take ballgame',
 'learn comfort',
 'like move move itflexible seating',
 'moving robots',
 'chromebook learning',
 'reading front lines 7th grade',
 'healthy lives archery',
 'tablet tech',
 'handwriting without tears',
 'stem twentyfirst century students',
 'getting comfy classroom library',
 'chromebooks chromebooks chromebooks everywhere',
 'bring color',
 'education sweet nice seat',
 'teach students remember engage students learn',
 'technology enhances learning',
 'kids coding creativity',
 'diving microscopic world',
 'collaborate chrome',
 'grab seat go',
 'help us rock learn',
 'learning use technology one ipad time',
 'wiggly worms',
 'splendid science',
 'balance balls balanced learning experience',
 'strike band',
 'ipad art room',
 'leveled readers happy students',
 'ebooks rus',
 'flexible seating focused students',
 'pretty presentations',
 'proficiency scientific presentations',
 'walls wiggly students need wiggly seats',
 'developing love reading part 3',
 'graphing calculators higher mathematics',
 'phonics reading club',
 'getting comfy engaged new carpet',
 'computer science math class',
 'no ordinary organizer',
 'turn frowny faces upside',
 'cleaning classroom library',
 'mini ipad huge difference',
 'poetry celebration',
 'active classroom',
 'break tech learn cooperatively',
 'seeking sensational supplies',
 'stopsafety patrol',
 'supplies success',
 'flexible seating flexible learning space',
 'hydroponic garden',
 'classroom rug ms clark class',
 'starting sounds words',
 'tech savvy third graders need tablets',
 'recording live music macbook pro',
 'books grow gradelevel readers',
 'flexible seating',
 'supplies needed',
 'laughs learning poetry',
 'meeting individual needs one scribble time',
 'algebra 1 supplies',
 'world end study dystopian literature',
 'today reader tomorrow leader',
 'stem learning brought life',
 '1st grade wise owls',
 'movement towards healthy lifestyle',
 'ties bind custom built writing portfolios',
 'full steam ahead complete chromebook cart',
 'help journalism students go pro',
 'reading takes greatest adventure',
 'face facts developing nonfiction classroom library',
 'take seat',
 'stand move',
 'hela cells',
 'pencils notebookand folders please',
 'classroom rugs center learning first grade',
 'kids programcode dash dot robots #csforall #hourofcode',
 'mind math',
 'creativity critical thinking interactive technology',
 'math tools classroom',
 'technology music classroom',
 'building forever readers',
 'chromebooks build confidence english language learners',
 'boxing way academic success',
 'movement freedom',
 'urban garden grows interest environmental science',
 'kindergarteners love wobble',
 'books carriers kindergarten literacy centers',
 'taking display student work next level',
 'chapter books third graders',
 'wireless tech developing journalists',
 'leave better found',
 'bean bag pod',
 'crisscross applesauce',
 'time graduate need textbook',
 'calc kids need calculators',
 'wobble away 2nd grade',
 'read lead succeed',
 'no squeaks squawks woodwind mouthpieces needed',
 'students need think feet',
 'stem kindergarten',
 'one book two books red book blue book',
 'letters numbers come life',
 'creating 3rd grade community learners',
 'technology alternative classroom experience',
 'chrome needs polishing order sparkle',
 'reading math helps mind bloom',
 'tune makes lesson better class',
 'operation graphic design',
 'smart tv needed smart music students',
 'louisiana flooded students growing giftedness',
 'extra extra classroom supplies needed',
 'landmark art',
 'mini ipads awesome 2nd grade learners',
 'explore tubs',
 'comfort classroom success',
 'fostering social emotional development multicultural prek class',
 'fourth r recess',
 'experience another dimension math 3d printer',
 'silence golden',
 'wobble wiggles away',
 'keeping newark fit',
 'extraordinary students need technology',
 'connecting beyond classroom',
 'graphic novels reading',
 'fun school',
 'neighborhood work',
 'get moving get cozy get learning',
 'need technology middle school',
 'superhero literacy',
 'hiho hiho need osmo',
 'making students feel home cozy classroom',
 'classroom library needs books',
 'technology finger tips',
 'keep calm use cromebook',
 'desktops desktops desktops',
 'loud think printer without ink technology sink',
 'flexible seating activity rug promote active healthy individuals',
 'need move move',
 'math must haves',
 'paramount technology integration',
 'personalized science notebooks',
 'bistro style library',
 'exploring science stem experiments',
 'sixth graders need book club books',
 'classroom students want',
 'learning flexible classroom',
 'mrs esposito class loves learning current events',
 'love literacy',
 'food fuel learning',
 'light kindle fire learning',
 'community graffiti wordle',
 'tablets third grade',
 'center time',
 'hula hoop moving groovin',
 'goodbye desk chairs',
 'get kinders fired reading',
 'dress play',
 'take seatlearning neat',
 'know h20 groundwater quality testing',
 'learn like 2099',
 'bounce learning',
 'tablets individualized learning',
 'chemists chrome books',
 'flexible seating flexible learning',
 'listening center 4 daily 5',
 'decreasing reading gap level',
 'math skills keep getting hotter hot dots',
 'play time first step learning',
 'math fingertips',
 'technology mrs wahlberg class',
 'mathematicians ahead',
 'making students centered learning',
 'creative kinders',
 'abcs kindergarten literacy materials',
 'need organized classroom',
 'world classroom',
 'painting outside part ii',
 'need amore speech therapy materials',
 'fidgets help us focus',
 'ifit going gold part ii',
 'critical thinking sensory play',
 'learning beyond classroom',
 'building community one recess game time',
 'robots stem education san bernardino',
 'super scientists',
 'mindblowing mathmotivating young mathematicians',
 'computers explore',
 'staying active indoor recess',
 'apple harvest knowledge farmers',
 'novels reach new levels',
 'classroom chromebooks college bound seniors',
 'stemulating lab phase ii',
 'teaching daily living skills special needs children',
 'clean tidy ready learn',
 'listen learn',
 'help wiggles',
 'burn calories desk',
 'pedaling proficiency pedal seats alternative seating options',
 'inspiring readers writers technology',
 'flexible seating',
 'fun pe equipment',
 'making insiders outsiders',
 'inspiring stem activities kindergarten',
 'ilearn ipads',
 'innovative providing tools interactive engagement',
 'demonstration tools learning fun',
 'wiggle roomflexible seating options small groups',
 'transforming stationary learning active movement opportunities',
 'moving grooving 5th grade',
 'taking care bodies one less concern',
 'chrome zone',
 'coding kindergarten',
 'let accessorize',
 'teamwork preschool',
 'side fairytale',
 'organization planning keys success',
 'math reading needing',
 'read teach repeat',
 'angry birds physics',
 'google apps helps us create',
 'stem k2',
 'making pens pencils others',
 'stand ipads',
 'daily road maps children',
 'getting staying healthy',
 'modeling multiple learning styles',
 'creative technology',
 'hands math science tools superhero class',
 'adventurous amazing books library',
 'ict class needs chromebook',
 'coding sphero',
 'imagination digital storytelling',
 'supplies not limiting factor',
 'using music teach reading',
 'exploring earth seismicity',
 'rise',
 'fairy tales folktales falling apart',
 'ambitious science teaching alaskan way viaduct collapse',
 'unleashing potential',
 'get gullah us',
 'texts',
 'language reading intervention',
 'bridging gap',
 'hear',
 'election fall 2016 materials',
 'full tummies full hearts full minds',
 'help young learners access technology',
 'give possibilities read favorite books',
 'notebooks young writers',
 'keep everything weighing',
 'let use math understand world',
 'loving literacy',
 'ears',
 'tetherball courts health exercise',
 'carnival indoor recess fun',
 'building bots',
 'shredding oldies',
 'taking closer look modeling independent learning',
 'getting fit ozo pedometers',
 'full stem ahead',
 'move music',
 'put listening ears',
 'mom dad see work portfolio',
 '21st century technology 21st century learners',
 'science art together no way',
 'classroom supplies',
 'back basics school supplies classroom',
 'empower young minds flexible seating classroom',
 'loud proud',
 'picking steam kindergarten',
 'time saved learning maximized',
 'graphic novels rescue',
 'fun games making academic gaines',
 'best seat class',
 'chromebook robotics stem part 2',
 'print world color',
 'classroom wish list year',
 'exploring enjoying life great book',
 'new year resolution become amazing readers',
 'meeting students fine motor sensory needs special education',
 'books reading levels',
 'engaging technology',
 'stem readers',
 'kindergarten stations full steam ahead',
 'starbuck goals',
 'books hand adventures school',
 'digital classroom library',
 'right red chair prek',
 'owl pellet',
 'ipads motivate engage students love reading',
 'microscopes engage elementary students scientific investigation',
 'students deserve best',
 'let make calender math possible',
 'sight mind',
 'extra extra read',
 'books nook',
 'tables fit needs little bodies',
 'soccer equipment',
 'bringing insects life 3d',
 'book read alouds catapulting students success',
 'tools build lifetime skills',
 'lockdown drills not annoyance',
 'wobble chairs keep moving',
 'closing gap apps',
 'reading using inference skills painting ocean friends',
 'library lacking literacy',
 'fun 3d doodle set',
 'reading rugs',
 'apple pi',
 'making music family affair',
 'make students tech savvy',
 'dear santa philadelphia 8th graders want books christmas',
 'magical math literature',
 'equal access',
 'hooked books',
 'moving target',
 'hear music pound let beef sound',
 'focused learning',
 'students rock',
 'perceiving patterns painting',
 'magazines make learning fun',
 'let play hockey',
 'innovation nation creating learning space student exploration',
 'tummies rumble empty',
 'flexible seating active seating active learners',
 'right track backpacks',
 'find colored square',
 'help 5th grade scientists learn technology',
 'dusting soul',
 'taking learning scholastic let find',
 'scholastic news',
 'learning science handson approach',
 'osmo ipad stem centers',
 'keep chrome books safe fully charged every day',
 'nothing end recess boredom get fit',
 'organizing guiding future readers',
 'oh baby parenting facs',
 'creativity crayola',
 'graphic novels library',
 'take value granted',
 'listen love learning headphones needed',
 'fidgeting students need fidgets',
 'ipad minis many learners',
 'macbook pro computer pros',
 'engage students flexible seating',
 'crazy ukulele',
 'wiggly bottoms need special seats',
 'got beat need drums',
 'life hurricane matthew',
 'making magical music',
 'steam stem growing together',
 'printing press 20',
 'clay glaze storage new kiln 050216',
 'curing autism mrs carter class',
 'reading table',
 'need reach virtual mentors',
 'walk',
 'let paint',
 'carts computers',
 'great bridge project',
 'flexible comfortable seating',
 'let strings sing',
 'pottery club',
 'art teaching kids need zen art school',
 'stem kits maker space',
 'stem books animal reports',
 'creative sticky murals',
 'feed minds hungry students need snacks',
 'keep school garden alive thriving',
 'walking playing purpose',
 'class library lacking chapter books',
 'stand success',
 'wiggle work',
 'music books new musicians',
 'flexible seating focus',
 'math manipulatives eq3',
 'help us put supply shelf back together',
 'special education students need work station desks chairs',
 'comfy chairs help us become scholars',
 'happy day prek',
 'learn science lost wax jewelry',
 'green screen projectshelp wanted',
 'handson exploration problem solving stem',
 'binder finder',
 'googlify classroom',
 'abstract reality',
 'active bodies engaged minds',
 'beautiful copies',
 'learning listening new literacy center',
 'classroom manipulatives amazing second graders',
 'help us play adapted sports',
 'technology technology',
 'reaching reading goals',
 'ipad minis kindergarten minis',
 'technology art oh',
 'building print rich classroom',
 'listen work',
 'math center activities',
 'ca not required reading without required book',
 'weaving history',
 'got wiggles',
 'bamboo pads differentiated learning',
 'want fitbits share please',
 'physical education move',
 'carpet heart classroom community',
 'organized manipulatives motivated mathmeticians',
 'art collaborative working',
 'help teach',
 'kill watt energy',
 'hot dots learning',
 'math tools create success',
 'read together learn together',
 'touch lives touchtronic technology',
 'sturdy shelving',
 'addition way life',
 'let calm read',
 'burlington backpacks win',
 'wiggle work flexible seating options',
 'teaching pitch critical period auditory development',
 'science technology mathyes please',
 'kindergarten makeover',
 'balance discs allow brain readiness learn',
 'ipad myclass',
 'rockin school chairs students autism spectrum',
 'creating digital learners',
 'mrs newsome',
 'desktop computers support inclusion special education students',
 'mathterpiece',
 'schoolwide mindfulness',
 'focus movement',
 'kinesthetic kinders like move move',
 'plant seed read',
 'backpacks class',
 'technology today transcendence tomorrow',
 'supplies needed growing minds',
 'flexible seating project',
 'help room got flooded',
 'creative comfortable stem projects',
 'extra extra read reading kindergarten',
 'highlight',
 'look grow',
 'building student knowledge geometric shape building sets',
 'organized classroom happy classroom',
 'help immerse art class watercolors',
 'multiple mallet mania',
 'robotics 3d printing urban makerspace classroom',
 'family engagement stem',
 'media center makeover bringing school library inviting students',
 'make learning permanent',
 'standup swing success',
 'tiles not comfy',
 'vivid visuals math reading',
 'full steam ahead',
 'teaching triumphantly tablets',
 'raved readers',
 'middle school supplies smiles',
 'variety spice literature',
 'book boxes clipboards mrs chen',
 'discovering phantom language phantom tollbooth',
 'extra extrastorage',
 'scientific calculators science',
 'charging chrome',
 'getting comfy cozy reading rug',
 'harnessing wiggles hokki stools',
 'like move move',
 'steming ahead folktales',
 'act books',
 'chromebooks classroom',
 'crazy coding',
 'controlling robotsone code time',
 'starbucks classroom',
 'empowering students art creativity comes alive',
 'check playosmocom',
 'food soul',
 'miss luce classroom mailbox',
 'never young healthy',
 'living color',
 'keep music alive',
 'chromebooks classroom',
 'hear music see music',
 'first grade full steam ahead',
 'chromebooks third grade class',
 'living color digital',
 'hear',
 '12books you34we thank even',
 '3d printer young designers innovators',
 'show money matters',
 'want learn english',
 'reaching new goals fitness mindfulness',
 'science much fun',
 'handson mindson',
 'technology please',
 'read',
 'painting supplies talented 4th graders',
 'movin groovin workin part 2',
 'healthier happier students',
 'watch techwatch learn learn',
 'supplies starting second grade',
 'let get rid desks',
 'backpacks organized scholars',
 '1st graders move groove technology',
 'literacy centers 20',
 'creative critical thinking technology literacy chromebooks',
 'ipads wanted cooperative learning environment',
 'puppets performance',
 'goldilocks trespasses understanding plot adaptation examinations',
 'folder frenzy',
 'tidy area better area learn',
 'studentled conferences',
 'share learning love',
 'chromebooks curious minds',
 'relaxing reading nook',
 'technology research',
 'materials learning centers sound like winner',
 'life cycles unit hatching chicks',
 'rhyme repeat learn read',
 'organization collaborative space',
 'dear diary help students express',
 'project read part 2',
 'chromebook math',
 'bridging technology gap',
 'technology kindergarten',
 'crazy kindles',
 'keeping teeth clean stomachs full',
 'cubbies please',
 'piano project producing proud performers',
 'digital magazine',
 'starting year right foot',
 'leaders techchology',
 'reading classics today',
 'digitalize classroom',
 'helping students become upfront learners',
 'let connect steam',
 'identity selfportrait',
 'apple',
 'touch feelshapes learning lives',
 '21st century skills technology optimized improve world',
 'alamo supplemental reading',
 'future health medicine',
 'empowering students art moving full steam ahead',
 'learning photography early age',
 'happiness seeing hearing students read',
 '4th graders need understand importance enviromental science',
 'scientist need journals',
 'books ahoy',
 'blue seat sacks engaging books esol classroom',
 'tools success',
 'wiggle wiggle wiggleand learn',
 'seeking knowledge technology',
 'let hit target active classroom',
 'reads around world',
 'learning technology',
 'never underestimate importance enough room work',
 'extra extra third graders read',
 'osmo save day',
 'math mania learn math better path',
 'future mathematicians scientists',
 'reading chairs',
 'weighty word wizards',
 'flexible seating classroom flexible minds control',
 'like move move',
 'making makerspacepart two',
 'extra extra read social justice readers',
 'classroom supplies needed',
 'eyes doc cam',
 'listen learn',
 'yoga exercise',
 'help us hear tasks',
 'basic needs keep 3rd graders healthy organized',
 'technology tubergen tigers',
 '',
 'white boards supplies students special needs',
 'calligraphy no agenda',
 'fly us moon astronomy lab supplies',
 'would nice see',
 'survival resilience redemption',
 'stem inspiration literature',
 'come along listen lullaby east la',
 'listening center extraordinaire',
 'bees flowers planets yippee',
 'ipads titus talented team',
 'initiate ipads',
 '3d printing innovation lab',
 'listening working wiping away workshop',
 'technology reading please',
 'recess relief',
 'kidney table small group instruction',
 'bouncing walls first grade',
 'leap learn',
 'stand deliver',
 'supplies school year',
 'student instruments',
 'magnificent math',
 'showingscientific minds',
 'balancing acts',
 'story acting ells',
 'flexible seating',
 'comfy cozy reading bags',
 'learning overcome sensory deficits different textures',
 'handson science tiny hands',
 'ipad accessories multiage',
 'bring learning life',
 'organize supplies please',
 'slap shot sports',
 'engaging bilingual learners maximizing classroom space',
 'flexible seating flexible brains',
 'technology today learners',
 'books build brilliant brains',
 'finding truth fiction',
 'flexible seating working wonders 2nd graders',
 'technology future',
 'chromebooks enhance learning',
 'third graders protecting environment',
 'complete core complete kids',
 'early chapter books',
 'kinders class needs safe place technology part 2',
 'please help students fulfill need speed',
 'last lecture middle school mantras',
 'fostering love literature',
 'making reading exciting technology',
 'flexible seating first graders',
 'seamlessly integrating technology esol curriculum',
 'start right art',
 'book tastings book clubs',
 'essential snack hungry learners',
 'plop read',
 'jazz',
 'coding fun part 1',
 'listening books helps us learn understand',
 'magazines assist fluency comprehension',
 'readers live thousand lives turing 5th graders bookworms',
 'picture books pop',
 'place learn grow',
 'learning better reader ipads',
 'teaching social justice read alouds',
 'increasing engagement technology',
 'need bullfrogs dissect please arcfsims',
 'reading essentials',
 'equitable access collaborate communicate chromebooks',
 'centers needed prek',
 '21st century learners need chromebooks',
 '1 2 3 eyes',
 'lady lancers basketball',
 'everyday counts especially math',
 'expanding learning',
 'ipads library media center part ii',
 'touchscreen tablets computer science mathematics',
 'shine light biology',
 'mars 2030',
 'reading fun',
 'seating success super heroes',
 'healthy bodies healthy minds',
 'engineering kindergarten',
 'chromebooks 21st century classroom',
 'virtual field trips kg kids',
 'creating lifelong readers learners thinkers',
 'technology stars limit',
 'staying indoor active gonoodle',
 'lego work',
 'project leopard cub coding club part iv',
 'wired sound',
 'fidget cubes fidgety',
 'economics market market learn economy',
 'handson math redesign',
 'technology sets us free',
 'technology prek',
 'ipad literacy math stations',
 'seat one seat',
 'creative coding',
 'project high',
 'today readers tomorrow leaders',
 'bee aware environment',
 'beautiful project',
 'new home growing turbo',
 'kindle excitement',
 'want omnikin ball',
 'shaping new year',
 'flexible seating',
 'no weighting fitness',
 'start something great',
 'excited active learning',
 'ilearn ipads',
 'hands learning technology',
 'gopro cameras going green environmental filmmaking',
 'growing garden',
 'kindergarten learners ipads',
 'new look new year',
 'wiggle n read',
 'super students need super supplies success second grade',
 'focus pocus',
 ...]
```

1. 4 Preparing data for models

In [245]:

```
project_data.columns
```

Out[245]:

```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_s
tate',
       'project_submitted_datetime', 'project_grade_category', 'proj
ect_title',
       'project_essay_1', 'project_essay_2', 'project_essay_3',
       'project_essay_4', 'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_a
pproved',
       'clean_categories', 'clean_subcategories', 'essay', 'price',
       'quantity'],
      dtype='object')
```

we are going to consider

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

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

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

1.4.1 Vectorizing Categorical data

- https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/ (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/)

In [246]:

```
# we use count vectorizer to convert the values into one hot encoded features
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()), lowercase=
False, binary=True)
vectorizer.fit(project_data['clean_categories'].values)
print(vectorizer.get_feature_names())


categories_one_hot = vectorizer.transform(project_data['clean_categories'].value
s)
print("Shape of matrix after one hot encodig ",categories_one_hot.shape)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLe
arning', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_
Language']
Shape of matrix after one hot encodig  (109248, 9)
```

In [247]:

```
# we use count vectorizer to convert the values into one hot encoded features
vectorizer = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys()), lowerc
ase=False, binary=True)
vectorizer.fit(project_data['clean_subcategories'].values)
print(vectorizer.get_feature_names())


sub_categories_one_hot = vectorizer.transform(project_data['clean_subcategories'
].values)
print("Shape of matrix after one hot encodig ",sub_categories_one_hot.shape)
```

```
['Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolv
ement', 'Extracurricular', 'Civics_Government', 'ForeignLanguages',
'NutritionEducation', 'Warmth', 'Care_Hunger', 'SocialSciences', 'Pe
rformingArts', 'CharacterEducation', 'TeamSports', 'Other', 'College
_CareerPrep', 'Music', 'History_Geography', 'Health_LifeScience', 'E
arlyDevelopment', 'ESL', 'Gym_Fitness', 'EnvironmentalScience', 'Vis
ualArts', 'Health_Wellness', 'AppliedSciences', 'SpecialNeeds', 'Lit
erature_Writing', 'Mathematics', 'Literacy']
Shape of matrix after one hot encodig  (109248, 30)
```

In [248]:

```
#  Please do the similar feature encoding with state, teacher_prefix and project
_grade_category also
```

In [249]:

```
# one hot encoding pandas https://stackoverflow.com/a/39287161/2515354
school_state_one_hot = pd.get_dummies(project_data.school_state)
print(school_state_one_hot.head())
print("Shape of school state - hot encoded frame: ", school_state_one_hot.shape)
```

```
   AK  AL  AR  AZ  CA  CO  CT  DC  DE  FL ...  SD  TN  TX  UT  VA  V
T  WA  WI  \
0   0   0   0   0   0   0   0   0   0   0 ...   0   0   0   0   0
0   0   0
1   0   0   0   0   0   0   0   0   0   1 ...   0   0   0   0   0
0   0   0
2   0   0   0   1   0   0   0   0   0   0 ...   0   0   0   0   0
0   0   0
3   0   0   0   0   0   0   0   0   0   0 ...   0   0   0   0   0
0   0   0
4   0   0   0   0   0   0   0   0   0   0 ...   0   0   1   0   0
0   0   0

   WV  WY
0   0   0
1   0   0
2   0   0
3   0   0
4   0   0

[5 rows x 51 columns]
Shape of school state - hot encoded frame:  (109248, 51)
```

In [250]:

```
# one hot encoding pandas https://stackoverflow.com/a/39287161/2515354
teacher_prefix_one_hot = pd.get_dummies(project_data.teacher_prefix)
print(teacher_prefix_one_hot.head())
print("Shape of teacher prefix - hot encoded frame: ", teacher_prefix_one_hot.sh
ape)
```

```
   Dr.  Mr.  Mrs.  Ms.  Teacher
0    0    0     1    0        0
1    0    1     0    0        0
2    0    0     0    1        0
3    0    0     1    0        0
4    0    0     1    0        0
Shape of teacher prefix - hot encoded frame:  (109248, 5)
```

In [251]:

```
# one hot encoding pandas https://stackoverflow.com/a/39287161/2515354
project_grade_category_one_hot = pd.get_dummies(project_data.project_grade_categ
ory)
print(project_grade_category_one_hot.head())
print("Shape of project grade category - hot encoded frame: ", project_grade_cat
egory_one_hot.shape)
```

```
   Grades 3-5  Grades 6-8  Grades 9-12  Grades PreK-2
0           0           0            0              1
1           0           1            0              0
2           0           1            0              0
3           0           0            0              1
4           0           0            0              1
Shape of project grade category - hot encoded frame:  (109248, 4)
```

1.4.2 Vectorizing Text data

1.4.2.1 Bag of words

In [252]:

```
# We are considering only the words which appeared in at least 10 documents(rows
 or projects).
vectorizer = CountVectorizer(min_df=10)
text_bow = vectorizer.fit_transform(preprocessed_essays)
print("Shape of matrix after one hot encodig ",text_bow.shape)
```

```
Shape of matrix after one hot encodig  (109248, 16623)
```

1.4.2.2 Bag of Words on `project_title`

In [253]:

```
# you can vectorize the title also
# before you vectorize the title make sure you preprocess it
# Similarly you can vectorize for title also

vectorizer = CountVectorizer(min_df=10)
cleaned_titles_bow = vectorizer.fit_transform(cleaned_titles)
print("Shape of cleaned titles dataframe after bag of words", cleaned_titles_bow
.shape)
```

```
Shape of cleaned titles dataframe after bag of words (109248, 3194)
```

1.4.2.3 TFIDF vectorizer

In [254]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=10)
text_tfidf = vectorizer.fit_transform(preprocessed_essays)
print("Shape of matrix after one hot encodig ",text_tfidf.shape)
```

```
Shape of matrix after one hot encodig  (109248, 16623)
```

1.4.2.4 TFIDF Vectorizer on `project_title`

In [255]:

```
# Similarly you can vectorize for title also
vectorizer = TfidfVectorizer(min_df=10)
cleaned_titles_tfidf = vectorizer.fit_transform(cleaned_titles)
print("Shape of cleaned titles dataframe after TFIDF ", cleaned_titles_tfidf.sha
pe)
```

```
Shape of cleaned titles dataframe after TFIDF  (109248, 3194)
```

1.4.2.5 Using Pretrained Models: Avg W2V

In [256]:

```
'''
# Reading glove vectors in python: https://stackoverflow.com/a/38230349/4084039
def loadGloveModel(gloveFile):
    print ("Loading Glove Model")
    f = open(gloveFile,'r', encoding="utf8")
    model = {}
    for line in tqdm(f):
        splitLine = line.split()
        word = splitLine[0]
        embedding = np.array([float(val) for val in splitLine[1:]])
        model[word] = embedding
    print ("Done.",len(model)," words loaded!")
    return model
model = loadGloveModel('glove.42B.300d.txt')

# ============================
Output:

Loading Glove Model
1917495it [06:32, 4879.69it/s]
Done. 1917495  words loaded!

# ============================

words = []
for i in preproced_texts:
    words.extend(i.split(' '))

for i in preproced_titles:
    words.extend(i.split(' '))
print("all the words in the coupus", len(words))
words = set(words)
print("the unique words in the coupus", len(words))

inter_words = set(model.keys()).intersection(words)
print("The number of words that are present in both glove vectors and our coupu
s", \
      len(inter_words),"(",np.round(len(inter_words)/len(words)*100,3),"%)")

words_courpus = {}
words_glove = set(model.keys())
for i in words:
    if i in words_glove:
        words_courpus[i] = model[i]
print("word 2 vec length", len(words_courpus))


# stronging variables into pickle files python: http://www.jessicayung.com/how-t
o-use-pickle-to-save-and-load-variables-in-python/

import pickle
with open('glove_vectors', 'wb') as f:
    pickle.dump(words_courpus, f)


'''
```

Out[256]:

```
'\n# Reading glove vectors in python: https://stackoverflow.com/a/38
230349/4084039\ndef loadGloveModel(gloveFile):\n    print ("Loading
Glove Model")\n    f = open(gloveFile,\'r\', encoding="utf8")\n    m
odel = {}\n    for line in tqdm(f):\n        splitLine = line.split
()\n        word = splitLine[0]\n        embedding = np.array([float
(val) for val in splitLine[1:]])\n        model[word] = embedding\n
   print ("Done.",len(model)," words loaded!")\n    return model\nmo
del = loadGloveModel(\'glove.42B.300d.txt\')\n\n# ==================
==========\nOutput:\n    \nLoading Glove Model\n1917495it [06:32, 48
79.69it/s]\nDone. 1917495  words loaded!\n\n# ======================
======\n\nwords = []\nfor i in preproced_texts:\n    words.extend(i.
split(\' \'))\n\nfor i in preproced_titles:\n    words.extend(i.spli
t(\' \'))\nprint("all the words in the coupus", len(words))\nwords =
set(words)\nprint("the unique words in the coupus", len(words))\n\ni
nter_words = set(model.keys()).intersection(words)\nprint("The numbe
r of words that are present in both glove vectors and our coupus",
     len(inter_words),"(",np.round(len(inter_words)/len(words)*100,
3),"%)")\n\nwords_courpus = {}\nwords_glove = set(model.keys())\nfor
i in words:\n    if i in words_glove:\n        words_courpus[i] = mo
del[i]\nprint("word 2 vec length", len(words_courpus))\n\n\n# strong
ing variables into pickle files python: http://www.jessicayung.com/h
ow-to-use-pickle-to-save-and-load-variables-in-python/\n\nimport pic
kle\nwith open(\'glove_vectors\', \'wb\') as f:\n    pickle.dump(wor
ds_courpus, f)\n\n\n'
```

In [257]:

```python
# stronging variables into pickle files python: http://www.jessicayung.com/how-t
o-use-pickle-to-save-and-load-variables-in-python/
# make sure you have the glove_vectors file
with open('glove_vectors', 'rb') as f:
    model = pickle.load(f)
    glove_words =  set(model.keys())
```

In [258]:

```
# average Word2Vec
# compute average word2vec for each review.
avg_w2v_vectors = []; # the avg-w2v for each sentence/review is stored in this l
ist
for sentence in tqdm(preprocessed_essays): # for each review/sentence
    vector = np.zeros(300) # 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[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors.append(vector)

print(len(avg_w2v_vectors))
print(len(avg_w2v_vectors[0]))
```

```
100%|██████████| 109248/109248 [00:30<00:00, 3630.56it/s]

109248
300
```

1.4.2.6 Using Pretrained Models: AVG W2V on `project_title`

In [263]:

```
# Similarly you can vectorize for title also
```

In [265]:

```
# average Word2Vec
# compute average word2vec for each clened project title.
# Referred earlier code
project_title_avg_w2v_vectors = []; # the avg-w2v for each project title
for sentence in tqdm(cleaned_titles): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words = 0; # num of words with a valid vector in the project title
    for word in sentence.split(): # for each word in a project title
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    project_title_avg_w2v_vectors.append(vector)

print(len(project_title_avg_w2v_vectors))
print(len(project_title_avg_w2v_vectors[0]))
```

```
100%|██████████| 109248/109248 [00:01<00:00, 80217.31it/s]

109248
300
```

1.4.2.7 Using Pretrained Models: TFIDF weighted W2V

In [267]:

```
# S = ["abc def pqr", "def def def abc", "pqr pqr def"]
tfidf_model = TfidfVectorizer()
tfidf_model.fit(preprocessed_essays)
# we are converting a dictionary with word as a key, and the idf as a value
dictionary = dict(zip(tfidf_model.get_feature_names(), list(tfidf_model.idf_)))
tfidf_words = set(tfidf_model.get_feature_names())
```

In [268]:

```
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors = []; # the avg-w2v for each sentence/review is stored in this
 list
for sentence in tqdm(preprocessed_essays): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =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) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf val
ue((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split
())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors.append(vector)

print(len(tfidf_w2v_vectors))
print(len(tfidf_w2v_vectors[0]))
```

```
100%|██████████| 109248/109248 [03:03<00:00, 596.60it/s]

109248
300
```

1.4.2.9 Using Pretrained Models: TFIDF weighted W2V on `project_title`

In [274]:

```
# Similarly you can vectorize for title also
```

In [270]:

```
# average Word2Vec
# compute average word2vec for each review.
# Referred earlier code

project_title_tfidf_w2v_vectors = []; # the avg-w2v for each project title
for sentence in tqdm(cleaned_titles): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the project title
    for word in sentence.split(): # for each word in a project title
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf val
ue((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split
())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    project_title_tfidf_w2v_vectors.append(vector)

print(len(project_title_tfidf_w2v_vectors))
print(len(project_title_tfidf_w2v_vectors[0]))
```

```
100%|██████████| 109248/109248 [00:02<00:00, 37895.05it/s]

109248
300
```

1.4.3 Vectorizing Numerical features

In [271]:

```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-learn.org/stable/modules/generated/skl
earn.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_scal
ar.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 [272]:

```
price_standardized
```

Out[272]:

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

1.4.4 Merging all the above features

- we need to merge all the numerical vectors i.e catogorical, text, numerical vectors

In [273]:

```
print(categories_one_hot.shape)
print(sub_categories_one_hot.shape)
print(text_bow.shape)
print(price_standardized.shape)
```

```
(109248, 9)
(109248, 30)
(109248, 16623)
(109248, 1)
```

In [280]:

```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
# with the same hstack function we are concatinating a sparse matrix and a dense
 matirx :)
X = hstack((categories_one_hot, sub_categories_one_hot, text_bow, price_standard
ized))
X.shape
```

Out[280]:

```
(109248, 16663)
```

Assignment 2: Apply TSNE

If you are using any code snippet from the internet, you have to provide the reference/citations, as we did in the above cells. Otherwise, it will be treated as plagiarism without citations.

1. In the above cells we have plotted and analyzed many features. Please observe the plots and write the observations in markdown cells below every plot.
2. EDA: Please complete the analysis of the feature: teacher_number_of_previously_posted_projects
3. Build the data matrix using these features
   - school_state : categorical data (one hot encoding)
   - clean_categories : categorical data (one hot encoding)
   - clean_subcategories : categorical data (one hot encoding)
   - teacher_prefix : categorical data (one hot encoding)
   - project_title : text data (BOW, TFIDF, AVG W2V, TFIDF W2V)
   - price : numerical
   - teacher_number_of_previously_posted_projects : numerical
4. Now, plot FOUR t-SNE plots with each of these feature sets.
   A. categorical, numerical features + project_title(BOW)
   B. categorical, numerical features + project_title(TFIDF)
   C. categorical, numerical features + project_title(AVG W2V)
   D. categorical, numerical features + project_title(TFIDF W2V)
5. Concatenate all the features and Apply TNSE on the final data matrix
6. Note 1: The TSNE accepts only dense matrices
7. Note 2: Consider only 5k to 6k data points to avoid memory issues. If you run into memory error issues, reduce the number of data points but clearly state the number of datat-poins you are using

In [312]:

```
# Normalizing teacher_number_of_previously_posted_projects
# Referred earlier code

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

# Now standardize the data with above mean and variance.
teacher_previous_projects_standardized = teacher_previous_projects_scalar.transf
orm(project_data['teacher_number_of_previously_posted_projects'].values.reshape(
-1, 1))
```

```
Mean : 11.153165275336848, Standard deviation : 27.77702641477403
```

TSNE function

In [395]:

```
# converting sparse matrix to dense matrix  https://stackoverflow.com/a/1650576
6/2515354
# list to tuple  https://stackoverflow.com/a/12836173/2515354
# https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
# Referred AppliedAI video for code sample as well

# Building data matrix using above features

from sklearn.manifold import TSNE
from scipy.sparse import csr_matrix

def build_tsne_data_model(extra_column, no_of_datapoints=5000, perplexity=30.0):
    """
        This function takes an extra_column, number of datapoints,
        & perplexity for TSNE, and returns the newly built data model alongwith
        labels
            school_state : categorical data (one hot encoding)
            clean_categories : categorical data (one hot encoding)
            clean_subcategories : categorical data (one hot encoding)
            teacher_prefix : categorical data (one hot encoding)
            price : numerical
            teacher_number_of_previously_posted_projects : numerical
    """
    if extra_column is None:
        data_matrix = hstack((school_state_one_hot, categories_one_hot, sub_cate
gories_one_hot,
            teacher_prefix_one_hot, price_standardized, teacher_previous_project
s_standardized,
            cleaned_titles_bow, cleaned_titles_tfidf, project_title_avg_w2v_vect
ors,
                               project_title_tfidf_w2v_vectors))
    else:
        data_matrix = hstack((school_state_one_hot, categories_one_hot, sub_cate
gories_one_hot,
            teacher_prefix_one_hot, price_standardized, teacher_previous_project
s_standardized, extra_column))

    print("Shape of data matrix after hstack: ", data_matrix.shape)

    # converting the data to dense matrix:
    dense_data_matrix = csr_matrix(data_matrix)

    # converting to dense numpy array
    dense_data_matrix = dense_data_matrix.toarray()

    # Reducing number of datapoints for faster computation
    print('Choosing ', no_of_datapoints, ' datapoints for faster computation')
    dense_data_matrix = dense_data_matrix[:no_of_datapoints, :]
    print('Shape of data matrix after reducing datapoints: ', dense_data_matrix.
shape)

    data_model = TSNE(n_components=2, perplexity=perplexity, random_state=0)
    tsne_data = data_model.fit_transform(dense_data_matrix)
    labels = project_data.project_is_approved[:no_of_datapoints]

    return tsne_data, labels
```

Scatter plot function

In [396]:

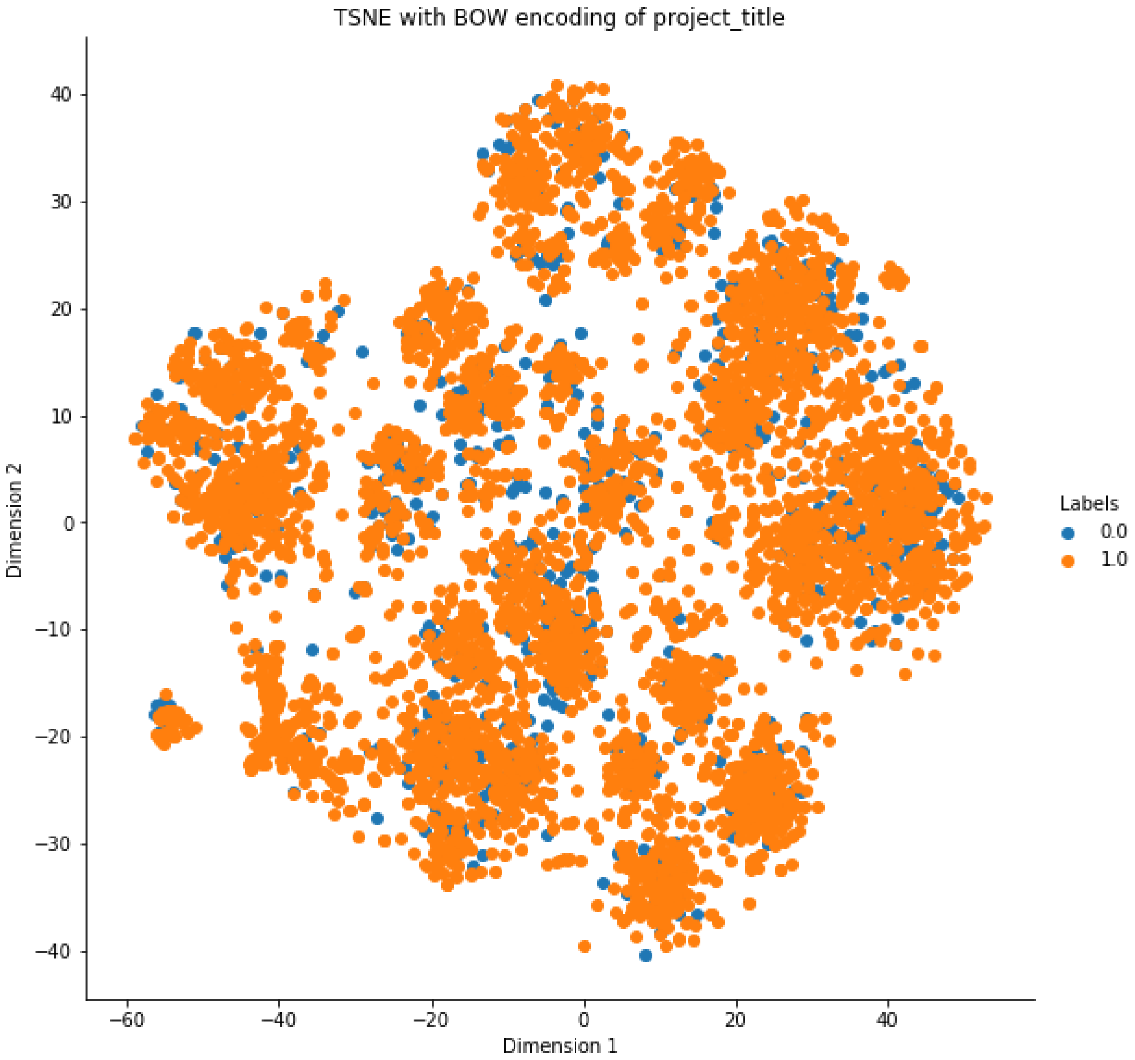
```
def plot_tsne(dataframe, title, x_axis_title, y_axis_title):
    sns.FacetGrid(dataframe, hue='Labels', size=8) \
       .map(plt.scatter, x_axis_title, y_axis_title).add_legend()
    plt.title(title)
    plt.show()
```

2.1 TSNE with `BOW` encoding of `project_title` feature

In [397]:

```
# Referred MNIST TSNE video for sample code
# please write all of the code with proper documentation and proper titles for each subsection
# 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

model, labels = build_tsne_data_model(cleaned_titles_bow, 5000, 50)

# Building a new data frame for visualization
tsne_data = np.vstack((model.T, labels)).T
tsne_dataframe = pd.DataFrame(data=tsne_data, columns=('Dimension 1', 'Dimension 2', 'Labels'))

# Plotting data
plot_tsne(tsne_dataframe, 'TSNE with BOW encoding of project_title', \
          'Dimension 1', 'Dimension 2')
```

```
Shape of data matrix after hstack:  (109248, 3291)
Choosing  5000  datapoints for faster computation
Shape of data matrix after reducing datapoints:  (5000, 3291)
```



TSNE with BOW encoding of project_title

Observations:

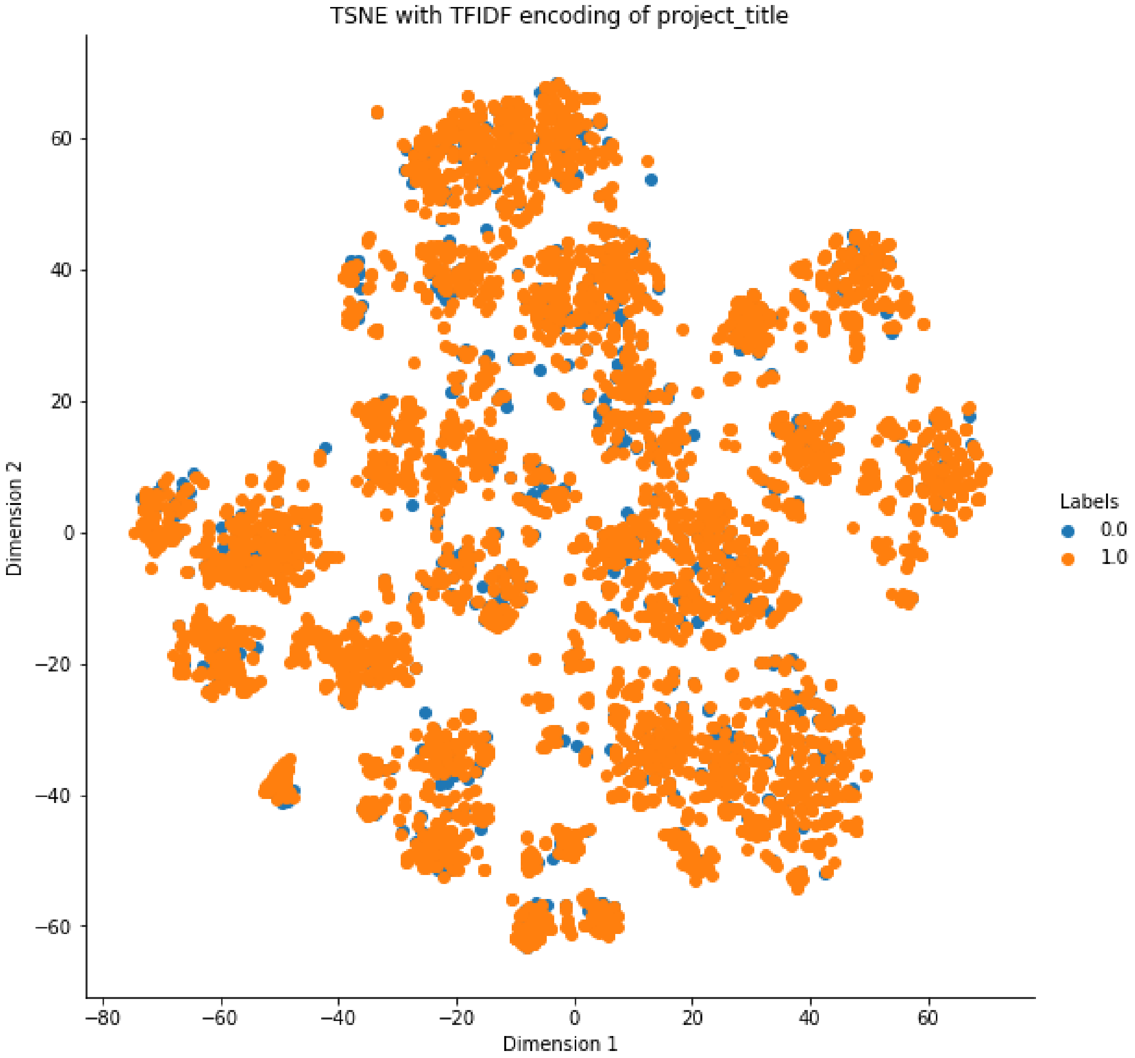
- The datapoints are overlapped, no useful information is present.
- Even after running this for 2 different perplexities (30, 50), we don't have any good result.

2.2 TSNE with `TFIDF` encoding of `project_title` feature

In [398]:

```
# please write all the code with proper documentation, and proper titles for each subsection
# 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

model, labels = build_tsne_data_model(cleaned_titles_tfidf, 5000, 50)

# Building a new data frame for visualization
tsne_data = np.vstack((model.T, labels)).T
tsne_dataframe = pd.DataFrame(data=tsne_data, columns=('Dimension 1', 'Dimension 2', 'Labels'))

# Plotting data
plot_tsne(tsne_dataframe, 'TSNE with TFIDF encoding of project_title', \
          'Dimension 1', 'Dimension 2')
```

```
Shape of data matrix after hstack:  (109248, 3291)
Choosing  5000  datapoints for faster computation
Shape of data matrix after reducing datapoints:  (5000, 3291)
```



Observations:

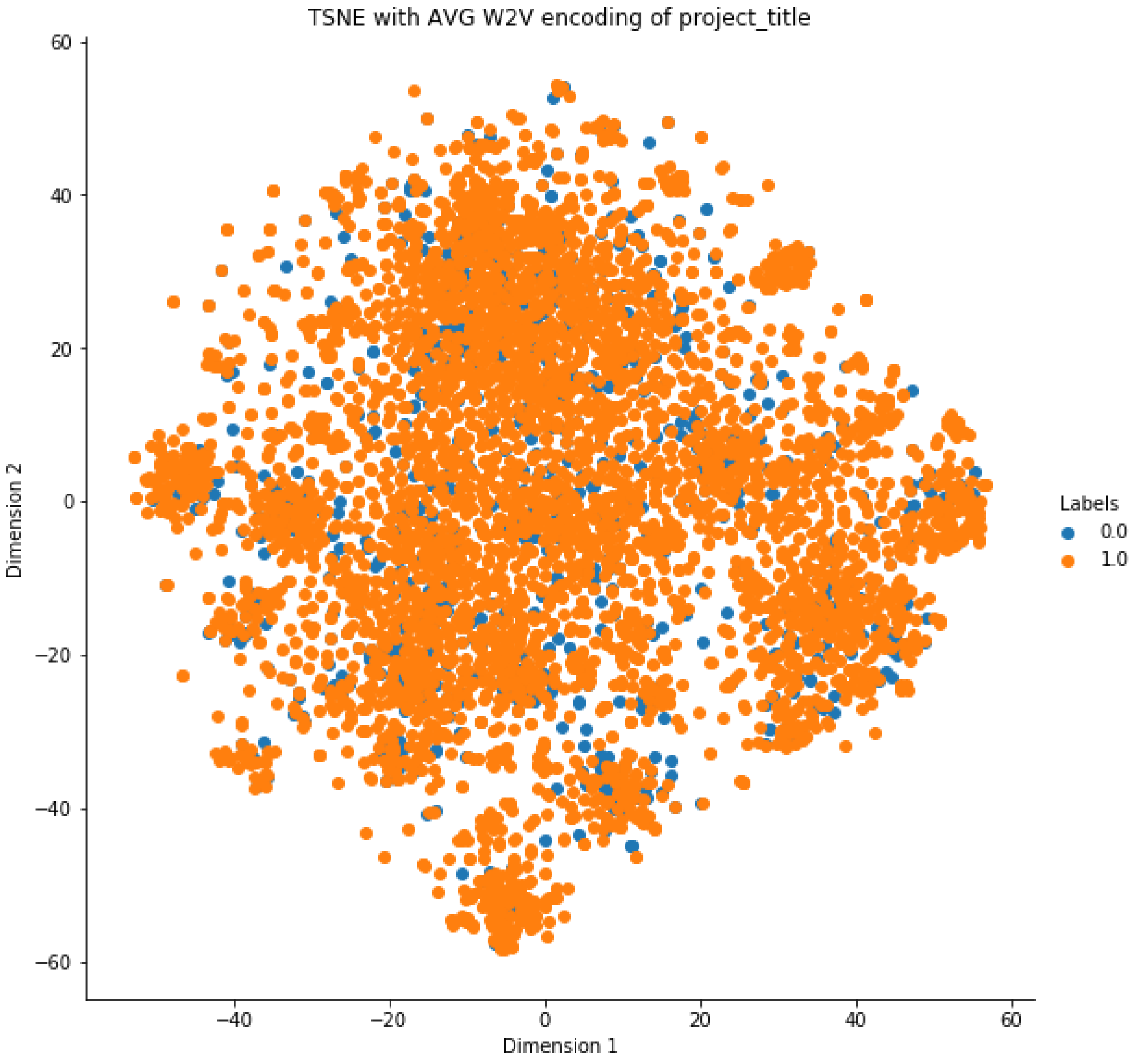
- There is a little bit better clustering for this plot. However the points are overlapping.
- project approval status(blue and orange) datapoints are not well separated, so no useful information here.

2.3 TSNE with `AVG W2V` encoding of `project_title` feature

In [399]:

```
# please write all the code with proper documentation, and proper titles for each subsection
# 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

model, labels = build_tsne_data_model(project_title_avg_w2v_vectors, 5000, 50)

# Building a new data frame for visualization
tsne_data = np.vstack((model.T, labels)).T
tsne_dataframe = pd.DataFrame(data=tsne_data, columns=('Dimension 1', 'Dimension 2', 'Labels'))

# Plotting data
plot_tsne(tsne_dataframe, 'TSNE with AVG W2V encoding of project_title', \
          'Dimension 1', 'Dimension 2')
```

```
Shape of data matrix after hstack:  (109248, 397)
Choosing  5000  datapoints for faster computation
Shape of data matrix after reducing datapoints:  (5000, 397)
```



Observations:

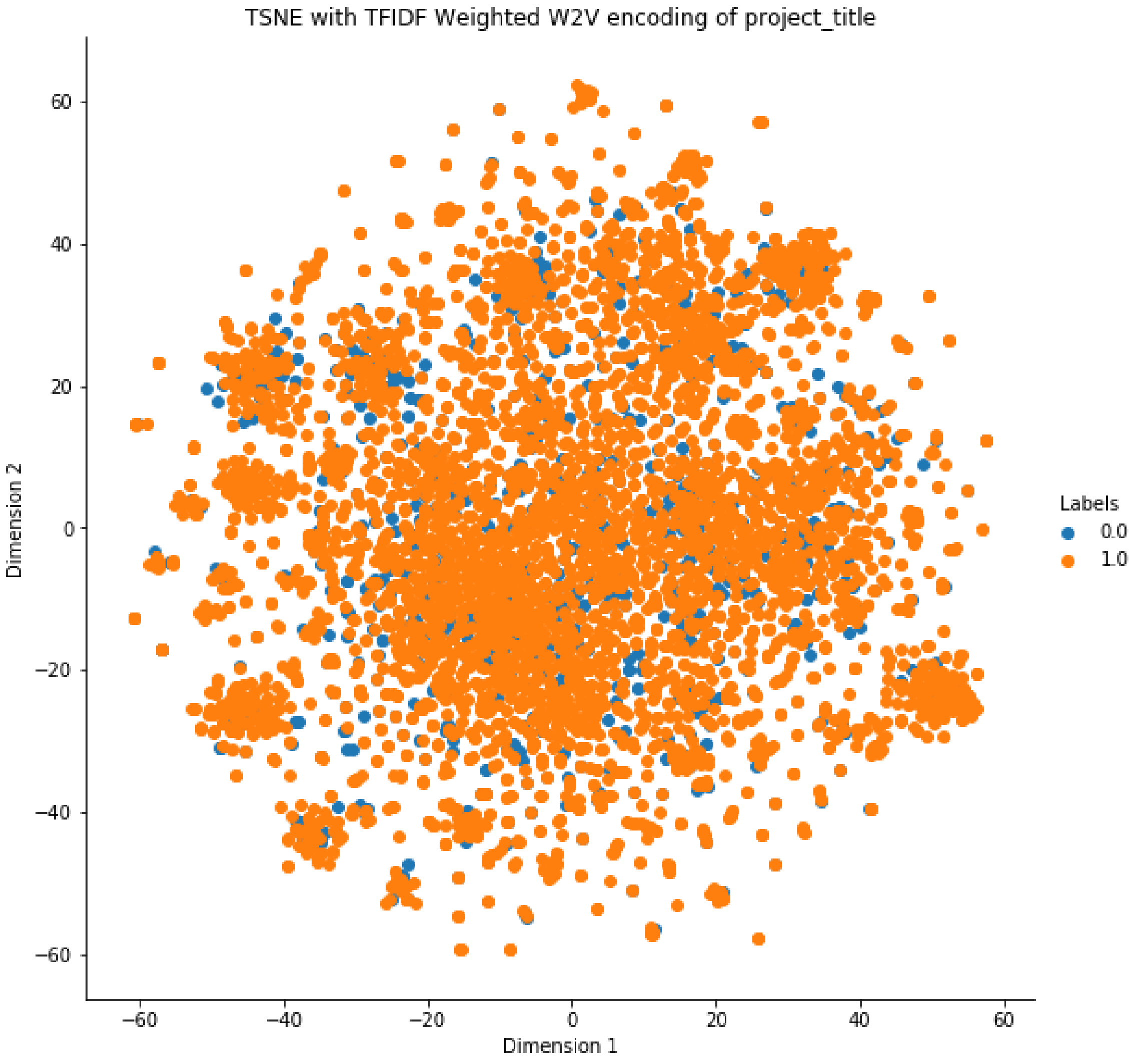
- We're able to see some of the blue datapoints, but still there's a lot of overlap.
- Also there are no separate clusters, so no useful insights here.

2.4 TSNE with `TFIDF Weighted W2V` encoding of `project_title` feature

In [400]:

```
# please write all the code with proper documentation, and proper titles for each subsection
# 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

model, labels = build_tsne_data_model(project_title_tfidf_w2v_vectors, 5000, 50)

# Building a new data frame for visualization
tsne_data = np.vstack((model.T, labels)).T
tsne_dataframe = pd.DataFrame(data=tsne_data, columns=('Dimension 1', 'Dimension 2', 'Labels'))

# Plotting data
plot_tsne(tsne_dataframe, 'TSNE with TFIDF Weighted W2V encoding of project_title', \
          'Dimension 1', 'Dimension 2')
```

```
Shape of data matrix after hstack:  (109248, 397)
Choosing  5000  datapoints for faster computation
Shape of data matrix after reducing datapoints:  (5000, 397)
```



Observations:

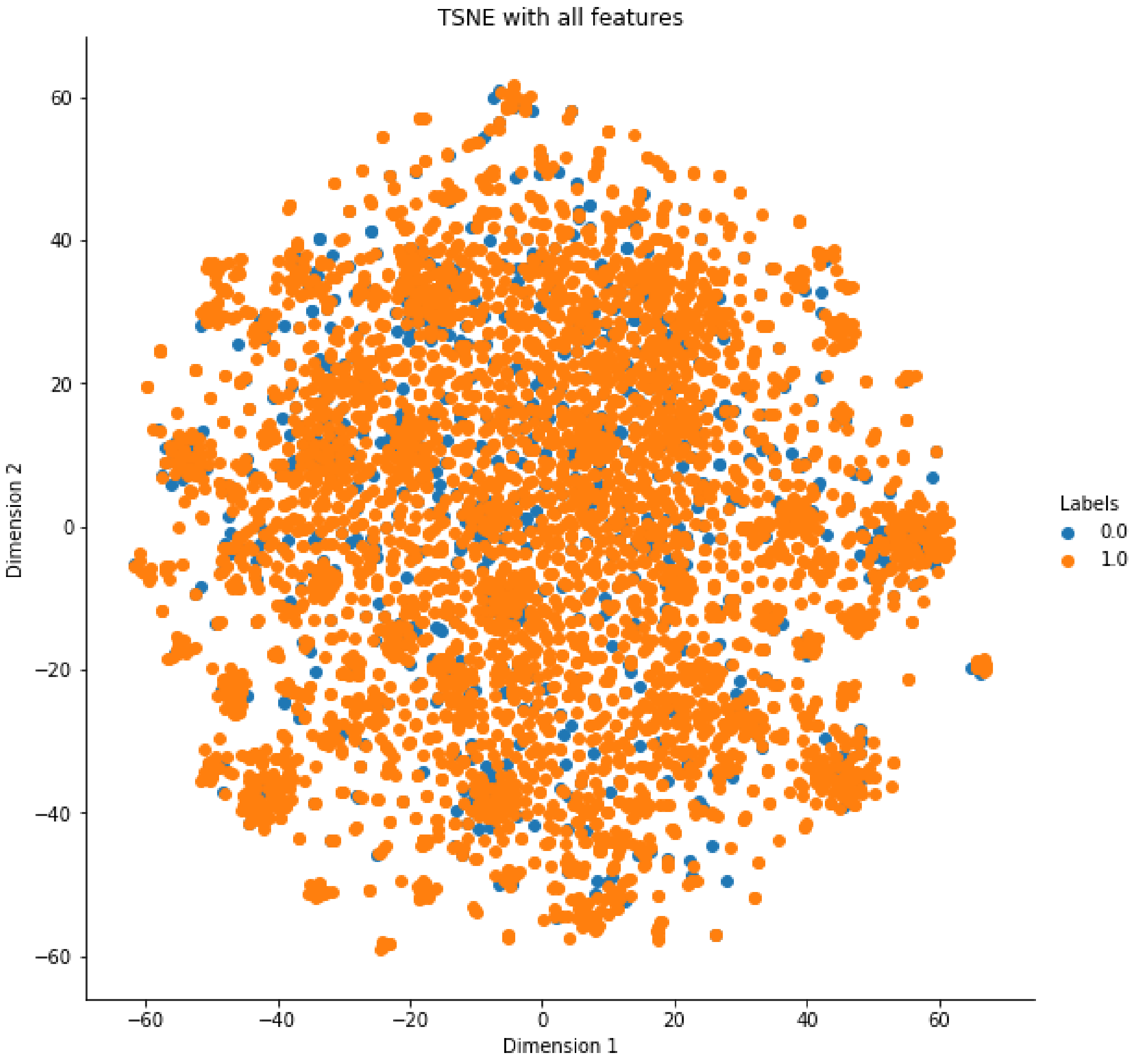
- Points are overlapped, and there's no well separation between the datapoints.

TSNE for the data matrix with all features

In [401]:

```
# TSNE for the combined data matrix with all the features

model, labels = build_tsne_data_model(None, 5000, 50)

# Building a new data frame for visualization
tsne_data = np.vstack((model.T, labels)).T
tsne_dataframe = pd.DataFrame(data=tsne_data, columns=('Dimension 1', 'Dimension
 2', 'Labels'))

# Plotting data
plot_tsne(tsne_dataframe, 'TSNE with all features', 'Dimension 1', 'Dimension 2'
)
```

```
Shape of data matrix after hstack:  (109248, 7085)
Choosing  5000  datapoints for faster computation
Shape of data matrix after reducing datapoints:  (5000, 7085)
```



Observations

- Although we can see some micro clustering between orange data points, it is not useful
- Points are overlapping, and there's no clear separation between the blue and orange points

2.5 Summary

- 5000 datapoints were used to run TSNE.
- TSNE doesn't seem to give good results even after running it for different perplexities.
- Clusters are not well separated, and the datapoints overlap a lot in all of the above plots.

Conclusion

- California & Vermont have the highest (15388) & lowest (80) number of project submissions respectively.
- Most of the teachers who submit projects are female (50% +).
- Most of the projects are submitted for pre kindergarden students (40000+).
- Projects are usually submitted for younger students as compared to elder students.
- More than 50000 literacy language projects have been approved. So people are more likely to donate for literacy projects than warmth/hunger-care/applied sciences projects.
- Project title word count doesn't have any real effect on project approval.
- Digit count in project title doesn't have a significant effect on project approval.
- The teachers who had posted some projects earlier have higher chance of project approvals than the ones who didn't.
- More than 80% projects are likely to get approved for a teacher who has submitted 13+ projects earlier.
- TSNE doesn't give us a good insight in the data in our case.