Document Created Date:02-07-2019

Image source : https://www.google.com/search?q=donors+choose&sxsrf=ACYBGNRh56fqsl74THPcxfW714XbHwNYAg:1577208827755&source=lnms&tbm=isch (https://www.google.com/search?q=donors+choose&sxsrf=ACYBGNRh56fqsl74THPcxfW714XbHwNYAg:1577208827755&source=lnms&tbm=isch

Understanding the data and datasource - 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. **Example**
`project_title`	Title of the projec • `Art Will Make Y` • `First`
`project_grade_category`	Grade level of students for which the project is targeted. One of t enumera • `Grad` • `G` • `G` • `Gr`
`project_subject_categories`	One or more (comma-separated) subject categories for the pro following enumerated li • `Applied` • `Care` • `Health` • `History` • `Literacy &` • `Math` • `Music &` • `Spec` •
	• `Music &` • `Literacy & Language, Math`
`school_state`	State where school is located (Two-letter U.S. (https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations#Pos **Ex**
`project_subject_subcategories`	One or more (comma-separated) subject subcategories fo • • `Literature & Writing, Social`
`project_resource_summary`	An explanation of the resources needed for the projec • `My students need hands on literacy materials f` `sensory nee`
`project_essay_1`	First applic
`project_essay_2`	Second applic
`project_essay_3`	Third applic
`project_essay_4`	Fourth applic
`project_submitted_datetime`	Datetime when project application was submitted. **Example:** `26` `12:`
`teacher_id`	A unique identifier for the teacher of the proposed proje `bdf8baa8fedef6bfeec7ae4`

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

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

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

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

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

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

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

Notes on the Essay Data

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

- **project_essay_1:** "Introduce us to your classroom"
- **project_essay_2:** "Tell us more about your students"
- **project_essay_3:** "Describe how your students will use the materials you're requesting"
- **project_essay_3:** "Close by sharing why your project will make a difference"

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

- **project_essay_1:** "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
- **project_essay_2:** "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

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

K-Means:AgglomerativeClustering:DBSCAN

Step by Step Procedure

- Understanding the Businessreal world problem
- Loading the data
- Preprocessing the data(based on the type of data = categorical , text, Numarical)
- Preprocessing data includes (removing outliers, impute missung values, cleaning data, remove spacial character, etc..)
- Vectorization data (one hot encoding)
- Vectorizing text data(bow, tfidf, avgw2v, tfidf weighted w2v)
- vectorizing numarical - Normalizer
- Merging all the above features
- Choosing the best data matrix on which you got the best AUC
- Dimensionality Reduction on the selected features
- Apply Kmeans - finding the best k using elbow-knee method
- Ploting wordcloud with essay for each cluster - KMeans
- Apply AgglomerativeClustering - selecet best k
- Apply AgglomerativeClustering for K = 2
- Ploting wordcloud with essay for cluster1 and cluster2 - AgglomerativeClustering
- Apply AgglomerativeClustering for K = 5
- Ploting wordcloud with essay for cluster1 and cluster2 - AgglomerativeClustering
- Finding the best eps using elbow-knee method
- Ploting wordcloud with essay for clusters - DBSCAN
- Ploting wordcloud with essay for cluster1 and noisecluster1 - DBSCAN
- Observation on overall model performences (Conclusion)
- Ploting the performences by tableu format.

```
C:\Users\Ramesh Battu> import required libraries
```

In [1]:

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

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

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

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

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

from tqdm import tqdm
import os
```

1.1 Reading Data

In [2]:

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

In [3]:

```
print("Number of data points in train data", project_data.shape)
print('-'*88)
print("The attributes of data :", project_data.columns.values)
print('-'*88)
```

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

In [4]:

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

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

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

project_data.head(2)
```

Out[4]:

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

In [5]:

```
print("Number of data points in train data", resource_data.shape)
print(resource_data.columns.values)
print('-'*60)
resource_data.tail(2)
```

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

Out[5]:

	id	description	quantity	price
1541270	p031981	Flormoon DC Motor Mini Electric Motor 0.5-3V 1...	2	8.14
1541271	p031981	WAYLLSHINE 6PCS 2 x 1.5V AAA Battery Spring Cl...	2	7.39

1.1.1 preprocessing of project_subject_categories

In [6]:

```
# remove special characters from list of strings python: https://stackoverflow.com/a/47
# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth
catogories = list(project_data['project_subject_categories'].values)

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
        if 'The' in j.split(): # this will split each of the catogory based on space "M
            j=j.replace('The','') # if we have the words "The" we are going to replace
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"M
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spa
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

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

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

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

1.1.2 preprocessing of project_subject_subcategories

In [7]:

```
sub_catogories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth

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
        if 'The' in j.split(): # this will split each of the catogory based on space "M
            j=j.replace('The','') # if we have the words "The" we are going to replace
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"M
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the trailing spa
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())

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

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

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

1.1.3 preprocessing of `school_state`

In [8]:

```
# remove special characters from list of strings python: https://stackoverflow.com/a/47
# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth
school_state_catogories = list(project_data['school_state'].values)
cat_list = []
for i in school_state_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
        if 'The' in j.split(): # this will split each of the catogory based on space "M
            j=j.replace('The','') # if we have the words "The" we are going to replace
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"M
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spa
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())
project_data['school_state'] = cat_list

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

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

1.1.4 Preprocessing of `teacher_prefix`

In [9]:

```
# citation code :https://www.datacamp.com/community/tutorials/categorical-data
project_data = project_data.fillna(project_data['teacher_prefix'].value_counts().index[0
teacher_prefix_catogories = list(project_data['teacher_prefix'].values)
# Citation code : https://stackoverflow.com/questions/39303912/tfidfvectorizer-in-scikit
# To convert the data type object to unicode string : used """astype('U')""" code from
# count of all the words in corpus python: https://stackoverflow.com/a/22898595/4084039
# remove special characters from list of strings python: https://stackoverflow.com/a/47
# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth
cat_list = []
for i in teacher_prefix_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
        if 'The' in j.split(): # this will split each of the catogory based on space "M
            j=j.replace('The','') # if we have the words "The" we are going to replace
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"M
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spa
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

project_data['teacher_prefix'] = cat_list

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

# dict sort by value python: https://stackoverflow.com/a/613218/4084039
teacher_prefix_dict = dict(my_counter)
sorted_teacher_prefix_dict = dict(sorted(teacher_prefix_dict.items(), key=lambda kv: kv
```

1.1.5 Preprocessing of project_grade_category

In [10]:

```
project_grade_catogories = list(project_data['project_grade_category'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-stri
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-pyth
cat_list = []
for i in project_grade_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
        if 'The' in j.split(): # this will split each of the catogory based on space "M
            j=j.replace('The','') # if we have the words "The" we are going to replace
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"M
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the trailing spa
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

project_data['project_grade_category'] = cat_list

#link : https://www.datacamp.com/community/tutorials/categorical-data
project_data = project_data.fillna(project_data['project_grade_category'].value_counts(

# 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['project_grade_category'].values:
    word = str(word)
    my_counter.update(word.split())

# dict sort by value python: https://stackoverflow.com/a/613218/4084039
project_grade_category_dict = dict(my_counter)
sorted_project_grade_category_dict = dict(sorted(project_grade_category_dict.items(), k
```

1.2 Text preprocessing

1.2.1 Text Preprocessing of essay

In [11]:

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

In [12]:

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

Out[12]:

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

In [13]:

```
# printing some random reviews
print(project_data['essay'].values[20])
print("="*125)
print(project_data['essay'].values[120])
print("="*125)
print(project_data['essay'].values[2020])
print("="*125)
print(project_data['essay'].values[41220])
print("="*125)
print(project_data['essay'].values[99920])
print("="*125)
```

```
Throughout this school year, I hope to enable my students to develop a lov
e for reading, discovering, and creating. Students in my classroom have th
e opportunity to explore learning through large group activities, small gr
oup learning times, and interactive play with other students and teachers.
My school is a Pre-K through 8 school with teachers and staff who are dete
rmined to help children in Detroit succeed. Students in my classroom are j
ust beginning their education in Pre-K and and I strive to enable my stude
nts to shine brightly as they learn using the High Scope Curriculum.The sh
opping carts would be a wonderful addition to our dramatic play area. With
them, the children will be able to work on their social skills. They will
learn about sharing, purchasing items, and the cost of different items. Si
milarly, the parachute will be used to encourage cooperative play. The chi
ldren will have fun learning and exercising. I'm looking forward to adding
the beads to our art area in the classroom to encourage both creativity as
well as fine motor skills as the children work on stringing them.Donations
to this project will allow me to encourage my students' curiosity and deve
lop a love for learning in them.\r\nWith these resources, my students will
start off on the right path in their education
==========================================================================
==================================================
Every afternoon we extend our learning through creative play. Creative pre
pares kids to be next generation innovators!My students' location does not
define their ability or determine their future.  Our city has been voted t
he \"Most Dangerous\" and \"Poorest\" city in the United States consistent
ly in the past 30 years.  Fortunately these labels are not actual descript
ions of who we are.  My students are innocent, hardworking, energetic, inq
uisitive and zealous for knowledge.  They are young leaders fighting for a
21st century education. \r\nIn my school, 100% of our eager students recei
ve free breakfast and lunch.  100% of our dedicated students are thirsty t
o learn.  100% of our teachers are driving our students to grow beyond a y
ear's growth.  We are in the fight and we will win.We are in need of resou
rces that will spark our creativity and open our minds to possibilities. T
he light table will supply a different way to view material. While the kin
etic sand and water bead will allow them to use their senses to create new
artifacts that extend their thinking and imagine boundless ideas.  These
tools will encourage discussion and team building techniques.These resourc
es will fuel our learning as they open their minds to what's possible. The
y will encourage my students to take change, problem solve, and collaborat
e. They will be come creative thinkers and doers.  It will also help my st
udents cope in their everyday lives.
==========================================================================
==================================================
I have long dreamed of teaching Angels in America, a play for my AP studen
ts that stimulated their thoughts and understand the universal promise of
the American dream.My students come from extremely diverse backgrounds.\r
\n\r\nThere are students who have fled war-torn countries such as the Ukra
ine, to first generation Mexican-Americans, to students dealing with Asber
```

gers and even a student who is in the advanced stages of Muscular Dystroph y. Through all their struggles, they are extremely resilient and come to s chool every day with hopes of a better future.In class we will be reading \"Angels in America\" together and discussing in Socratic seminars and wri ting papers on themes such as: visions of America, magical realism, the ne ed for a sense of community in our lives, and how caustic and demeaning st ereotyping can be.AP students are at an age in their lives where they are ready to see the world through many lenses. These unique perspectives mak e they not only better readers and writers, but also more prepared for the world they are entering.

==========

Hi, I work at a Title one school where must of our students have free or r educe lunch. My students are very bright and smart. They are always asking a lot of questions and my Busy Bees love to move! They are almost never st ill, but I love it because I enjoy moving and finding resources that incor porate dancing and jumping. Unfortunately, 90% of my students have experie nced some type of life changing event at an early age. I believe my job is to create a safe haven for these students while they are in my presence.Ha ving the set of tablets will allow for different structures of learning in my classroom. On way they will be used is as a STEM center. Students can l ook up pictures and then design their own using nothing more than Popsicle sticks, rubber bands, and Pom-poms. Another way they will be used is by re cording the results of an experiment. Recently, we investigated how Gobsto ppers reacted when placed in various liquids. Students could take pictures of their results at different time intervals and record observations throu gh notes. They could collaborate on group findings by sharing through the cloud. I could download apps for various math practice skills for a math c enter, which would allow me to have activities for both my highest perform ing and most struggling students. My project will make a difference becaus e, as I said before, these students are the digital natives.Mrs.Mrs.

==========

I have a large population of Haitian students in our beautiful classroom i n P.H. Elementary in Florida. Many enter my class struggling with the Engl ish language and we work very hard to keep up with the Common Core Standar ds. I work in a Title I school and my students live in high poverty. It's a very old school building and needles to say supplies of any kind are sho rt in demand. Luckily love and determination are in abundance in my young hard workers.We don't have such beautiful supplies that can quickly engage the mind of the learner and show them that there are so many ways to learn through play. We have Center-Time in our classrooms which must show them t o learn independently and these types of supplies would push them to creat e exemplary center work. These games will allow them to grow academically while actually playing a game. What is now undeniably clear in the 21st ce ntury is that play is essential, vital, critical, and fundamental to a chi ld’s social, emotional, physical, and intellectual development. These type s of supplies hold these words to be true in a learning classroom.Mrs.Mrs.

==========

In [14]:

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

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

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

In [15]:

```
sent = decontracted(project_data['essay'].values[2020])
print(sent)
print("="*120)
```

```
I have long dreamed of teaching Angels in America, a play for my AP studen
ts that stimulated their thoughts and understand the universal promise of
the American dream.My students come from extremely diverse backgrounds.\r
\n\r\nThere are students who have fled war-torn countries such as the Ukra
ine, to first generation Mexican-Americans, to students dealing with Asber
gers and even a student who is in the advanced stages of Muscular Dystroph
y. Through all their struggles, they are extremely resilient and come to s
chool every day with hopes of a better future.In class we will be reading
\"Angels in America\" together and discussing in Socratic seminars and wri
ting papers on themes such as: visions of America, magical realism, the ne
ed for a sense of community in our lives, and how caustic and demeaning st
ereotyping can be.AP students are at an age in their lives where they are
ready to see the world through many lenses.  These unique perspectives mak
e they not only better readers and writers, but also more prepared for the
world they are entering.
========================================================================
========================================
```

In [16]:

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

```
I have long dreamed of teaching Angels in America, a play for my AP studen
ts that stimulated their thoughts and understand the universal promise of
the American dream.My students come from extremely diverse backgrounds.
There are students who have fled war-torn countries such as the Ukraine, t
o first generation Mexican-Americans, to students dealing with Asbergers a
nd even a student who is in the advanced stages of Muscular Dystrophy. Thr
ough all their struggles, they are extremely resilient and come to school
every day with hopes of a better future.In class we will be reading  Angel
s in America  together and discussing in Socratic seminars and writing pap
ers on themes such as: visions of America, magical realism, the need for a
sense of community in our lives, and how caustic and demeaning stereotypin
g can be.AP students are at an age in their lives where they are ready to
see the world through many lenses.  These unique perspectives make they no
t only better readers and writers, but also more prepared for the world th
ey are entering.
```

In [17]:

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

```
I have long dreamed of teaching Angels in America a play for my AP student
s that stimulated their thoughts and understand the universal promise of t
he American dream My students come from extremely diverse backgrounds Ther
e are students who have fled war torn countries such as the Ukraine to fir
st generation Mexican Americans to students dealing with Asbergers and eve
n a student who is in the advanced stages of Muscular Dystrophy Through al
l their struggles they are extremely resilient and come to school every da
y with hopes of a better future In class we will be reading Angels in Amer
ica together and discussing in Socratic seminars and writing papers on the
mes such as visions of America magical realism the need for a sense of com
munity in our lives and how caustic and demeaning stereotyping can be AP s
tudents are at an age in their lives where they are ready to see the world
through many lenses These unique perspectives make they not only better re
aders and writers but also more prepared for the world they are entering
```

In [18]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ["a","about","above","after","again","against","ain","all","am","an","and","
            "as","at","be","because","been","before","being","below","between","both",
            "d","did","didn","didn't","do","does","doesn","doesn't","doing","don","don
            "for","from","further","had","hadn","hadn't","has","hasn","hasn't","have",
            "here","hers","herself","him","himself","his","how","i","if","in","into","
            "itself","just","ll","m","ma","me","mightn","mightn't","more","most","must
            "needn't","no","nor","not","now","o","of","off","on","once","only","or","o
            "out","over","own","re","s","same","shan","shan't","she","she's","should",
            "so","some","such","t","than","that","that'll","the","their","theirs","the
            "these","they","this","those","through","to","too","under","until","up","v
            "we","were","weren","weren't","what","when","where","which","while","who",
            "won't","wouldn","wouldn't","y","you","you'd","you'll","you're","you've","
            "yourselves","could","he'd","he'll","he's","here's","how's","i'd","i'll","
            "she'd","she'll","that's","there's","they'd","they'll","they're","they've"
            "what's","when's","where's","who's","why's","would","able","abst","accorda
            "across","act","actually","added","adj","affected","affecting","affects","
            "along","already","also","although","always","among","amongst","announce",
            "anymore","anyone","anything","anyway","anyways","anywhere","apparently","
            "around","aside","ask","asking","auth","available","away","awfully","b","b
            "becoming","beforehand","begin","beginning","beginnings","begins","behind"
            "beyond","biol","brief","briefly","c","ca","came","cannot","can't","cause"
            "co","com","come","comes","contain","containing","contains","couldnt","dat
            "due","e","ed","edu","effect","eg","eight","eighty","either","else","elsewh
            "especially","et","etc","even","ever","every","everybody","everyone","ever
            "f","far","ff","fifth","first","five","fix","followed","following","follow
            "found","four","furthermore","g","gave","get","gets","getting","give","giv
            "gone","got","gotten","h","happens","hardly","hed","hence","hereafter","he
            "hes","hi","hid","hither","home","howbeit","however","hundred","id","ie","
            "importance","important","inc","indeed","index","information","instead","i
            "it'll","j","k","keep","keeps","kept","kg","km","know","known","knows","l"
            "later","latter","latterly","least","less","lest","let","lets","like","lik
            "'ll","look","looking","looks","ltd","made","mainly","make","makes","many"
            "meantime","meanwhile","merely","mg","might","million","miss","ml","moreov
            "mug","must","n","na","name","namely","nay","nd","near","nearly","necessar
            "neither","never","nevertheless","new","next","nine","ninety","nobody","no
            "normally","nos","noted","nothing","nowhere","obtain","obtained","obviousl
            "omitted","one","ones","onto","ord","others","otherwise","outside","overal
            "particular","particularly","past","per","perhaps","placed","please","plus
            "potentially","pp","predominantly","present","previously","primarily","pro
            "provides","put","q","que","quickly","quite","qv","r","ran","rather","rd",
            "recently","ref","refs","regarding","regardless","regards","related","rela
            "resulted","resulting","results","right","run","said","saw","say","saying"
            "seeing","seem","seemed","seeming","seems","seen","self","selves","sent","
            "shes","show","showed","shown","showns","shows","significant","significant
            "six","slightly","somebody","somehow","someone","somethan","something","so
            "somewhere","soon","sorry","specifically","specified","specify","specifyin
            "sub","substantially","successfully","sufficiently","suggest","sup","sure"
            "tends","th","thank","thanks","thanx","thats","that've","thence","thereaft
            "therein","there'll","thereof","therere","theres","thereto","thereupon","t
            "thou","though","thoughh","thousand","throug","throughout","thru","thus","
            "toward","towards","tried","tries","truly","try","trying","ts","twice","tw
            "unless","unlike","unlikely","unto","upon","ups","us","use","used","useful
            "using","usually","v","value","various","'ve","via","viz","vol","vols","vs
            "wed","welcome","went","werent","whatever","what'll","whats","whence","whe
            "whereby","wherein","wheres","whereupon","wherever","whether","whim","whit
            "who'll","whomever","whos","whose","widely","willing","wish","within","wit
            "wouldnt","www","x","yes","yet","youd","youre","z","zero","a's","ain't","a
```

```
            "appreciate","appropriate","associated","best","better","c'mon","c's","can
            "consequently","consider","considering","corresponding","course","currentl
            "entirely","exactly","example","going","greetings","hello","help","hopeful
            "indicated","indicates","inner","insofar","it'd","keep","keeps","novel","p
            "secondly","sensible","serious","seriously","sure","t's","third","thorough
            "wonder"]
```

In [19]:

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

```
Wall time: 0 ns

100%|██████████████████████████████████████████████████████████████████████
████| 109248/109248 [06:24<00:00, 284.23it/s]
```

In [20]:

```
# after preprocesing
preprocessed_essays[2020]
```

Out[20]:

```
'long dreamed teaching angels america play ap students stimulated thoughts
understand universal promise american dream students extremely diverse bac
kgrounds students fled war torn countries ukraine generation mexican ameri
cans students dealing asbergers student advanced stages muscular dystrophy
struggles extremely resilient school day hopes future class reading angels
america discussing socratic seminars writing papers themes visions america
magical realism sense community lives caustic demeaning stereotyping ap st
udents age lives ready lenses unique perspectives readers writers prepared
entering'
```

1.2.2 Text Preprocessing of project_title

In [21]:

```
print(project_data['project_title'].tail(1))
```

```
78306    News for Kids
Name: project_title, dtype: object
```

In [22]:

```
# printing some random title texts
print(project_data['project_title'].values[20])
print('--'*20)
print(project_data['project_title'].values[120])
print('--'*20)
print(project_data['project_title'].values[2020])
print('--'*20)
print(project_data['project_title'].values[99920])
print('--'*20)
```

```
Pre-K Classroom Materials
----------------------------------------
Empowering Kindergarteners
----------------------------------------
Angels in America - A Must Read
----------------------------------------
Turn That TV Game Off and Turn on the Learning!
----------------------------------------
```

In [23]:

```
# 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 [24]:

```
sent = decontracted(project_data['project_title'].values[99999])
print(sent)
print("="*120)
```

```
Turning to Flexible Seating: One Sixth-Grade Class is Journey to Freedom
========================================================================================================================
```

In [25]:

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

```
Turning to Flexible Seating: One Sixth-Grade Class is Journey to Freedom
```

In [26]:

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

```
Turning to Flexible Seating One Sixth Grade Class is Journey to Freedom
```

In [27]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ["a","about","above","after","again","against","ain","all","am","an","and","
            "as","at","be","because","been","before","being","below","between","both",
            "d","did","didn","didn't","do","does","doesn","doesn't","doing","don","don
            "for","from","further","had","hadn","hadn't","has","hasn","hasn't","have",
            "here","hers","herself","him","himself","his","how","i","if","in","into","
            "itself","just","ll","m","ma","me","mightn","mightn't","more","most","must
            "needn't","no","nor","not","now","o","of","off","on","once","only","or","o
            "out","over","own","re","s","same","shan","shan't","she","she's","should",
            "so","some","such","t","than","that","that'll","the","their","theirs","the
            "these","they","this","those","through","to","too","under","until","up","v
            "we","were","weren","weren't","what","when","where","which","while","who",
            "won't","wouldn","wouldn't","y","you","you'd","you'll","you're","you've","
            "yourselves","could","he'd","he'll","he's","here's","how's","i'd","i'll","
            "she'd","she'll","that's","there's","they'd","they'll","they're","they've"
            "what's","when's","where's","who's","why's","would","able","abst","accorda
            "across","act","actually","added","adj","affected","affecting","affects","
            "along","already","also","although","always","among","amongst","announce",
            "anymore","anyone","anything","anyway","anyways","anywhere","apparently","
            "around","aside","ask","asking","auth","available","away","awfully","b","b
            "becoming","beforehand","begin","beginning","beginnings","begins","behind"
            "beyond","biol","brief","briefly","c","ca","came","cannot","can't","cause"
            "co","com","come","comes","contain","containing","contains","couldnt","dat
            "due","e","ed","edu","effect","eg","eight","eighty","either","else","elsewh
            "especially","et","etc","even","ever","every","everybody","everyone","ever
            "f","far","ff","fifth","first","five","fix","followed","following","follow
            "found","four","furthermore","g","gave","get","gets","getting","give","giv
            "gone","got","gotten","h","happens","hardly","hed","hence","hereafter","he
            "hes","hi","hid","hither","home","howbeit","however","hundred","id","ie","
            "importance","important","inc","indeed","index","information","instead","i
            "it'll","j","k","keep","keeps","kept","kg","km","know","known","knows","l"
            "later","latter","latterly","least","less","lest","let","lets","like","lik
            "'ll","look","looking","looks","ltd","made","mainly","make","makes","many"
            "meantime","meanwhile","merely","mg","might","million","miss","ml","moreov
            "mug","must","n","na","name","namely","nay","nd","near","nearly","necessar
            "neither","never","nevertheless","new","next","nine","ninety","nobody","no
            "normally","nos","noted","nothing","nowhere","obtain","obtained","obviousl
            "omitted","one","ones","onto","ord","others","otherwise","outside","overal
            "particular","particularly","past","per","perhaps","placed","please","plus
            "potentially","pp","predominantly","present","previously","primarily","pro
            "provides","put","q","que","quickly","quite","qv","r","ran","rather","rd",
            "recently","ref","refs","regarding","regardless","regards","related","rela
            "resulted","resulting","results","right","run","said","saw","say","saying"
            "seeing","seem","seemed","seeming","seems","seen","self","selves","sent","
            "shes","show","showed","shown","showns","shows","significant","significant
            "six","slightly","somebody","somehow","someone","somethan","something","so
            "somewhere","soon","sorry","specifically","specified","specify","specifyin
            "sub","substantially","successfully","sufficiently","suggest","sup","sure"
            "tends","th","thank","thanks","thanx","thats","that've","thence","thereaft
            "therein","there'll","thereof","therere","theres","thereto","thereupon","t
            "thou","though","thoughh","thousand","throug","throughout","thru","thus","
            "toward","towards","tried","tries","truly","try","trying","ts","twice","tw
            "unless","unlike","unlikely","unto","upon","ups","us","use","used","useful
            "using","usually","v","value","various","'ve","via","viz","vol","vols","vs
            "wed","welcome","went","werent","whatever","what'll","whats","whence","whe
            "whereby","wherein","wheres","whereupon","wherever","whether","whim","whit
            "who'll","whomever","whos","whose","widely","willing","wish","within","wit
            "wouldnt","www","x","yes","yet","youd","youre","z","zero","a's","ain't","a
```

```
            "appreciate","appropriate","associated","best","better","c'mon","c's","can
            "consequently","consider","considering","corresponding","course","currentl
            "entirely","exactly","example","going","greetings","hello","help","hopeful
            "indicated","indicates","inner","insofar","it'd","keep","keeps","novel","p
            "secondly","sensible","serious","seriously","sure","t's","third","thorough
            "wonder"]
```

In [28]:

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

```
100%|██████████████████████████████████████████████████████████████████████████
███| 109248/109248 [00:13<00:00, 8138.44it/s]

Wall time: 13.4 s
```

In [29]:

```
# after preprocessed
preprocessed_project_title[99920]
```

Out[29]:

```
'turn tv game turn learning'
```

1.3. Numerical normalization

1.3.1 normalization_price

In [30]:

```
# merge data frames
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'}).reset_i
project_data = pd.merge(project_data, price_data, on='id', how='left')
project_data.shape
```

Out[30]:

```
(109248, 20)
```

In [31]:

```
project_data.head(1)
```

Out[31]:

Unnamed: 0	id	teacher_id	teacher_prefix	school_state	Dat
0	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs.	CA	2016-04-2 00:27:3

In [32]:

```
print(project_data["price"].shape)
```

```
(109248,)
```

In [33]:

```
# Link: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Normali
# Reshaping price data using array.reshape(1,-1)
from sklearn.preprocessing import Normalizer
# Reshaping price data using array.reshape(1,-1)
price_normalize = Normalizer()
price_normalizer = price_normalize.fit_transform(project_data['price'].values.reshape(1
price_normalizer = price_normalizer.T
print(price_normalizer)
print("-------------------------------------------------------")
print("shape of price_normalizer:", price_normalizer.shape)
```

```
[[4.63560392e-03]
 [1.36200635e-03]
 [2.10346002e-03]
 ...
 [2.55100471e-03]
 [1.83960046e-03]
 [3.51642253e-05]]
-------------------------------------------------------
shape of price_normalizer: (109248, 1)
```

1.3.2 Normalization of teacher_number_of_previously_posted_projects (tnppp)

In [34]:

```
normalizer = Normalizer()
tnppp_normalizer = normalizer.fit_transform(project_data["teacher_number_of_previously_
tnppp_normalizer = tnppp_normalizer.T
print(tnppp_normalizer)
print("--------------------------------")
print("Shape of tnppp:", tnppp_normalizer)
```

```
[[0.00535705]
 [0.00040431]
 [0.00101076]
 ...
 [0.        ]
 [0.00010108]
 [0.00020215]]
--------------------------------
Shape of tnppp: [[0.00535705]
 [0.00040431]
 [0.00101076]
 ...
 [0.        ]
 [0.00010108]
 [0.00020215]]
```

1.4. Vectorizing Categorical data

1.4.1 Vectorization of project_subject_categories

- 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 [35]:

```
# we use count vectorizer to convert the values into one
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_cat = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()), lowercase=Fal
clean_categories_one_hot = vectorizer_cat.fit_transform(project_data['clean_categories'
print("vectorizer of clean_categories feature names :", vectorizer_cat.get_feature_name
print("------------------------------------------------------------")
print("Shape of clean_categories_one_hot encodig : ",clean_categories_one_hot.shape)
```

```
vectorizer of clean_categories feature names : ['Warmth', 'Care_Hunger',
'History_Civics', 'Music_Arts', 'AppliedLearning', 'SpecialNeeds', 'Health
_Sports', 'Math_Science', 'Literacy_Language']
------------------------------------------------------------
Shape of clean_categories_one_hot encodig :  (109248, 9)
```

1.4.2 vectorization of project_subject_subcategories

In [36]:

```
# we count vectorizer to convert the values into one
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_subcat = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys()), lowerc
clean_subcat_one_hot = vectorizer_subcat.fit_transform(project_data['clean_subcategorie
print("vectorizer of subcategories feature names:",vectorizer_subcat.get_feature_names(
print(40*"-")
print("Shape of subcategories on hot encoding:", clean_subcat_one_hot.shape)
```

```
vectorizer of subcategories feature names: ['Economics', 'CommunityServic
e', 'FinancialLiteracy', 'ParentInvolvement', 'Extracurricular', 'Civics_G
overnment', 'ForeignLanguages', 'NutritionEducation', 'Warmth', 'Care_Hung
er', 'SocialSciences', 'PerformingArts', 'CharacterEducation', 'TeamSport
s', 'Other', 'College_CareerPrep', 'Music', 'History_Geography', 'Health_L
ifeScience', 'EarlyDevelopment', 'ESL', 'Gym_Fitness', 'EnvironmentalScien
ce', 'VisualArts', 'Health_Wellness', 'AppliedSciences', 'SpecialNeeds',
'Literature_Writing', 'Mathematics', 'Literacy']
----------------------------------------
Shape of subcategories on hot encoding: (109248, 30)
```

1.4.3 Vectorization of school_state

In [37]:

```
# we use count vectorizer to convert the values into one
vectorizer_ss = CountVectorizer(vocabulary=list(sorted_school_state_dict.keys()), lower
school_state_one_hot = vectorizer_ss.fit_transform(project_data['school_state'].values)
print("vectorizer of school_state feature names :",vectorizer_ss.get_feature_names())
print("----------------------------------------------------------")
print("Shape of matrix after one hot encodig : ",school_state_one_hot.shape)
```

```
vectorizer of school_state feature names : ['VT', 'WY', 'ND', 'MT', 'RI',
'SD', 'NE', 'DE', 'AK', 'NH', 'WV', 'ME', 'HI', 'DC', 'NM', 'KS', 'IA', 'I
D', 'AR', 'CO', 'MN', 'OR', 'KY', 'MS', 'NV', 'MD', 'CT', 'TN', 'UT', 'A
L', 'WI', 'VA', 'AZ', 'NJ', 'OK', 'WA', 'MA', 'LA', 'OH', 'MO', 'IN', 'P
A', 'MI', 'SC', 'GA', 'IL', 'NC', 'FL', 'NY', 'TX', 'CA']
----------------------------------------------------------
Shape of matrix after one hot encodig :  (109248, 51)
```

1.4.4 Vectorization of teacher_prefix

In [38]:

```
# we use count vectorizer to convert the values into one
vectorizer_tp = CountVectorizer(vocabulary=list(sorted_teacher_prefix_dict.keys()), lowe
teacher_prefix_one_hot = vectorizer_tp.fit_transform(project_data['teacher_prefix'].valu
print(" vectorizer of teacher prefix feature names:",vectorizer_tp.get_feature_names())
print(100*"-")
print("Shape of teacher prefix one hot encoding Matrix :", teacher_prefix_one_hot.shape
```

```
 vectorizer of teacher prefix feature names: ['Dr.', 'Teacher', 'Mr.', 'M
s.', 'Mrs.']
----------------------------------------------------------------------
--------------------------
Shape of teacher prefix one hot encoding Matrix : (109248, 5)
```

In [39]:

```
vectorizer_cat.get_feature_names
```

Out[39]:

```
<bound method CountVectorizer.get_feature_names of CountVectorizer(analyze
r='word', binary=True, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='cont
ent',
                lowercase=False, max_df=1.0, max_features=None, min_df=1,
                ngram_range=(1, 1), preprocessor=None, stop_words=None,
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None,
                vocabulary=['Warmth', 'Care_Hunger', 'History_Civics',
                            'Music_Arts', 'AppliedLearning', 'SpecialNeed
s',
                            'Health_Sports', 'Math_Science',
                            'Literacy_Language'])>
```

1.4.5 Vectorization of project_grade_categorie

In [40]:

```
# we use count vectorizer to convert the values into one hot encoding feature
vectorizer_pgc = CountVectorizer(vocabulary=list(sorted_project_grade_category_dict.keys
project_grade_cat_one_hot = vectorizer_pgc.fit_transform(project_data['project_grade_cat
print("vectorizer of project grade category feature names: ", vectorizer_pgc.get_feature
print("Shape of project grade category one hoted matrix:", project_grade_cat_one_hot.sha
```

```
vectorizer of project grade category feature names:  ['Grades9-12', 'Grade
s6-8', 'Grades3-5', 'GradesPreK-2']
Shape of project grade category one hoted matrix: (109248, 4)
```

1.5. Vectorizing Text

1.5.1 Vectorization of essays bow

In [41]:

```
project_data['essay'].tail(5)
```

Out[41]:

```
109243    Our day starts with about 100 students athlete...
109244    My students range from age four to five years ...
109245    We are a Title 1 school  650 total students. O...
109246    I teach many different types of students.  My ...
109247    My first graders are eager to learn about the ...
Name: essay, dtype: object
```

In [42]:

```
# we are considering only the words which appeared in at least 10 documents (rows or pr
vectorizer_essays_bow = CountVectorizer(preprocessed_essays, min_df=10, max_features=10
essays_bow_one_hot = vectorizer_essays_bow.fit_transform(project_data['essay'].values)
print("Shape of essays bow matrix after one hot encodig  : ",essays_bow_one_hot.shape)
```

```
Shape of essays bow matrix after one hot encodig  :  (109248, 10000)
```

1.5.1.1 Vectorization of essays TFIDF

In [43]:

```
# we are considering only the words which appeared in at least 10 documents (rows or pr
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer_essays_tfidf = TfidfVectorizer(preprocessed_essays, min_df=10, max_features=
essays_tfidf_one_hot = vectorizer_essays_tfidf.fit_transform(project_data['essay'].valu
print("Shape of essays_tfidf after one hot encoding:", essays_tfidf_one_hot.shape)
```

```
Shape of essays_tfidf after one hot encoding: (109248, 10000)
```

1.5.2 Vectorization of project_title bow

In [44]:

```
# we are considering only the words which appeared in at least 10 documents
vectorizer_proj_title_bow = CountVectorizer(preprocessed_project_title, min_df=10, max_
project_title_tfidf_one_hot = vectorizer_proj_title_bow.fit_transform(project_data['pro
print("Shape of project_title after one hot encoding :", project_title_tfidf_one_hot.sh
```

```
Shape of project_title after one hot encoding : (109248, 3349)
```

1.5.2.1 Vectorization of prject_title TFIDF

In [45]:

```
# we are consodering only the words which appeared in at least 10 documents
vectorizer_proj_title_tfidf = TfidfVectorizer(preprocessed_project_title, min_df=10, ng
project_title_tfidf_one_hot = vectorizer_proj_title_tfidf.fit_transform(project_data['p
print("shape of project title tfidf after one hot encoding :", project_title_tfidf_one_
```

```
shape of project title tfidf after one hot encoding : (109248, 3349)
```

In [46]:

```
project_data.columns
```

Out[46]:

```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'Date', 'project_grade_category', 'project_title', 'project_essay_
1',
       'project_essay_2', 'project_essay_3', 'project_essay_4',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approve
d',
       '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 (optinal)

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

1.5.3 Merging all the above features

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

In [47]:

```
print("Shape of clean categories      :", clean_categories_one_hot.shape)
print("Shape of clean subcategories   :", clean_subcat_one_hot.shape)
print("Shape of school steate         :", school_state_one_hot.shape)
print("Shape of teacher prefix        :", teacher_prefix_one_hot.shape)
print("Shape of project grade category:", project_grade_cat_one_hot.shape)
print("Shape of essays tfidf          :", essays_tfidf_one_hot.shape)
print("Shape of project title tfidf   :", project_title_tfidf_one_hot.shape)
print("Shape of price                 :", price_normalizer.shape)
print("Shape of tnppp                 :", tnppp_normalizer.shape)
```

```
Shape of clean categories      : (109248, 9)
Shape of clean subcategories   : (109248, 30)
Shape of school steate         : (109248, 51)
Shape of teacher prefix        : (109248, 5)
Shape of project grade category: (109248, 4)
Shape of essays tfidf          : (109248, 10000)
Shape of project title tfidf   : (109248, 3349)
Shape of price                 : (109248, 1)
Shape of tnppp                 : (109248, 1)
```

Assignment : Clustering

- step 1: Choose any vectorizer (data matrix) that you have worked in any of the assignments, and got the best AUC value.
- step 2: Choose any of the feature selection (https://scikit-learn.org/stable/modules/feature_selection.html)/reduction algorithms (https://scikit-learn.org/stable/modules/decomposition.html) ex: selectkbest features, pretrained word vectors, model based feature selection etc and reduce the number of features to 5k features.
- step 3: Apply all three kmeans, Agglomerative clustering, DBSCAN
  - **K-Means Clustering:**
    - Find the best 'k' using the elbow-knee method (plot k vs inertia_)
  - **Agglomerative Clustering:**
    - Apply agglomerative algorithm (https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/) and try a different number of clusters like 2,5 etc.
    - As this is very computationally expensive, take **5k** datapoints only to perform hierarchical clustering because they do take a considerable amount of time to run.
  - **DBSCAN Clustering:**
    - Find the best 'eps' using the elbow-knee method (https://stackoverflow.com/a/48558030/4084039).
    - Take **5k** datapoints only.
- step 4: Summarize each cluster by manually observing few points from each cluster.
- step 5: You need to plot the word cloud with essay text for each cluster for each of algorithms mentioned in step 3.

2. Clustering

2.1 Choosen the best data matrix on which you got the best AUC

In [48]:

```
# i have taken  the TFIDF set as the heighest AUC in logestic regression
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
# with the same hstack function we are concatenating a sparse matrix and dense matrix
# set = all categorical + Numarical + Text

set_tfidf = hstack((clean_categories_one_hot, clean_subcat_one_hot, teacher_prefix_one_
                project_grade_cat_one_hot,essays_tfidf_one_hot, project_title_tfidf_one_
                     tnppp_normalizer)).tocsr()
print("Shape : set tfidf:", set_tfidf.shape)
```

```
Shape : set tfidf: (109248, 13450)
```

2.2 Dimensionality Reduction on the selected features

In [55]:

```
from sklearn.feature_selection import SelectKBest, f_classif
y = project_data['project_is_approved']
selector = SelectKBest(f_classif, k=5000)
X_new = selector.fit_transform(set_tfidf,y)
print(X_new.shape)
```

```
(109248, 5000)
```

In [60]:

```
from sklearn.decomposition import TruncatedSVD
svd = TruncatedSVD()
best_data_5k = svd.fit(X_new)
print("Shape of bes_data_5k:", best_data_5k)
```

```
Shape of bes_data_5k: TruncatedSVD(algorithm='randomized', n_components=2,
n_iter=5,
             random_state=None, tol=0.0)
```

2.3 Apply Kmeans

In [62]:

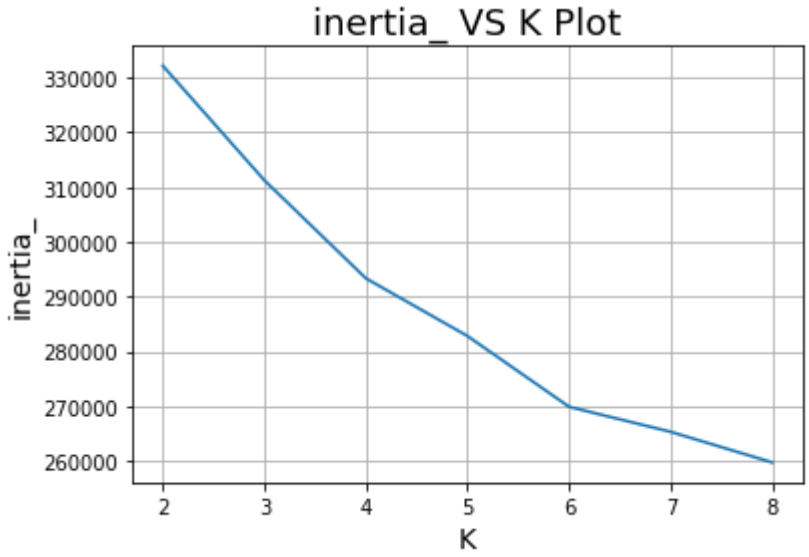
```
from sklearn.cluster import KMeans

k_values = [2,3,4,5,6,7,8]
inertia_ = []
for i in k_values:
    kmeans = KMeans(n_clusters=i, n_jobs=-1).fit(X_new)
    inertia_.append(kmeans.inertia_)
```

2.3.1 finding the best k using elbow-knee method

In [63]:

```
plt.plot(k_values, inertia_)
plt.xlabel('K',size=14)
plt.ylabel('inertia_',size=14)
plt.title('inertia_ VS K Plot',size=18)
plt.grid()
plt.show()
```



In [64]:

```
optimal_k = 6
kmeans = KMeans(n_clusters=optimal_k, n_jobs=-1).fit(X_new)
```

2.3.3 Ploting wordcloud with essay for each cluster - KMeans

In [66]:

```
essays = project_data['essay'].values

cluster1 = []
cluster2 = []
cluster3 = []
cluster4 = []
cluster5 = []
cluster6 = []
for i in range(kmeans.labels_.shape[0]):
    if kmeans.labels_[i] == 0:
        cluster1.append(essays[i])
    elif kmeans.labels_[i] == 1:
        cluster2.append(essays[i])
    elif kmeans.labels_[i] == 2:
        cluster3.append(essays[i])
    elif  kmeans.labels_[i] == 3:
        cluster4.append(essays[i])
    elif  kmeans.labels_[i] == 4:
        cluster5.append(essays[i])
    elif  kmeans.labels_[i] == 5:
        cluster6.append(essays[i])
```

In [68]:

```
print((cluster1[i]))
```

```
Imagine being 8-9 years old. You're in your third grade classroom. You see
bright lights, the kid next to you is chewing gum, the birds are making no
ise, the street outside is buzzing with cars, it's hot, and your teacher i
s asking you to focus on learning. Ack! You need a break! So do my student
s.Most of my students have autism, anxiety, another disability, or all of
the above. It is tough to focus in school due to sensory overload or emoti
ons. My students have a lot to deal with in school, but I think that makes
them the most incredible kids on the planet. They are kind, caring, and sy
mpathetic. They know what it's like to be overwhelmed, so they understand
when someone else is struggling. They are open-minded and compassionate. T
hey are the kids who will someday change the world.It is tough to do more
than one thing at a time. When sensory overload gets in the way, it is the
hardest thing in the world to focus on learning. My students need many bre
aks throughout the day, and one of the best items we've used is a Boogie B
oard. If we had a few in our own classroom, my students could take a break
exactly when they need one, regardless of which other rooms in the school
are occupied. Many of my students need to do something with their hands in
order to focus on the task at hand. Putty will give the sensory input they
need in order to focus, it will calm them when they are overloaded, it wil
l help improve motor skills, and it will make school more fun.When my stud
ents are able to calm themselves down, they are ready to learn. When they
are able to focus, they will learn more and retain more. They will get the
sensory input they need and it will prevent meltdowns (which are scary for
everyone in the room). This will lead to a better, happier classroom commu
nity that is able to learn the most they can in the best way possible.
```

In [71]:

```
for i in range(2):
    print((cluster2[i]))
```

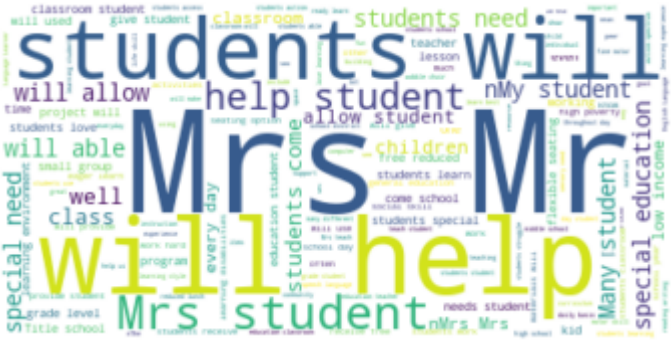
```
I have been fortunate enough to use the Fairy Tale STEM kits in my classro
om as well as the STEM journals, which my students really enjoyed.  I woul
d love to implement more of the Lakeshore STEM kits in my classroom for th
e next school year as they provide excellent and engaging STEM lessons.My
students come from a variety of backgrounds, including language and socioe
conomic status.  Many of them don't have a lot of experience in science an
d engineering and these kits give me the materials to provide these exciti
ng opportunities for my students.Each month I try to do several science or
STEM/STEAM projects.  I would use the kits and robot to help guide my scie
nce instruction in engaging and meaningful ways.  I can adapt the kits to
my current language arts pacing guide where we already teach some of the m
aterial in the kits like tall tales (Paul Bunyan) or Johnny Appleseed.  Th
e following units will be taught in the next school year where I will impl
ement these kits: magnets, motion, sink vs. float, robots.  I often get to
these units and don't know If I am teaching the right way or using the rig
ht materials.    The kits will give me additional ideas, strategies, and l
essons to prepare my students in science.It is challenging to develop high
quality science activities.  These kits give me the materials I need to pr
ovide my students with science activities that will go along with the curr
iculum in my classroom.  Although I have some things (like magnets) in my
classroom, I don't know how to use them effectively.  The kits will provid
e me with the right amount of materials and show me how to use them in an
appropriate way.
It's the end of the school year. Routines have run their course, and stude
nts are in need of a boost in the curriculum. Enter the Breakout Box! My s
tudents can prove their knowledge of content by solving riddles and findin
g clues that lead them to unlocking the box for the prize.My students desi
re challenges, movement, and collaboration. They thrive on rigor, but are
far more successful with content that is engaging and rich, but still fun.
My students have been working and developing on who they are as teammates,
and how one works best in a group. I work at a school that holds the bar f
or student achievement very high, and I would like to raise that bar even
further with a Breakout Box.I will design different clues using specific c
ontent knowledge to get students to solve puzzles in search of unlocking t
he Breakout Box. This activity can be used in all subject areas and have s
tudents working in teams, whole class, or even individually. I will be abl
e to use these materials year round to review the content of the unit in a
fun and engaging way that keeps students highly motivated to prove their l
earning.Donations to this project will immediately improve my classroom by
changing up our routines. Students are at the end of a long and difficult
year, which is when behaviors can begin to become an issue. A Breakout Box
will provide us with the fun and interactive way to break the routine but
still build up that cognitive sweat that is so important everyday. I will
also use these materials beyond this year, and incorporate this fun activi
ty as a review or assessment at the culmination of a unit.
```

2.3.3.1 Ploting wordcloud with essay for cluster1 - KMeans

In [72]:

```
#cluster 1
words=''
for i in cluster1:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

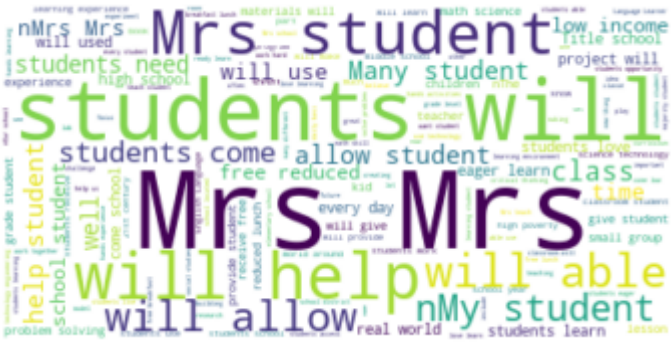


2.3.3.2 Ploting wordcloud with essay for cluster2 - KMeans

In [73]:

```
#cluster 2
words=''
for i in cluster2:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



2.3.3.3 Ploting wordcloud with essay for cluster3 and cluster4 - KMeans

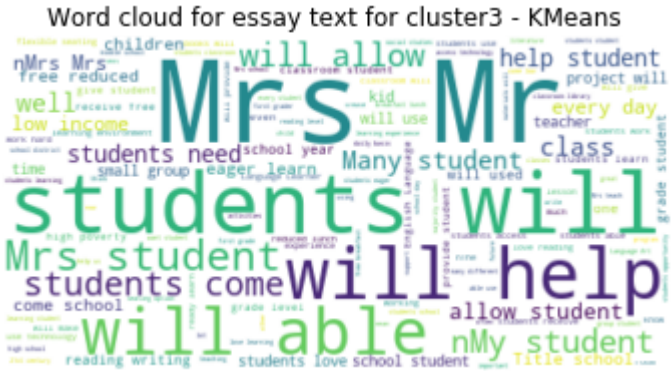
In [74]:

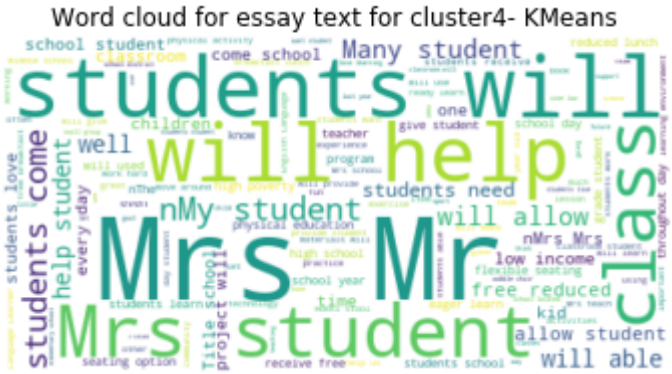
```
#cluster 3
words=''
for i in cluster3:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Word cloud for essay text for cluster3 - KMeans ")
plt.axis("off")
plt.show()
print("----------------------------------------------------------------------------------------------------")

#cluster 4
words=''
for i in cluster4:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Word cloud for essay text for cluster4- KMeans ")
plt.axis("off")
plt.show()
```





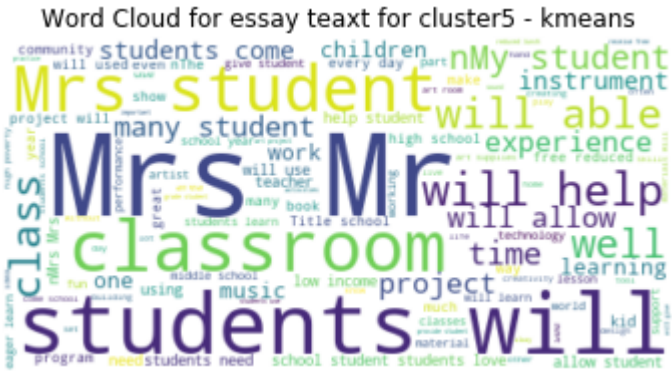
2.3.3.4 Ploting wordcloud with essay for cluster5 and cluster6 - KMeans

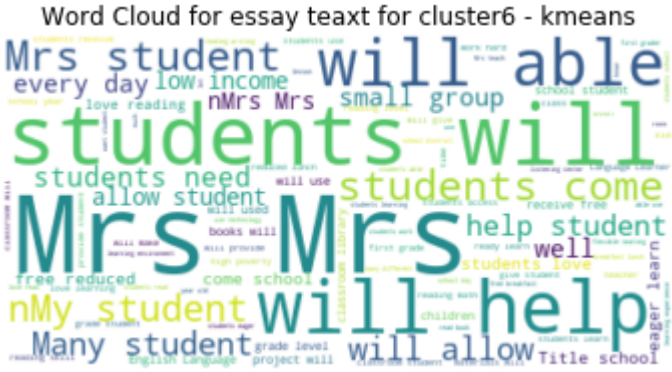
In [75]:

```
#cluster 5
words=''
for i in cluster5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Word Cloud for essay teaxt for cluster5 - kmeans")
plt.axis("off")
plt.show()
print('------------------------------------------------')
# cluser6
#cluster 6
words=''
for i in cluster6:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Word Cloud for essay teaxt for cluster6 - kmeans")
plt.axis("off")
plt.show()
```





2.4 Apply AgglomerativeClustering

In [77]:

```
# by doing dimentionality reduction for taking 5K datapoints to apply agglomerative clu
from sklearn.feature_selection import SelectKBest, chi2
X_new2 = SelectKBest(chi2,k=5000).fit_transform(X_new,y)
#########################################################################
print("Final Data matrix on TFIDF")
print(X_new2.shape )
```

```
Final Data matrix on TFIDF
(109248, 5000)
```

In [79]:

```
X_new_best5k = X_new2[:5000]
print("Shape of best 5k data:", X_new_best5k.shape)
```

```
Shape of best 5k data: (5000, 5000)
```

2.4.1 Apply AgglomerativeClustering for K =2

In [82]:

```
from sklearn.cluster import AgglomerativeClustering
AgglomCluster = AgglomerativeClustering(n_clusters=2 ).fit(X_new_best5k.toarray())
```

2.4.1.1 Ploting wordcloud with essay for cluster1 and cluster2 - AgglomerativeClustering

In [94]:

```
cluster1_agg=[]
cluster2_agg=[]
essays = project_data['essay'].values
for i in range(AgglomCluster.labels_.shape[0]):
    if AgglomCluster.labels_[i] == 0:
        cluster1_agg.append(essays[i])
    elif AgglomCluster.labels_[i] == 1:
        cluster2_agg.append(essays[i])
```

In [96]:

```
for i in range(2):
    print((cluster1_agg[i]))
```

I have been fortunate enough to use the Fairy Tale STEM kits in my classro
om as well as the STEM journals, which my students really enjoyed. I woul
d love to implement more of the Lakeshore STEM kits in my classroom for th
e next school year as they provide excellent and engaging STEM lessons.My
students come from a variety of backgrounds, including language and socioe
conomic status. Many of them don't have a lot of experience in science an
d engineering and these kits give me the materials to provide these exciti
ng opportunities for my students.Each month I try to do several science or
STEM/STEAM projects. I would use the kits and robot to help guide my scie
nce instruction in engaging and meaningful ways. I can adapt the kits to
my current language arts pacing guide where we already teach some of the m
aterial in the kits like tall tales (Paul Bunyan) or Johnny Appleseed. Th
e following units will be taught in the next school year where I will impl
ement these kits: magnets, motion, sink vs. float, robots. I often get to
these units and don't know If I am teaching the right way or using the rig
ht materials.   The kits will give me additional ideas, strategies, and l
essons to prepare my students in science.It is challenging to develop high
quality science activities. These kits give me the materials I need to pr
ovide my students with science activities that will go along with the curr
iculum in my classroom. Although I have some things (like magnets) in my
classroom, I don't know how to use them effectively. The kits will provid
e me with the right amount of materials and show me how to use them in an
appropriate way.
Imagine being 8-9 years old. You're in your third grade classroom. You see
bright lights, the kid next to you is chewing gum, the birds are making no
ise, the street outside is buzzing with cars, it's hot, and your teacher i
s asking you to focus on learning. Ack! You need a break! So do my student
s.Most of my students have autism, anxiety, another disability, or all of
the above. It is tough to focus in school due to sensory overload or emoti
ons. My students have a lot to deal with in school, but I think that makes
them the most incredible kids on the planet. They are kind, caring, and sy
mpathetic. They know what it's like to be overwhelmed, so they understand
when someone else is struggling. They are open-minded and compassionate. T
hey are the kids who will someday change the world.It is tough to do more
than one thing at a time. When sensory overload gets in the way, it is the
hardest thing in the world to focus on learning. My students need many bre
aks throughout the day, and one of the best items we've used is a Boogie B
oard. If we had a few in our own classroom, my students could take a break
exactly when they need one, regardless of which other rooms in the school
are occupied. Many of my students need to do something with their hands in
order to focus on the task at hand. Putty will give the sensory input they
need in order to focus, it will calm them when they are overloaded, it wil
l help improve motor skills, and it will make school more fun.When my stud
ents are able to calm themselves down, they are ready to learn. When they
are able to focus, they will learn more and retain more. They will get the
sensory input they need and it will prevent meltdowns (which are scary for
everyone in the room). This will lead to a better, happier classroom commu
nity that is able to learn the most they can in the best way possible.

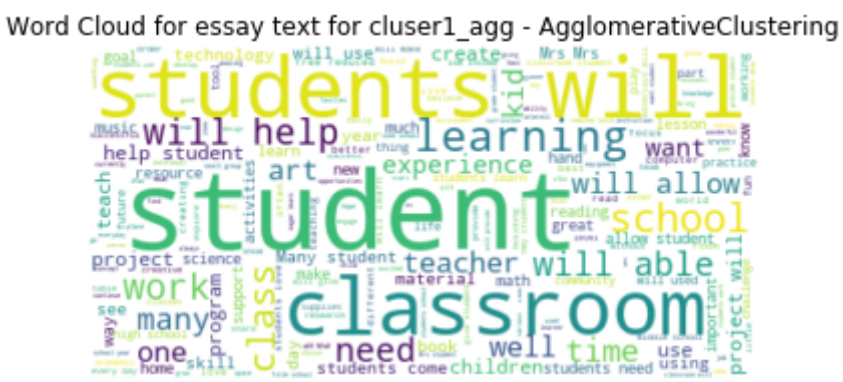
In [97]:

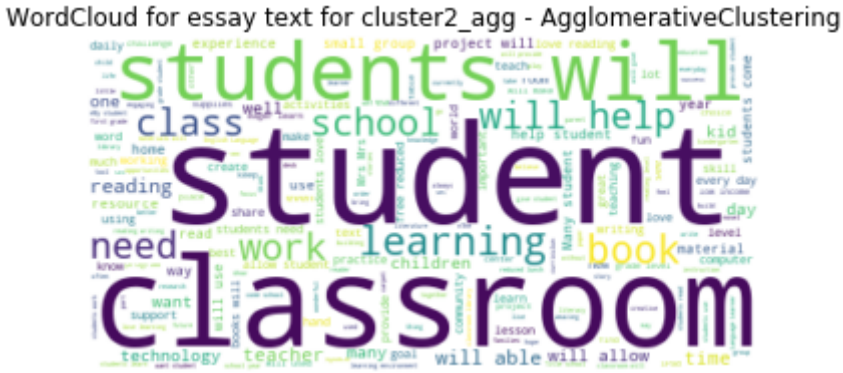
```
for i in range(2):
    print((cluster2_agg[i]))
```

```
Having a class of 24 students comes with diverse learners.  Some students
learn best through auditory means.I have a class of twenty-four kindergart
en students.\r\nMy students attend a Title 1 school and a great majority a
re English language learners. Most of our students come from low-income ho
mes, and all students receive free breakfast and lunch. My students are en
thusiastic learners, but too often are faced with many types of hardships
at home. School is often a safe haven for them.By having a mobile listenin
g and storage center, my students will be able to reinforce and enhance wh
at they are learning.  They will be able to listen to stories using the mo
bile listening center to help reinforce the high frequency words that have
been introduced.  In addition, they will be able to listen to stories that
reinforce reading comprehension skills and strategies amongst other audito
ry experiences.A mobile listening center will help keep equipment neat and
organized....ready to use to help reinforce and enhance literacy skills. N
umerous students will be able to use the center to help increase student l
earning.
My students crave challenge, they eat obstacles for breakfast! These new t
exts help me ensure I have materials to keep them challenged and thinking!
We are an urban, public k-5 elementary school. Our class is comprised of 1
2 girls and 16 boys. We incorporate hands on experiences to make learning
meaningful. My students are eager, curious and creative learners who have
a heart for social justice. They are a delight to teach!With the new commo
n core standards that have been adopted by our district, students need to
understand the author's craft of structure and analyze how the framework i
mpacts the readers interaction with the text and its characters. Both of t
hese texts are also read-alouds in our classroom. They are rich with inner
thinking so students can delve deep as they examine characters, their moti
ves and how they change over the course of a story.These remarkable gifts
will provide students with complex texts that take analytical skills to cu
ll and ponder. They would be an extravagant and remarkable gift that would
add depth to our library. Thank you for considering our classroom for your
donation!
```

In [99]:

```
#cluster1_agg
words=''
for i in cluster1_agg:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)
# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Word Cloud for essay text for cluser1_agg - AgglomerativeClustering")
plt.axis("off")
plt.show()
print("----------------------------------------------------")
print("----------------------------------------------------")
# cluster2_agg
words = ''
for i in cluster2_agg:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)
# Displaying the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("WordCloud for essay text for cluster2_agg - AgglomerativeClustering")
plt.axis("off")
plt.show()
```





2.4.2 Apply AgglometrativeClustering for K = 5

In [100]:

```
# n_cluster =5
aggm5 = AgglomerativeClustering(n_clusters=5).fit(X_new_best5k.toarray())
```

2.4.2.1 Ploting wordcloud with essay for 5 clusters - AgglomerativeClustering

In [102]:

```
cluster1_aggm5 = []
cluster2_aggm5 = []
cluster3_aggm5 = []
cluster4_aggm5 = []
cluster5_aggm5 = []
essays = project_data['essay'].values
for i in range(aggm5.labels_.shape[0]):
    if aggm5.labels_[i] == 0:
        cluster1_aggm5.append(essays[i])
    elif aggm5.labels_[i] == 1:
        cluster2_aggm5.append(essays[i])
    elif aggm5.labels_[i] == 2:
        cluster3_aggm5.append(essays[i])
    elif  aggm5.labels_[i] == 3:
        cluster4_aggm5.append(essays[i])
    elif  aggm5.labels_[i] == 4:
        cluster5_aggm5.append(essays[i])
```

In [109]:

```
for i in range(1):
    print("Cluster1 aggm5 :", (cluster1_aggm5[i]))
    print(120*"-")
for i in range(1):
    print("Cluster2 aggm5 :", (cluster2_aggm5[i]))
    print(120*"-")
for i in range(1):
    print("Cluster3 aggm5 :", (cluster3_aggm5[i]))
    print(120*"-")
for i in range(1):
    print("Cluster4 aggm5 :", (cluster4_aggm5[i]))
    print(120*"-")
for i in range(1):
    print("Cluster5 aggm5 :", (cluster5_aggm5[i]))
    print(120*"-")
```

```
Cluster1 aggm5 : Having a class of 24 students comes with diverse learner
s.  Some students learn best through auditory means.I have a class of twen
ty-four kindergarten students.\r\nMy students attend a Title 1 school and
a great majority are English language learners. Most of our students come
from low-income homes, and all students receive free breakfast and lunch.
My students are enthusiastic learners, but too often are faced with many t
ypes of hardships at home. School is often a safe haven for them.By having
a mobile listening and storage center, my students will be able to reinfor
ce and enhance what they are learning.  They will be able to listen to sto
ries using the mobile listening center to help reinforce the high frequenc
y words that have been introduced.  In addition, they will be able to list
en to stories that reinforce reading comprehension skills and strategies a
mongst other auditory experiences.A mobile listening center will help keep
equipment neat and organized....ready to use to help reinforce and enhance
literacy skills. Numerous students will be able to use the center to help
increase student learning.
------------------------------------------------------------------------
----------------------------------------------
Cluster2 aggm5 : I recently read an article about giving students a choice
about how they learn. We already set goals; why not let them choose where
to sit, and give them options of what to sit on?I teach at a low-income (T
itle 1) school. Every year, I have a class with a range of abilities, yet
they are all the same age. They learn differently, and they have different
interests. Some have ADHD, and some are fast learners. Yet they are eager
and active learners that want and need to be able to move around the room,
yet have a place that they can be comfortable to complete their work.We ne
ed a classroom rug that we can use as a class for reading time, and studen
ts can use during other learning times. I have also requested four Kore Ki
ds wobble chairs and four Back Jack padded portable chairs so that student
s can still move during whole group lessons without disrupting the class.
Having these areas will provide these little ones with a way to wiggle whi
le working.Benjamin Franklin once said, \"Tell me and I forget, teach me a
nd I may remember, involve me and I learn.\" I want these children to be i
nvolved in their learning by having a choice on where to sit and how to le
arn, all by giving them options for comfortable flexible seating.
------------------------------------------------------------------------
----------------------------------------------
Cluster3 aggm5 : Imagine being 8-9 years old. You're in your third grade c
lassroom. You see bright lights, the kid next to you is chewing gum, the b
irds are making noise, the street outside is buzzing with cars, it's hot,
and your teacher is asking you to focus on learning. Ack! You need a brea
k! So do my students.Most of my students have autism, anxiety, another dis
ability, or all of the above. It is tough to focus in school due to sensor
```

y overload or emotions. My students have a lot to deal with in school, but I think that makes them the most incredible kids on the planet. They are k ind, caring, and sympathetic. They know what it's like to be overwhelmed, so they understand when someone else is struggling. They are open-minded a nd compassionate. They are the kids who will someday change the world.It i s tough to do more than one thing at a time. When sensory overload gets in the way, it is the hardest thing in the world to focus on learning. My stu dents need many breaks throughout the day, and one of the best items we've used is a Boogie Board. If we had a few in our own classroom, my students could take a break exactly when they need one, regardless of which other r ooms in the school are occupied. Many of my students need to do something with their hands in order to focus on the task at hand. Putty will give th e sensory input they need in order to focus, it will calm them when they a re overloaded, it will help improve motor skills, and it will make school more fun.When my students are able to calm themselves down, they are ready to learn. When they are able to focus, they will learn more and retain mor e. They will get the sensory input they need and it will prevent meltdowns (which are scary for everyone in the room). This will lead to a better, ha ppier classroom community that is able to learn the most they can in the b est way possible.

Cluster4 aggm5 : Media and cinematography has been an extremely fast growi ng subject area since the rise in technology. Students are captivated by s ocial media outlets such as Twitter and Instagram. At Tri-Valley, I am the teacher of a media class where we aim to create meaningful videos to promo te our great school.My students in media are mostly junior and senior stud ents who are extremely creative. Their imaginations are on display when we sit and think of ideas for our next video or commercial to show to the sch ool. It is a joy to watch them light up when they see the video that they made playing in the cafeteria for 400 students to watch. The students' ima ginations often times are bigger than the equipment we have and I feel bad telling them that we cannot get certain shots or footage because our equip ment is not up to par. The students love using the editing equipment that the school provides but they do not like the old cameras. I want to be abl e to let their creative juices flow and have equipment that can keep up.Th e materials that I am requesting will be used in a multitude of ways. Firs t, the Phantom Video drone will be used to make a commercial for the schoo l to put on the school website. The footage that a drone can capture will be the perfect intro to our video about what a great school and little tow n we have. The drone will also be used to get unique footage at various sp orting events. Imagine the drone footage of the game winning touchdown, or the game winning home run in a softball game, or the last leg of the 4x100 meter relay team. The footage captured with the drone will be uploaded, ed iting, and put onto the school and athletic websites to promote the cultur e at Tri-Valley. The GoPro camera will be used much in the same way. We wi ll have students assigned to different sporting events and they will captu re footage with the GoPro. We will then upload that footage and merge it w ith footage from the drone to complete the videos that will be uploaded on the school and athletic website.My project will make a difference. My proj ect will give the students something to look forward to every single day. My project will show the community what great young people we have in the district. The donations would be greatly appreciated by myself and my stud ents so they can showcase their creativity to the entire school. I want to have a classroom where students are learning by inquiry and solving proble ms on their own. A generous donation would allow me to capture the student s attention.

Cluster5 aggm5 : I have been fortunate enough to use the Fairy Tale STEM k its in my classroom as well as the STEM journals, which my students really

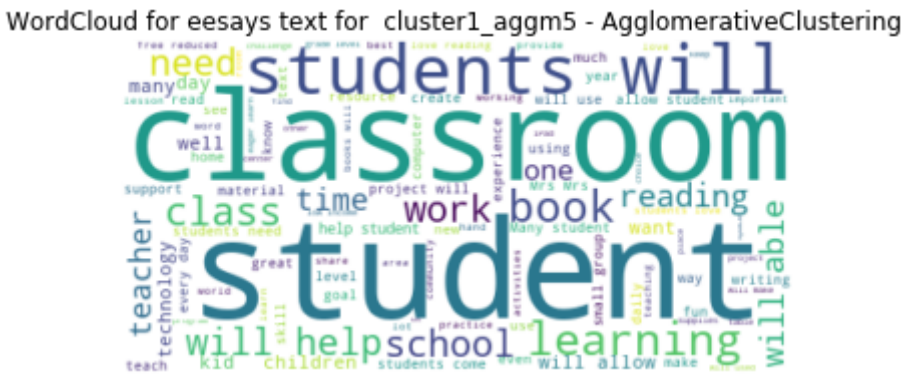
enjoyed. I would love to implement more of the Lakeshore STEM kits in my classroom for the next school year as they provide excellent and engaging STEM lessons.My students come from a variety of backgrounds, including lan guage and socioeconomic status. Many of them don't have a lot of experien ce in science and engineering and these kits give me the materials to prov ide these exciting opportunities for my students.Each month I try to do se veral science or STEM/STEAM projects. I would use the kits and robot to h elp guide my science instruction in engaging and meaningful ways. I can a dapt the kits to my current language arts pacing guide where we already te ach some of the material in the kits like tall tales (Paul Bunyan) or John ny Appleseed. The following units will be taught in the next school year where I will implement these kits: magnets, motion, sink vs. float, robot s. I often get to these units and don't know If I am teaching the right w ay or using the right materials.   The kits will give me additional idea s, strategies, and lessons to prepare my students in science.It is challen ging to develop high quality science activities. These kits give me the m aterials I need to provide my students with science activities that will g o along with the curriculum in my classroom. Although I have some things (like magnets) in my classroom, I don't know how to use them effectively. The kits will provide me with the right amount of materials and show me ho w to use them in an appropriate way.

2.4.2.2 Ploting wordcloud with essay for cluster1 - AgglomerativeClustering

In [110]:

```
# cluster1 aggm5 wordcloud
words=''
for i in cluster1_aggm5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("WordCloud for eesays text for  cluster1_aggm5 - AgglomerativeClustering ")
plt.axis("off")
plt.show()
```



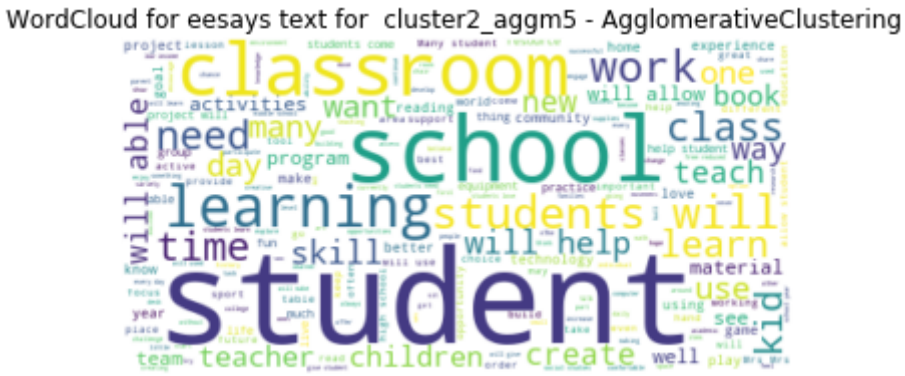
2.4.2.3 Ploting wordcloud with essay for cluster2 and cluster3 - AgglomerativeClustering

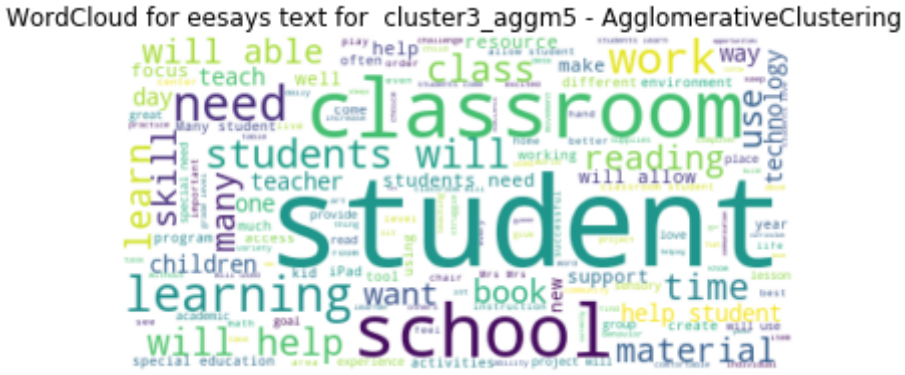
In [111]:

```
# cluster2_aggm5 wordcloud
words=''
for i in cluster2_aggm5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("WordCloud for eesays text for  cluster2_aggm5 - AgglomerativeClustering ")
plt.axis("off")
plt.show()
print("------------------------------------------------------------------")
# cluster3_aggm5 wordcloud
words=''
for i in cluster3_aggm5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("WordCloud for eesays text for  cluster3_aggm5 - AgglomerativeClustering ")
plt.axis("off")
plt.show()
```





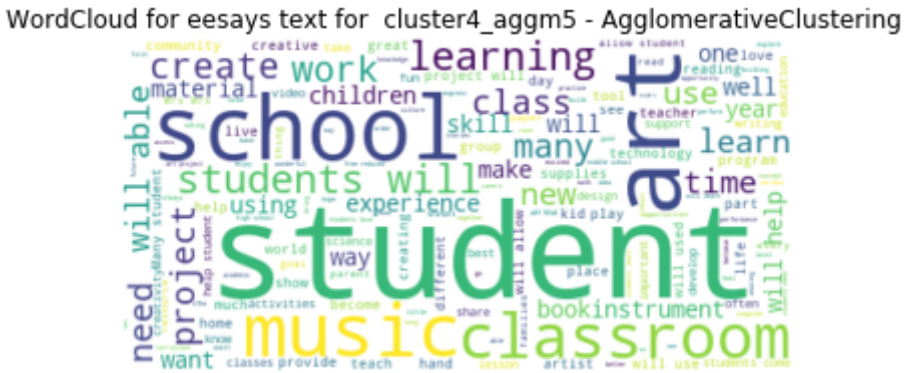
2.4.2.4 Ploting wordcloud with essay for cluster4 and cluster5 - AgglomerativeClustering

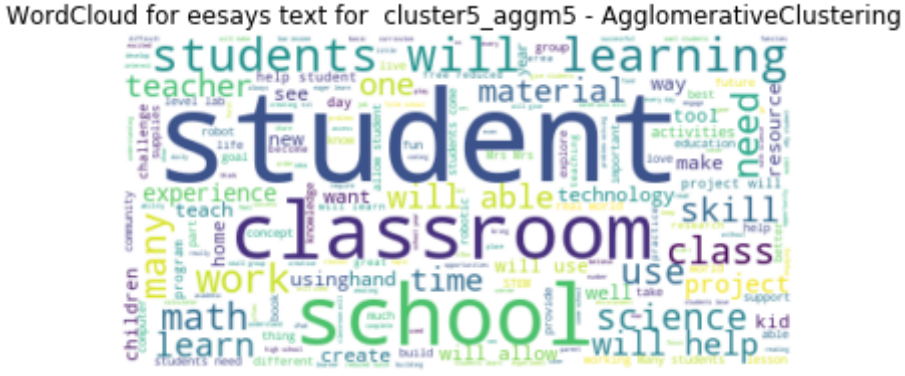
In [112]:

```
# cluster4_aggm5 wordcloud
words=''
for i in cluster4_aggm5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("WordCloud for eesays text for  cluster4_aggm5 - AgglomerativeClustering ")
plt.axis("off")
plt.show()
print("------------------------------------------------------------")
# cluster5_aggm5 wordcloud
words=''
for i in cluster5_aggm5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("WordCloud for eesays text for  cluster5_aggm5 - AgglomerativeClustering ")
plt.axis("off")
plt.show()
```





2.5 Apply DBSCAN

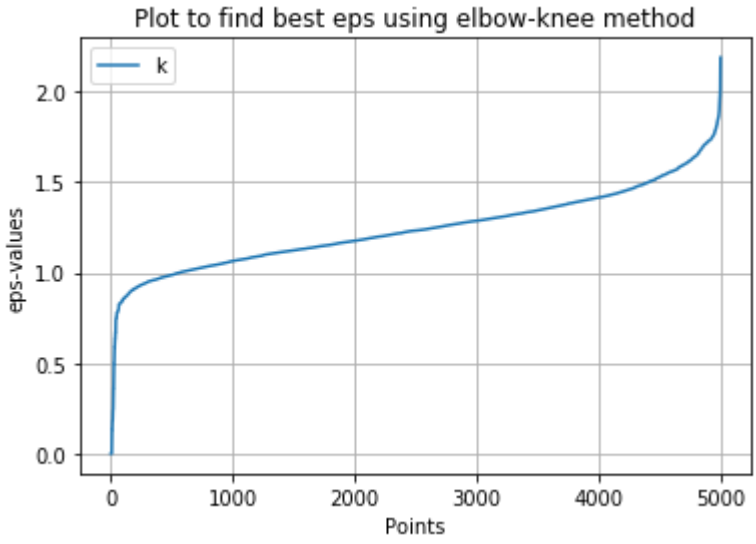
In [125]:

```
#Cite:https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-o
# Finding the best eps using elbow-knee method
# we can calculate the distance from each point to its closest neighbour using the Near
from sklearn.neighbors import NearestNeighbors
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(X_new_best5k)
distances, indices = nbrs.kneighbors(X_new_best5k)
```

2.5.1 Finding the best eps using elbow-knee method

In [126]:

```
# we sort and plot results.
distances = np.sort(distances, axis=0)
distances = distances[:,1]
plt.plot(distances)
plt.title("Plot to find best eps using elbow-knee method")
plt.xlabel('Points')
plt.ylabel('eps-values')
plt.legend('kneee')
plt.grid(True)
plt.show()
```



we can see that point of inflexion is at eps=0.9

In [127]:

```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.9, n_jobs=-1).fit(X_new_best5k)
# The labels_ property contains the list of clusters and their respective points.
print('No of clusters: ',len(set(dbscan.labels_)))
print('Cluster are including noise i.e -1: ',set(dbscan.labels_))
```

```
No of clusters:  6
Cluster are including noise i.e -1:  {0, 1, 2, 3, 4, -1}
```

2.5.2 Ploting wordcloud with essay for clusters - DBSCAN

In [158]:

```
#ignoring -1 as it is for noise
cluster1_db=[]
noisecluster1=[]
cluster2_db=[]
for i in range(dbscan.labels_.shape[0]):
    if dbscan.labels_[i] == 0:
        cluster1_db.append(essays[i])
    elif dbscan.labels_[i] == -1:
        noisecluster1.append(essays[i])
    elif dbscan.labels_[i] == 1:
        cluster2_db.append(essays[i])
```

In [149]:

```
for i in range(1):
    print("Cluster1_db :", cluster1_db[i])
    print(120*"-")
for i in range(1):
    print("noisecluster1:", noisecluster1[i])
```

```
Cluster1_db : I have a classroom full of 27 unique, inquiring minded first
graders that wonder and ask questions constantly. My students love to rese
arch topics in all areas. They support each other as if they are a family
and are continually going further as a class and as individuals.My class i
s filled with students with individual strengths and challenges. My studen
ts are at many different reading and math levels. As a class, we celebrate
our diverse abilities and use each other’s talents to help us grow.  Altho
ugh students are at various levels, every single one of them loves to pick
up books and read or challenge themselves with a math talk.  My students e
njoy reading and math on their own, with a partner and in small groups. Th
ey love to share what they learn with each other as they inquire about the
knowledge they are gaining and the strategies they are using.  \r\n\r\nOur
school is located in a suburb of Phoenix. We serve free breakfast in our c
lassroom every day and are 100% free lunch.  I feel my students are not ex
posed to technology at home and these resources will make a difference all
owing student to anchor their learning and achieve the things I am so conf
ident they are capable of.By having Ipad tablets in my classroom, my stude
nts' eyes would be open to the world beyond Sunset Elementary. Students wi
ll be able to explore their many wonderings through inquiry based learnin
g, as well as receive differentiated instruction to meet their individual
needs, and connect with others around the world. These materials will allo
w me to conduct technology rich lessons that drive student learning by cat
ering to my students' individual interests. My class is made up of a varie
ty of learners with a large rang of abilities and these materials will all
ow for all of my students' learning needs to be meet. My students will use
the Ipad Mini's to enrich learning across all content areas.It is my job,
as a teacher, to teach my students to have a love for learning. The Ipad m
ini's are the missing key that will allow student additional opportunities
to discover the joy of learning through technology. \r\nIn my classroom, s
tudents will use technology to drive student centered learning by using th
e internet as a resource to research and extend their learning. Students w
ill also use apps and teacher created screen casts designed to fill learni
ng gaps and individualize instruction.\r\n
--------------------------------------------------------------------------
----------------------------------------------
noisecluster1: I have been fortunate enough to use the Fairy Tale STEM kit
s in my classroom as well as the STEM journals, which my students really e
njoyed.  I would love to implement more of the Lakeshore STEM kits in my c
lassroom for the next school year as they provide excellent and engaging S
TEM lessons.My students come from a variety of backgrounds, including lang
uage and socioeconomic status.  Many of them don't have a lot of experienc
e in science and engineering and these kits give me the materials to provi
de these exciting opportunities for my students.Each month I try to do sev
eral science or STEM/STEAM projects.  I would use the kits and robot to he
lp guide my science instruction in engaging and meaningful ways.  I can ad
apt the kits to my current language arts pacing guide where we already tea
ch some of the material in the kits like tall tales (Paul Bunyan) or Johnn
y Appleseed.  The following units will be taught in the next school year w
here I will implement these kits: magnets, motion, sink vs. float, robots.
I often get to these units and don't know If I am teaching the right way o
r using the right materials.    The kits will give me additional ideas, st
rategies, and lessons to prepare my students in science.It is challenging
to develop high quality science activities.  These kits give me the materi
```

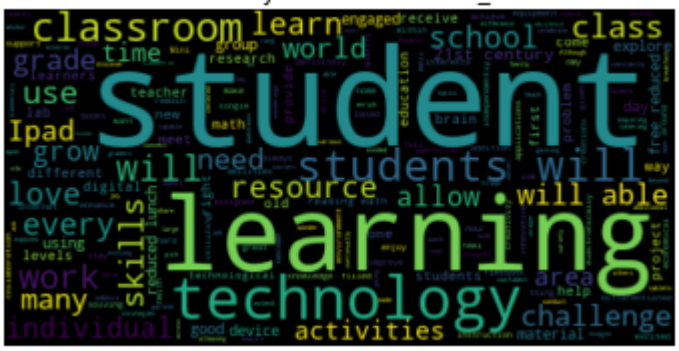
```
als I need to provide my students with science activities that will go alo
ng with the curriculum in my classroom.  Although I have some things (like
magnets) in my classroom, I don't know how to use them effectively.  The k
its will provide me with the right amount of materials and show me how to
use them in an appropriate way.
```

2.5.2.1 Ploting wordcloud with essay for cluster1 and noisecluster1 - DBSCAN

In [154]:

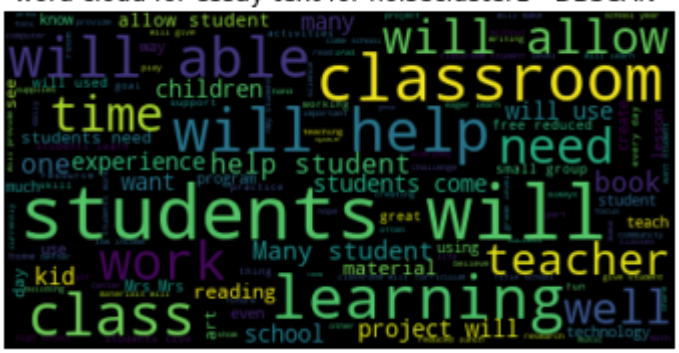
```
# cluster1_db wordcloud
words = ''
for i in cluster1_db:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud().generate(words)
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("word cloud for essay text for cluster1_db - DBSCAN ")
plt.axis("off")
plt.show()
print("----------------------------------------")
# noisecluster1
words =''
for i in noisecluster1:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud().generate(words)
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("word cloud for essay text for noisecluster1 - DBSCAN ")
plt.axis("off")
plt.show()
```

word cloud for essay text for cluster1_db - DBSCAN



---

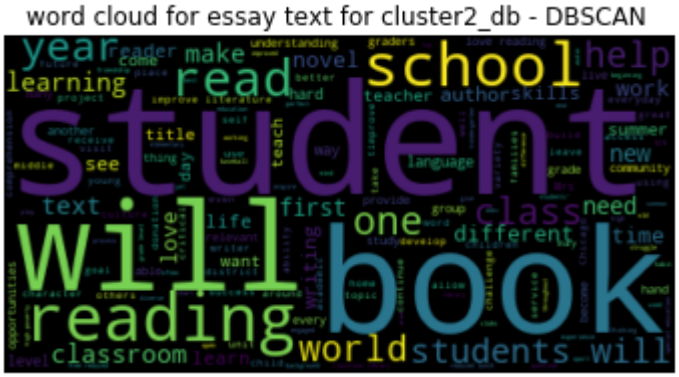
word cloud for essay text for noisecluster1 - DBSCAN



2.5.2.2 Ploting wordcloud with essay for cluster2 - DBSCAN

In [159]:

```
# cluster2_db wordcloud
words = ''
for i in cluster2_db:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud().generate(words)
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("word cloud for essay text for cluster2_db - DBSCAN ")
plt.axis("off")
plt.show()
print("----------------------------------------------")
```



---

3. Cocnlusions

In [4]:

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

p = PrettyTable()
p.field_names = ["Vectorizer","Best k", "Algorithm"]
p.add_row(['TFIDF','6', 'KMeans'])
p.add_row(['TFIDF', '2', 'AgglomerativeClustering'])
p.add_row(['TFIDF', '5', 'AgglomerativeClustering'])
p.add_row(['TFIDF', '6', 'DBSCAN'])
print(p)
print("*****************************************")
p1 = PrettyTable()
p1.field_names = ["Vectorizer", "Best K", "EPS", "No's NoiseClusters", "Algorithm"]
p1.add_row(['TFIDF', '6', 0.9, 1, 'DBSCAN'])
print(p1)
```

```
+------------+--------+-------------------------+
| Vectorizer | Best k |        Algorithm        |
+------------+--------+-------------------------+
|   TFIDF    |   6    |          KMeans         |
|   TFIDF    |   2    | AgglomerativeClustering |
|   TFIDF    |   5    | AgglomerativeClustering |
|   TFIDF    |   6    |          DBSCAN         |
+------------+--------+-------------------------+
*****************************************
+------------+--------+-----+--------------------+-----------+
| Vectorizer | Best K | EPS | No's NoiseClusters | Algorithm |
+------------+--------+-----+--------------------+-----------+
|   TFIDF    |   6    | 0.9 |         1          |   DBSCAN  |
+------------+--------+-----+--------------------+-----------+
```

Thank You.

Sign Off RAMESH BATTU (https://www.linkedin.com/in/rameshbattuai/)