DonorsChoose

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

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

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

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

About the DonorsChoose Data Set

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

Feature	Description
`project_id`	A unique identifier for the proposed project. **Example:** `p036502`
`project_title`	Title of the project. **Examples:** • `Art Will Make You Happy!` • `First Grade Fun`
`project_grade_category`	Grade level of students for which the project is targeted. One of the following enumerated values: • `Grades PreK-2` • `Grades 3-5` • `Grades 6-8` • `Grades 9-12`
`project_subject_categories`	One or more (comma-separated) subject categories for the project from the following enumerated list of values: • `Applied Learning` • `Care & Hunger` • `Health & Sports` • `History & Civics` • `Literacy & Language` • `Math & Science` • `Music & The Arts` • `Special Needs` • `Warmth` **Examples:** • `Music & The Arts` • `Literacy & Language, Math & Science`

`school_state`	State where school is located (Two-letter U.S. postal code). **Example:** `WY`
`project_subject_subcategories`	One or more (comma-separated) subject subcategories for the project. **Examples:** • `Literacy` • `Literature & Writing, Social Sciences`
`project_resource_summary`	An explanation of the resources needed for the project. **Example:** • `My students need hands on literacy materials to manage sensory needs!`
`project_essay_1`	First application essay*
`project_essay_2`	Second application essay*
`project_essay_3`	Third application essay*
`project_essay_4`	Fourth application essay*
`project_submitted_datetime`	Datetime when project application was submitted. **Example:** `2016-04-28 12:43:56.245`
`teacher_id`	A unique identifier for the teacher of the proposed project. **Example:** `bdf8baa8fedef6bfeec7ae4ff1c15c56`
`teacher_prefix`	Teacher's title. One of the following enumerated values: • `nan` • `Dr.` • `Mr.` • `Mrs.` • `Ms.` • `Teacher.`
`teacher_number_of_previously_posted_projects`	Number of project applications previously submitted by the same teacher. **Example:** `2`

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

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

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

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

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

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

Notes on the Essay Data

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

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

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

- __project_essay_1:__ "Describe your students: What makes

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

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

In [1]:

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

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

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

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

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

from tqdm import tqdm
import os

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

```
C:\Users\Public\Anaconda3\lib\site-packages\ge
nsim\utils.py:1197: UserWarning: detected Wind
ows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing ch
unkize to chunkize_serial")
```

1.1 Reading Data

In [2]:

```
project_data = pd.read_csv('C:/Users/pramod reddy chandi/Desk
top/pram/applied ai course/DonorsChoose_2018/train_data.csv')
resource_data = pd.read_csv('C:/Users/pramod reddy chandi/Des
ktop/pram/applied ai course/DonorsChoose_2018/resources.csv')
```

In [3]:

```
print("Number of data points in train data", project_data.sha
pe)
print('-'*50)
print("The attributes of data :", project_data.columns.values
)
```

```
Number of data points in train data (109248, 1
7)
---------------------------------------------
----
The attributes of data : ['Unnamed: 0' 'id' 't
eacher_id' 'teacher_prefix' 'school_state'
 'project_submitted_datetime' 'project_grade_c
ategory'
 'project_subject_categories' 'project_subject
_subcategories'
 'project_title' 'project_essay_1' 'project_es
say_2' 'project_essay_3'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects
' 'project_is_approved']
```

In [4]:

```
print("Number of data points in train data", resource_data.sh
ape)
print(resource_data.columns.values)
resource_data.head(2)
```

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

Out[4]:

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

In [5]:

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


#sort dataframe based on time pandas python: https://stackove
rflow.com/a/49702492/4084039
project_data['Date'] = pd.to_datetime(project_data['project_s
ubmitted_datetime'])
project_data.drop('project_submitted_datetime', axis=1, inpla
ce=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[5]:

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

In [6]:

```
print("Number of data points in train data", resource_data.sh
ape)
print(resource_data.columns.values)
resource_data.head(2)
```

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

Out[6]:

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

1.2 preprocessing of `project_subject_categories`

In [7]:

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

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

```
    cat_list.append(temp.strip())

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

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

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

1.3 preprocessing of `project_subject_subcategories`

In [8]:

```
sub_catogories = list(project_data['project_subject_subcatego
ries'].values)
# remove special characters from list of strings python: http
s://stackoverflow.com/a/47301924/4084039

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

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

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

# count of all the words in corpus python: https://stackoverf
low.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=l
ambda kv: kv[1]))
```

1.3 Text preprocessing

In [9]:

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

In [10]:

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

Out[10]:

	Unnamed: 0	id	teach
55660	8393	p205479	2bf07ba08945e5d8b2a3f269b2b
76127	37728	p043609	3f60494c61921b3b43ab61bdde2

In [11]:

```python
# printing some random reviews
print(project_data['essay'].values[0])
print("="*50)
print(project_data['essay'].values[150])
print("="*50)
print(project_data['essay'].values[1000])
print("="*50)
print(project_data['essay'].values[20000])
print("="*50)
print(project_data['essay'].values[99999])
print("="*50)
```

```
I have been fortunate enough to use the Fairy
Tale STEM kits in my classroom as well as the
STEM journals, which my students really enjoye
d.  I would love to implement more of the Lake
shore STEM kits in my classroom for the next s
chool year as they provide excellent and engag
ing STEM lessons.My students come from a varie
ty of backgrounds, including language and soci
oeconomic status.  Many of them don't have a l
ot of experience in science and engineering an
d these kits give me the materials to provide
these exciting opportunities for my students.E
ach month I try to do several science or STEM/
STEAM projects.  I would use the kits and robo
t to help guide my science instruction in enga
ging and meaningful ways.  I can adapt the kit
s to my current language arts pacing guide whe
re we already teach some of the material in th
e kits like tall tales (Paul Bunyan) or Johnny
 Appleseed.  The following units will be taugh
t in the next school year where I will impleme
nt 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 right materials.    The kits will give me
```

 additional ideas, strategies, and lessons to
prepare my students in science.It is challengi
ng to develop high quality science activities.
  These kits give me the materials I need to p
rovide my students with science activities tha
t will go along with the curriculum in my clas
sroom. Although I have some things (like magn
ets) in my classroom, I don't know how to use
them effectively. The kits will provide me wi
th the right amount of materials and show me h
ow to use them in an appropriate way.
==
====
I teach high school English to students with l
earning and behavioral disabilities. My studen
ts all vary in their ability level. However, t
he ultimate goal is to increase all students l
iteracy levels. This includes their reading, w
riting, and communication levels.I teach a rea
lly dynamic group of students. However, my stu
dents face a lot of challenges. My students al
l live in poverty and in a dangerous neighborh
ood. Despite these challenges, I have students
 who have the the desire to defeat these chall
enges. My students all have learning disabilit
ies and currently all are performing below gra
de level. My students are visual learners and
will benefit from a classroom that fulfills th
eir preferred learning style.The materials I a
m requesting will allow my students to be prep
ared for the classroom with the necessary supp
lies. Too often I am challenged with students
 who come to school unprepared for class due t
o economic challenges. I want my students to
be able to focus on learning and not how they
will be able to get school supplies. The supp
lies will last all year. Students will be abl

e to complete written assignments and maintain
 a classroom journal. The chart paper will be
 used to make learning more visual in class an
d to create posters to aid students in their l
earning. The students have access to a classr
oom printer. The toner will be used to print
student work that is completed on the classroo
m Chromebooks.I want to try and remove all bar
riers for the students learning and create opp
ortunities for learning. One of the biggest ba
rriers is the students not having the resource
s to get pens, paper, and folders. My students
 will be able to increase their literacy skill
s because of this project.
==
====
\"Life moves pretty fast. If you don't stop an
d look around once in awhile, you could miss i
t.\" from the movie, Ferris Bueller's Day Off
. Think back...what do you remember about you
r grandparents? How amazing would it be to be
 able to flip through a book to see a day in t
heir lives?My second graders are voracious rea
ders! They love to read both fiction and nonfi
ction books. Their favorite characters includ
e Pete the Cat, Fly Guy, Piggie and Elephant,
and Mercy Watson. They also love to read about
 insects, space and plants. My students are hu
ngry bookworms! My students are eager to learn
 and read about the world around them. My kids
 love to be at school and are like little spon
ges absorbing everything around them. Their pa
rents work long hours and usually do not see t
heir children. My students are usually cared f
or by their grandparents or a family friend. M
ost of my students do not have someone who spe
aks English at home. Thus it is difficult for

```
my students to acquire language.Now think forw
ard... wouldn't it mean a lot to your kids, ni
eces or nephews or grandchildren, to be able t
o see a day in your life today 30 years from n
ow? Memories are so precious to us and being a
ble to share these memories with future genera
tions will be a rewarding experience.  As part
 of our social studies curriculum, students wi
ll be learning about changes over time.  Stude
nts will be studying photos to learn about how
 their community has changed over time.  In pa
rticular, we will look at photos to study how
the land, buildings, clothing, and schools hav
e changed over time.  As a culminating activit
y, my students will capture a slice of their h
istory and preserve it through scrap booking.
Key important events in their young lives will
 be documented with the date, location, and na
mes.   Students will be using photos from home
 and from school to create their second grade
memories.   Their scrap books will preserve th
eir unique stories for future generations to e
njoy.Your donation to this project will provid
e my second graders with an opportunity to lea
rn about social studies in a fun and creative
manner.  Through their scrapbooks, children wi
ll share their story with others and have a hi
storical document for the rest of their lives.
==============================================
====
\"A person's a person, no matter how small.\"
(Dr.Seuss) I teach the smallest students with
the biggest enthusiasm for learning. My studen
ts learn in many different ways using all of o
ur senses and multiple intelligences. I use a
wide range of techniques to help all my studen
ts succeed. \r\nStudents in my class come from
```

```
 a variety of different backgrounds which make
s for wonderful sharing of experiences and cul
tures, including Native Americans.\r\nOur scho
ol is a caring community of successful learner
s which can be seen through collaborative stud
ent project based learning in and out of the c
lassroom. Kindergarteners in my class love to 
work with hands-on materials and have many dif
ferent opportunities to practice a skill befor
e it is mastered. Having the social skills to 
work cooperatively with friends is a crucial a
spect of the kindergarten curriculum.Montana i
s the perfect place to learn about agriculture
 and nutrition. My students love to role play 
in our pretend kitchen in the early childhood 
classroom. I have had several kids ask me, \"C
an we try cooking with REAL food?\" I will tak
e their idea and create \"Common Core Cooking 
Lessons\" where we learn important math and wr
iting concepts while cooking delicious healthy
 food for snack time. My students will have a 
grounded appreciation for the work that went i
nto making the food and knowledge of where the
 ingredients came from as well as how it's hea
lthy for their bodies. This project would expa
nd our learning of nutrition and agricultural 
cooking recipes by having us peel our own appl
es to make homemade applesauce, make our own b
read, and mix up healthy plants from our class
room garden in the spring. We will also create
 our own cookbooks to be printed and shared wi
th families. \r\nStudents will gain math and l
iterature skills as well as a life long enjoym
ent for healthy cooking.nannan
==============================================
====
My classroom consists of twenty-two amazing si
```

xth graders from different cultures and backgr
ounds. They are a social bunch who enjoy worki
ng in partners and working with groups. They a
re hard-working and eager to head to middle sc
hool next year. My job is to get them ready to
 make this transition and make it as smooth as
 possible. In order to do this, my students ne
ed to come to school every day and feel safe a
nd ready to learn. Because they are getting re
ady to head to middle school, I give them lots
 of choice- choice on where to sit and work, t
he order to complete assignments, choice of pr
ojects, etc. Part of the students feeling safe
 is the ability for them to come into a welcom
ing, encouraging environment. My room is color
ful and the atmosphere is casual. I want them
to take ownership of the classroom because we
ALL share it together. Because my time with th
em is limited, I want to ensure they get the m
ost of this time and enjoy it to the best of t
heir abilities.Currently, we have twenty-two d
esks of differing sizes, yet the desks are sim
ilar to the ones the students will use in midd
le school. We also have a kidney table with cr
ates for seating. I allow my students to choos
e their own spots while they are working indep
endently or in groups. More often than not, mo
st of them move out of their desks and onto th
e crates. Believe it or not, this has proven t
o be more successful than making them stay at
their desks! It is because of this that I am l
ooking toward the “Flexible Seating” option fo
r my classroom.\r\n The students look forward
to their work time so they can move around the
 room. I would like to get rid of the constric
ting desks and move toward more “fun” seating
options. I am requesting various seating so my

```
 students have more options to sit. Currently,
 I have a stool and a papasan chair I inherite
d from the previous sixth-grade teacher as wel
l as five milk crate seats I made, but I would
 like to give them more options and reduce the
 competition for the “good seats”. I am also r
equesting two rugs as not only more seating op
tions but to make the classroom more welcoming
 and appealing. In order for my students to be
 able to write and complete work without desks
, I am requesting a class set of clipboards. F
inally, due to curriculum that requires groups
 to work together, I am requesting tables that
 we can fold up when we are not using them to
leave more room for our flexible seating optio
ns.\r\nI know that with more seating options,
they will be that much more excited about comi
ng to school! Thank you for your support in ma
king my classroom one students will remember f
orever!nannan
==============================================
====
```

In [12]:

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

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

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

In [13]:

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

```
\"A person is a person, no matter how small.\"
 (Dr.Seuss) I teach the smallest students with
 the biggest enthusiasm for learning. My stude
nts learn in many different ways using all of
our senses and multiple intelligences. I use a
 wide range of techniques to help all my stude
nts succeed. \r\nStudents in my class come fro
m a variety of different backgrounds which mak
es for wonderful sharing of experiences and cu
ltures, including Native Americans.\r\nOur sch
ool is a caring community of successful learne
rs which can be seen through collaborative stu
dent project based learning in and out of the
classroom. Kindergarteners in my class love to
 work with hands-on materials and have many di
fferent opportunities to practice a skill befo
re it is mastered. Having the social skills to
 work cooperatively with friends is a crucial
aspect of the kindergarten curriculum.Montana
is the perfect place to learn about agricultur
e and nutrition. My students love to role play
 in our pretend kitchen in the early childhood
 classroom. I have had several kids ask me, \"
Can we try cooking with REAL food?\" I will ta
ke their idea and create \"Common Core Cooking
```

```
 Lessons\" where we learn important math and w
riting concepts while cooking delicious health
y food for snack time. My students will have a
 grounded appreciation for the work that went 
into making the food and knowledge of where th
e ingredients came from as well as how it is h
ealthy for their bodies. This project would ex
pand our learning of nutrition and agricultura
l cooking recipes by having us peel our own ap
ples to make homemade applesauce, make our own
 bread, and mix up healthy plants from our cla
ssroom garden in the spring. We will also crea
te our own cookbooks to be printed and shared 
with families. \r\nStudents will gain math and
 literature skills as well as a life long enjo
yment for healthy cooking.nannan
==============================================
====
```

In [14]:

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

```
 A person is a person, no matter how small.  (
Dr.Seuss) I teach the smallest students with t
he biggest enthusiasm for learning. My student
s learn in many different ways using all of ou
r senses and multiple intelligences. I use a w
ide range of techniques to help all my student
s succeed.   Students in my class come from a 
variety of different backgrounds which makes f
or wonderful sharing of experiences and cultur
es, including Native Americans.  Our school is
```

```
 a caring community of successful learners whi
ch can be seen through collaborative student p
roject based learning in and out of the classr
oom. Kindergarteners in my class love to work
with hands-on materials and have many differen
t opportunities to practice a skill before it
is mastered. Having the social skills to work
cooperatively with friends is a crucial aspect
 of the kindergarten curriculum.Montana is the
 perfect place to learn about agriculture and
nutrition. My students love to role play in ou
r pretend kitchen in the early childhood class
room. I have had several kids ask me,  Can we
try cooking with REAL food?  I will take their
 idea and create  Common Core Cooking Lessons
 where we learn important math and writing con
cepts while cooking delicious healthy food for
 snack time. My students will have a grounded
appreciation for the work that went into makin
g the food and knowledge of where the ingredie
nts came from as well as how it is healthy for
 their bodies. This project would expand our l
earning of nutrition and agricultural cooking
recipes by having us peel our own apples to ma
ke homemade applesauce, make our own bread, an
d mix up healthy plants from our classroom gar
den in the spring. We will also create our own
 cookbooks to be printed and shared with famil
ies.   Students will gain math and literature
skills as well as a life long enjoyment for he
althy cooking.nannan
```

In [15]:

```
#remove spacial character: https://stackoverflow.com/a/584354
7/4084039
sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
```

```python
print(sent)
```

```
 A person is a person no matter how small Dr S
euss I teach the smallest students with the bi
ggest enthusiasm for learning My students lear
n in many different ways using all of our sens
es and multiple intelligences I use a wide ran
ge of techniques to help all my students succe
ed Students in my class come from a variety of
 different backgrounds which makes for wonderf
ul sharing of experiences and cultures includi
ng Native Americans Our school is a caring com
munity of successful learners which can be see
n through collaborative student project based 
learning in and out of the classroom Kindergar
teners in my class love to work with hands on 
materials and have many different opportunitie
s to practice a skill before it is mastered Ha
ving the social skills to work cooperatively w
ith friends is a crucial aspect of the kinderg
arten curriculum Montana is the perfect place 
to learn about agriculture and nutrition My st
udents love to role play in our pretend kitche
n in the early childhood classroom I have had 
several kids ask me Can we try cooking with RE
AL food I will take their idea and create Comm
on Core Cooking Lessons where we learn importa
nt math and writing concepts while cooking del
icious healthy food for snack time My students
 will have a grounded appreciation for the wor
k that went into making the food and knowledge
 of where the ingredients came from as well as
 how it is healthy for their bodies This proje
ct would expand our learning of nutrition and 
agricultural cooking recipes by having us peel
 our own apples to make homemade applesauce ma
ke our own bread and mix up healthy plants fro
```

```
m our classroom garden in the spring We will a
lso create our own cookbooks to be printed and
 shared with families Students will gain math
and literature skills as well as a life long e
njoyment for healthy cooking nannan
```

In [16]:

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

In [17]:

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

```
100%|██████████| 109248/109248 [00:58<00:00, 1
875.60it/s]
```

In [18]:

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

Out[18]:

```
'a person person no matter small dr seuss i te
ach smallest students biggest enthusiasm learn
ing my students learn many different ways usin
g senses multiple intelligences i use wide ran
ge techniques help students succeed students c
lass come variety different backgrounds makes
wonderful sharing experiences cultures includi
```

```
ng native americans our school caring communit
y successful learners seen collaborative stude
nt project based learning classroom kindergart
eners class love work hands materials many dif
ferent opportunities practice skill mastered h
aving social skills work cooperatively friends
 crucial aspect kindergarten curriculum montan
a perfect place learn agriculture nutrition my
 students love role play pretend kitchen early
 childhood classroom i several kids ask can tr
y cooking real food i take idea create common 
core cooking lessons learn important math writ
ing concepts cooking delicious healthy food sn
ack time my students grounded appreciation wor
k went making food knowledge ingredients came 
well healthy bodies this project would expand 
learning nutrition agricultural cooking recipe
s us peel apples make homemade applesauce make
 bread mix healthy plants classroom garden spr
ing we also create cookbooks printed shared fa
milies students gain math literature skills we
ll life long enjoyment healthy cooking nannan'
```

In [19]:

```
#Project essay word count

essay_word_count = []

for ess in project_data["essay"] :
    c = len(ess.split())
    essay_word_count.append(c)

project_data["essay_word_count"] = essay_word_count
```

In [20]:

```
project_data['preprocessed_essays'] = preprocessed_essays
```

In [21]:

```python
import nltk

from nltk.sentiment.vader import SentimentIntensityAnalyzer

analyser = SentimentIntensityAnalyzer()

pos =[]
neg = []
neu = []
compound = []

for a in tqdm(project_data["preprocessed_essays"]) :
    b = analyser.polarity_scores(a)['neg']
    c = analyser.polarity_scores(a)['pos']
    d = analyser.polarity_scores(a)['neu']
    e = analyser.polarity_scores(a)['compound']
    neg.append(b)
    pos.append(c)
    neu.append(d)
    compound.append(e)
```

```
100%|██████████| 109248/109248 [12:36<00:00, 1
44.50it/s]
```

In [22]:

```python
project_data["pos"] = pos
project_data["neg"] = neg
project_data["neu"] = neu
project_data["compound"] = compound
```

1.4 Preprocessing of $project_tit \leq$

In [23]:

```
# similarly you can preprocess the titles also

# similarly you can preprocess the titles also

project_data.columns
#sent1= decontracted(project_data['project_title'].values[200
00])
preprocessed_title = []
# tqdm is for printing the status bar
for sentance in tqdm(project_data['project_title'].values):
    sent1 = decontracted(sentance)
    sent1 = sent1.replace('\\r', ' ')
    sent1 = sent1.replace('\\"', ' ')
    sent1 = sent1.replace('\\n', ' ')
    sent1 = re.sub('[^A-Za-z0-9]+', ' ', sent1)
    # https://gist.github.com/sebleier/554280
    sent1 = ' '.join(e for e in sent1.split() if e not in sto
pwords)
    preprocessed_title.append(sent.lower().strip())
```

```
100%|██████████| 109248/109248 [00:02<00:00, 4
3985.78it/s]
```

In [24]:

```
#Project title word count
title_word_count = []

for a in project_data["project_title"] :
    b = len(a.split())
    title_word_count.append(b)
```

```
project_data["title_word_count"] = title_word_count
```

In [25]:

```
project_data['preprocessed_title'] = preprocessed_title
```

1.5 Preparing data for models

In [26]:

```
project_data.columns
```

Out[26]:

```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teac
her_prefix', 'school_state',
       'Date', 'project_grade_category', 'proj
ect_title', 'project_essay_1',
       'project_essay_2', 'project_essay_3', '
project_essay_4',
       'project_resource_summary',
       'teacher_number_of_previously_posted_pr
ojects', 'project_is_approved',
       'clean_categories', 'clean_subcategorie
s', 'essay', 'essay_word_count',
       'preprocessed_essays', 'pos', 'neg', 'n
eu', 'compound',
       'title_word_count', 'preprocessed_title
'],
      dtype='object')
```

In [27]:

```
Y=project_data['project_is_approved']
```

In [28]:

```
price_data = resource_data.groupby('id').agg({'price':'sum',
'quantity':'sum'}).reset_index()
project_data = pd.merge(project_data, price_data, on='id', ho
w='left')
```

In [29]:

```
project_data['preprocessed_essays'] = preprocessed_essays
project_data['preprocessed_title'] = preprocessed_title
```

In [30]:

```
column_values=['clean_categories', 'clean_subcategories', 'sc
hool_state', 'project_grade_category', 'teacher_prefix','prep
rocessed_essays','preprocessed_title' ,'price','quantity','te
acher_number_of_previously_posted_projects','pos','neg','neu',
'compound','title_word_count','essay_word_count']

def select_columns(dataframe, column_names):
    new_frame = dataframe.loc[:, column_names]
    return new_frame

process_columns=select_columns(project_data,column_values)

process_columns.head()
```

Out[30]:

	clean_categories	clean_subcategories	school_state	project_grade_category
0	Math_Science	AppliedSciences Health_LifeScience	CA	Grades PreK-2
1	SpecialNeeds	SpecialNeeds	UT	Grades 3-5
2	Literacy_Language	Literacy	CA	Grades PreK-2
3	AppliedLearning	EarlyDevelopment	GA	Grades PreK-2
4	Literacy_Language	Literacy	WA	Grades 3-5

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 : nu
   merical
       - price : numerical
```

1.5.1 Vectorizing Categorical data

In [31]:

```
# we use count vectorizer to convert the values into one
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_categories= CountVectorizer(vocabulary=list(sorted
_cat_dict.keys()), lowercase=False, binary=True)

vectorizer_categories.fit(process_columns['clean_categories'].
values)

categories_one_hot = vectorizer_categories.transform(process_
columns['clean_categories'].values)



print(vectorizer_categories.get_feature_names())
print("Shape of  matrix after one hot encodig ",categories_on
e_hot.shape)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'M
usic_Arts', 'AppliedLearning', 'SpecialNeeds',
 'Health_Sports', 'Math_Science', 'Literacy_La
nguage']
Shape of  matrix after one hot encodig  (10924
8, 9)
```

In [32]:

```
# we use count vectorizer to convert the values into one
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_subcategories = CountVectorizer(vocabulary=list(so
rted_sub_cat_dict.keys()), lowercase=False, binary=True)
vectorizer_subcategories.fit(process_columns['clean_subcatego
ries'].values)

print(vectorizer_subcategories.get_feature_names())

sub_categories_one_hot = vectorizer_subcategories.transform(p
rocess_columns['clean_subcategories'].values)

print("Shape of  matrix after one hot encodig ",sub_categorie
s_one_hot.shape)
```

```
['Economics', 'CommunityService', 'FinancialLi
teracy', 'ParentInvolvement', 'Extracurricular
', 'Civics_Government', 'ForeignLanguages', 'N
utritionEducation', 'Warmth', 'Care_Hunger', '
SocialSciences', 'PerformingArts', 'CharacterE
ducation', 'TeamSports', 'Other', 'College_Car
eerPrep', 'Music', 'History_Geography', 'Healt
h_LifeScience', 'EarlyDevelopment', 'ESL', 'Gy
m_Fitness', 'EnvironmentalScience', 'VisualArt
s', 'Health_Wellness', 'AppliedSciences', 'Spe
cialNeeds', 'Literature_Writing', 'Mathematics
', 'Literacy']
Shape of  matrix after one hot encodig  (10924
8, 30)
```

In [33]:

```
# we use count vectorizer to convert the values of categorica
l data :school_state
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_schoolstate= CountVectorizer()
vectorizer_schoolstate.fit(process_columns['school_state'])

print(vectorizer_schoolstate.get_feature_names())

school_state_one_hot = vectorizer_schoolstate.transform(proce
ss_columns['school_state'].values)

print("Shape of  matrix after one hot encodig ",school_state_
one_hot.shape)
```

```
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc
', 'de', 'fl', 'ga', 'hi', 'ia', 'id', 'il', '
in', 'ks', 'ky', 'la', 'ma', 'md', 'me', 'mi',
 'mn', 'mo', 'ms', 'mt', 'nc', 'nd', 'ne', 'nh
', 'nj', 'nm', 'nv', 'ny', 'oh', 'ok', 'or', '
pa', 'ri', 'sc', 'sd', 'tn', 'tx', 'ut', 'va',
 'vt', 'wa', 'wi', 'wv', 'wy']
Shape of  matrix after one hot encodig  (10924
8, 51)
```

In [34]:

```
#we use count vectorizer to convert the values of categorical
 data :project_grade_category
```

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_project_grade_category = CountVectorizer(stop_word
s=None)

k=process_columns['project_grade_category']


k.replace(['Grades PreK-2', 'Grades 6-8', 'Grades 3-5','Grade
s 9-12'], ['A1', 'B2' ,'C3', 'D4'],inplace=True)

vectorizer_project_grade_category.fit(k)

project_grade_category_one_hot=vectorizer_project_grade_categ
ory.transform(process_columns['project_grade_category'].values
)

print("Shape of matrix after one hot encodig ",project_grade_
category_one_hot.shape)
```

```
Shape of matrix after one hot encodig  (109248
, 4)
```

In [35]:

```
#we use count vectorizer to convert the values of categorical
 data : teacher_prefix
# getting error as we have null balues replacing them with 0
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_teacher_prefix = CountVectorizer()
project_data['teacher_prefix'].unique()

process_columns['teacher_prefix'].fillna("", inplace = True)


vectorizer_teacher_prefix.fit(process_columns['teacher_prefix'
].values)
```

```
print(vectorizer_teacher_prefix.get_feature_names())

teacher_prefix_one_hot = vectorizer_teacher_prefix.transform(
process_columns['teacher_prefix'].values)

print("Shape of  matrix after one hot encodig ",teacher_prefi
x_one_hot.shape)
```

```
['dr', 'mr', 'mrs', 'ms', 'teacher']
Shape of  matrix after one hot encodig  (10924
8, 5)
```

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

1.5.2 Vectorizing Text data

1.5.2.1 Bag of words

In [36]:

```
# We are considering only the words which appeared in at leas
t 10 documents(rows or projects).
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_bow_essay = CountVectorizer(min_df=10)
vectorizer_bow_essay.fit(process_columns['preprocessed_essays'
])

text_bow= vectorizer_bow_essay.transform(process_columns['pre
processed_essays'])

print("Shape of  matrix after one hot encodig ",text_bow.shap
e)
```

```
Shape of  matrix after one hot encodig  (10924
8, 16623)
```

In [37]:

```
# before you vectorize the title make sure you preprocess it
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_bow_title = CountVectorizer(min_df=10)
vectorizer_bow_title.fit(process_columns['preprocessed_title'
])

title_bow = vectorizer_bow_title.transform(process_columns['p
reprocessed_title'])

print("Shape of matrix after one hot encodig title_bow",title
_bow.shape)
```

```
Shape of matrix after one hot encodig title_bo
w (109248, 91)
```

1.5.3 Vectorizing Numerical features

In [38]:

```
price_data = resource_data.groupby('id').agg({'price':'sum',
'quantity':'sum'}).reset_index()
project_data = pd.merge(project_data, price_data, on='id', ho
w='left')
```

In [39]:

```
#scaling of price feature
```

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

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

price_scalar = Normalizer()
price_scalar.fit(process_columns['price'].values.reshape(-1,1
)) # finding the mean and standard deviation of this data


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

print(price_standardized.shape)
```

```
(109248, 1)
```

In [40]:

```
#scaling of qunatity feature

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

# price_standardized = standardScalar.fit(project_data['price
'].values)
```

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

quantity_scalar = Normalizer()
quantity_scalar.fit(process_columns['quantity'].values.reshape
(-1,1)) # finding the mean and standard deviation of this dat
a



# Now standardize the data with above maen and variance.
quantity_standardized= quantity_scalar.transform(process_colu
mns['quantity'].values.reshape(-1, 1))

print(quantity_standardized.shape)
```

```
(109248, 1)
```

In [41]:

```
#scaling of teachers number of previously posted projects

from sklearn.preprocessing import Normalizer

normalizer_projects_num = Normalizer()

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

normalizer_projects_num.fit(process_columns['teacher_number_o
f_previously_posted_projects'].values.reshape(-1,1))
```

```
prev_projects = normalizer_projects_num.transform(process_col
umns['teacher_number_of_previously_posted_projects'].values.r
eshape(-1,1))

print(prev_projects.shape)
```

```
(109248, 1)
```

In [42]:

```
# normalixing the title word count

from sklearn.preprocessing import Normalizer

normalizer_title_word = Normalizer()

normalizer_title_word.fit(process_columns['title_word_count'].
values.reshape(-1,1))

title_word_count = normalizer_title_word.transform(process_co
lumns['title_word_count'].values.reshape(-1,1))

print(title_word_count.shape)
print("="*100)
```

```
(109248, 1)
==============================================
==============================================
========
```

In [43]:

```
# normalixing the essay word count

from sklearn.preprocessing import Normalizer

normalizer_ess_count = Normalizer()
```

```
normalizer_ess_count.fit(process_columns['essay_word_count'].
values.reshape(-1,1))

essay_word_count = normalizer_ess_count.transform(process_col
umns['essay_word_count'].values.reshape(-1,1))

print(essay_word_count.shape)
```

```
(109248, 1)
```

In [44]:

```
#normalizing the data for  essay sentiment-pos
from sklearn.preprocessing import Normalizer
normalizer_pos = Normalizer()

normalizer_pos.fit(process_columns['pos'].values.reshape(-1,1
))

essay_sent_pos = normalizer_pos.transform(process_columns['po
s'].values.reshape(-1,1))

print(essay_sent_pos.shape)
```

```
(109248, 1)
```

In [45]:

```
#normalizing the data for  essay sentiment-neg
from sklearn.preprocessing import Normalizer

normalizer_neg= Normalizer()

normalizer_neg.fit(process_columns['neg'].values.reshape(-1,1
))

essay_sent_neg = normalizer_neg.transform(process_columns['ne
g'].values.reshape(-1,1))

print(essay_sent_neg.shape)
```

```
(109248, 1)
```

In [46]:

```
#normalizing the data for  essay sentiment-neu
from sklearn.preprocessing import Normalizer

normalizer_nue= Normalizer()

normalizer_nue.fit(process_columns['neu'].values.reshape(-1,1
))

essay_sent_nue = normalizer_nue.transform(process_columns['ne
u'].values.reshape(-1,1))

print(essay_sent_nue.shape)
```

```
(109248, 1)
```

In [47]:

```
#normalizing the data for  essay sentiment-compound
from sklearn.preprocessing import Normalizer

normalizer_compound= Normalizer()

normalizer_compound.fit(process_columns['compound'].values.re
shape(-1,1))

essay_sent_comp = normalizer_compound.transform(process_colum
ns['compound'].values.reshape(-1,1))

print(essay_sent_comp.shape)
print("="*100)
```

```
(109248, 1)
==============================================
==============================================
========
```

1.5.4 Merging all the above features

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

In [48]:

```
from scipy.sparse import hstack

#define categorical and numerical features
cat_num=hstack((school_state_one_hot,categories_one_hot,sub_categories_one_hot,teacher_prefix_one_hot,project_grade_category_one_hot,price_standardized, quantity_standardized, prev_projects, title_word_count, essay_word_count, essay_sent_pos, essay_sent_neg, essay_sent_nue, essay_sent_comp))

#combining categorical  numerical ,project_title(BOW)  and preprocessed_essay (BOW)
set1_train = hstack((cat_num, text_bow,title_bow))
```

In [90]:

```
#saving all the variables for future use

import pickle
f=open('10_variables.pckl','wb')
pickle.dump([set1_train],f)
f.close()
```

In [ ]:

Assignment 10: 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/reduction algorithms 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 and try a different number of clusters like 2,5 etc.
    ● You can take less data points (as this is very computationally expensive one) 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.
    ● You can take a smaller sample size for this as well.
- 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 Choose the best data matrix on which you got the best AUC

considering only 10k points as im facing issues with laptop

In [49]:

```
set1_train.shape
```

Out[49]:

```
(109248, 16822)
```

In [50]:

```
from sklearn.feature_selection import SelectKBest
feature= SelectKBest(k=5000)
```

In [51]:

```
X_all=feature.fit_transform(set1_train,Y)
```

```
C:\Users\Public\Anaconda3\lib\site-packages\sk
learn\feature_selection\univariate_selection.p
y:114: UserWarning:

Features [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
are constant.
```

In [52]:

```
X_all.shape
```

Out[52]:

```
(109248, 5000)
```

In [53]:

```
type(X_all)
```

Out[53]:

```
scipy.sparse.csr.csr_matrix
```

In [54]:

```
x_vales=X_all.toarray()
```

In [56]:

```
type(x_vales)
```

Out[56]:

```
numpy.ndarray
```

In [55]:

```
# taking 10k data points with random index
index = np.random.choice(x_vales.shape[0], 10000, replace=Fal
se)
x_vales_10k=x_vales[index,:]
```

In [68]:

```
# stroing essay vector for forming word cloud
final_essay=process_columns['preprocessed_essays']
```

In [58]:

```
final_essay[index]
```

Out[58]:

```
64390     my kindergarten students amazing chi
ldren they...
85915     our classroom full 21 students eager
 excited l...
75243     this year i teaching sixth grade lan
guage arts...
66575     i loved school i student it goal stu
dents love...
87072     my school highest poverty area phila
delphia mo...
2683      my students special needs i special 
education ...
8652      our school district serves 2500 tota
l students...
9849      my students amazing group 45 childre
n come poo...
66374     whatever good one abraham lincoln th
ese words ...
12972     my students different areas within n
ewport new...
22065     technology use technology common den
ominator s...
93386     our classroom working develop commun
ity strong...
92704     our school located pacoima impoveris
hed city c...
42728     my students absolutely incredible i 
not think ...
91219     my students urban international some
 students ...
90101     my students 5th 8th graders read gra
de level t...
34624     this coming school year added 10th s
econd grad...
91866     our students come diverse background
s experien...
82151     our school mission statement every c
```

```
hild every...
94045     during academic school year signific
ant portio...
47059     i school library media specialist ti
tle 1 scho...
85983     my students come school every day sm
ile faces ...
75033     this year i blessed also working alo
ngside thr...
98902     i teach 100 seventh graders they coo
l school u...
91910     my students bring much joy happiness
 daily bas...
2029      it experience students learn love bo
oks read b...
85719     our students different backgrounds a
cross nort...
1158      do remember first time earned trophy
 ribbon wh...
40342     our school democratic constructionis
t based ch...
64424     my school great school located city
atlanta th...
                                ...

21829     our day typically consists play base
d learning...
103003    my fourth grade dual language studen
ts come hi...
59149     my students not access lot technolog
y we limit...
85414     i work rural elementary school servi
cing kinde...
56412     each morning students eager excited
come schoo...
35000     we active group second graders we ti
tle one sc...
```

```
35100     my students looking forward getting 
back routi...
10753     kids engaged participatory learning 
includes h...
43304     my students diverse dedicated kind s
tudents li...
92002     i work school full diverse backgroun
ds culture...
101607    every morning students come school r
eady learn...
101195    my students come inner city visual p
erforming ...
24842     my kindergarten students walk classr
oom every ...
104954    my students come significantly behin
d reading ...
51207     all students succeed not day way geo
rge evans ...
95576     i teach preschool title i school ser
ves divers...
92040     my students come everyday eager lear
n thrive s...
17420     each 4th 5th grader school opportuni
ty part ba...
36851     there never dull moment classroom wh
at i love ...
51573     my classroom second graders energeti
c learners...
49295     my 2nd grade students come variety d
ifferent e...
65698     my amazing eighth grade students liv
e staten i...
71160     our students ages 3 5 years old our 
classroom ...
70156     our school rural school population 7
51 people ...
47908     we title i school 98 students receiv
```

```
ing free r...
92498     the freedom make choices important s
tudents es...
103049    the students i work identified acade
mically in...
13378     as incoming first year teacher i not
 lucky eno...
54397     i kindergarten teacher loves working
 little on...
79466     i teach inclusion first grade my cla
ssroom mad...
Name: preprocessed_essays, Length: 10000, dtyp
e: object
```

In [59]:

```
x_vales_10k.shape
```

Out[59]:

```
(10000, 5000)
```

In [91]:

```
import pickle
f=open('10_variables_x_vales_10k.pckl','wb')
pickle.dump([x_vales_10k],f)
f.close()
```

In [54]:

```
import pickle as pickle
#with open('C:/Users/pramod reddy chandi/Desktop/pram/applied
 ai course/DonorsChoose_2018/cat_num.pckl', 'rb') as f:
f=open('C:/Users/pramod reddy chandi/Desktop/pram/applied ai
course/DonorsChoose_2018/10_variables_x_vales_10k.pckl','rb')
x_vales_10k=pickle.load(f)
f.close()
```

In [57]:

```
type(x_vales_10k)
```

Out[57]:

```
list
```

2.5 Apply Kmeans

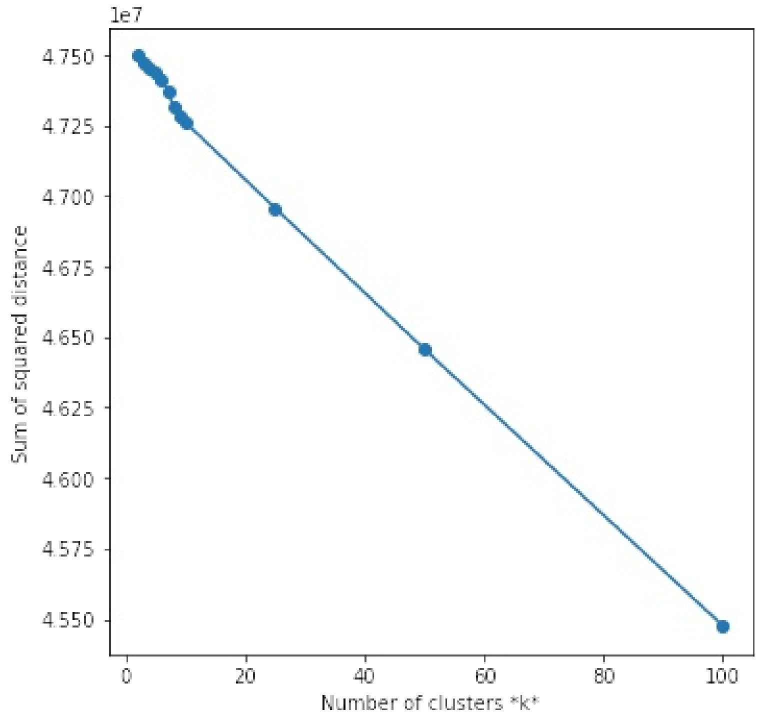
In [61]:

```
from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
scalar.fit(x_vales_10k)
X_vectors = scalar.transform(x_vales_10k)
print("The shape of the X_vectors is : {}".format(X_vectors.s
hape))
```

```
The shape of the X_vectors is : (10000, 5000)
```

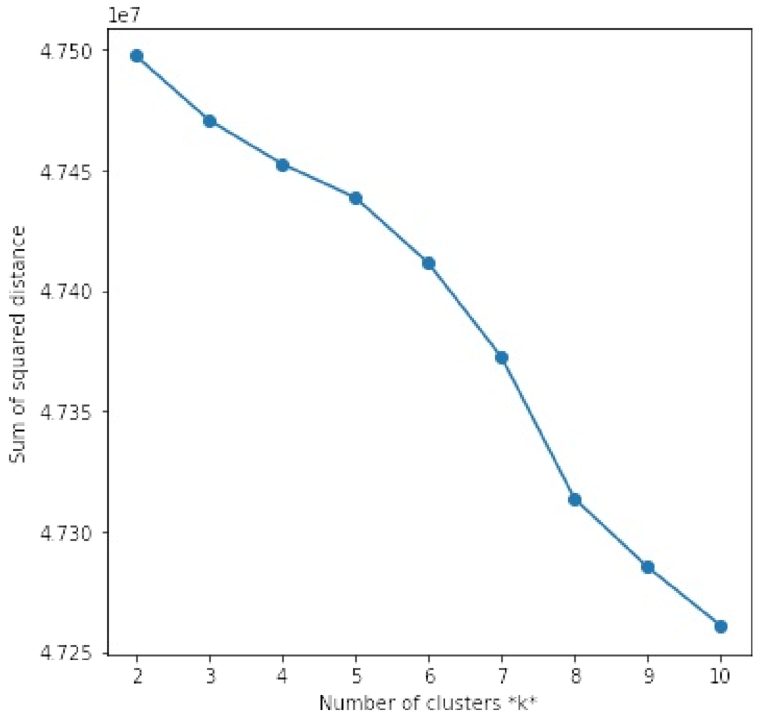
In [79]:

```
#Run the Kmeans algorithm and get the
from sklearn.cluster import KMeans, SpectralClustering
sse = []
list_k =[2,3,4,5,6,7,8,9,10,25,50,100]

for k in list_k:
    km = KMeans(n_clusters=k,random_state=42,n_jobs=-1,precom
pute_distances=True)
    km.fit(X_vectors)
    sse.append(km.inertia_)

# Plot sse against k
plt.figure(figsize=(6, 6))
plt.plot(list_k, sse, '-o')
plt.xlabel(r'Number of clusters *k*')
plt.ylabel('Sum of squared distance');
```



In [87]:

```
plt.figure(figsize=(6, 6))
plt.plot(list_k[0:9], sse[0:9], '-o')
plt.xlabel(r'Number of clusters *k*')
plt.ylabel('Sum of squared distance');
```


1e7
4.750
4.745
4.740
4.735
4.730
4.725
Sum of squared distance
2
3
4
5
6
7
8
9
10
Number of clusters *k*

we consider 8 as the best value as we can inflection at that point and afterthat MSE is decreasing faster than before that.

In [92]:

```
sse
```

Out[92]:

```
[47497470.40167063,
 47470787.67710049,
 47452625.71695323,
 47438717.452127986,
 47411427.041385256,
 47372405.07624722,
 47313597.48231196,
 47285213.73695182,
 47260894.972364634,
 46957258.79223214,
 46457112.94989913,
 45476412.156416714]
```

In [89]:

```
optimal_k = 8
# Variable that will be used in the conclusion
bow_means_k = optimal_k

# Implementing K-Means++ using optimal value of K
kmeans = KMeans(n_clusters=optimal_k, random_state=42,n_jobs=
-1,precompute_distances=True).fit(X_vectors)
```

In [106]:

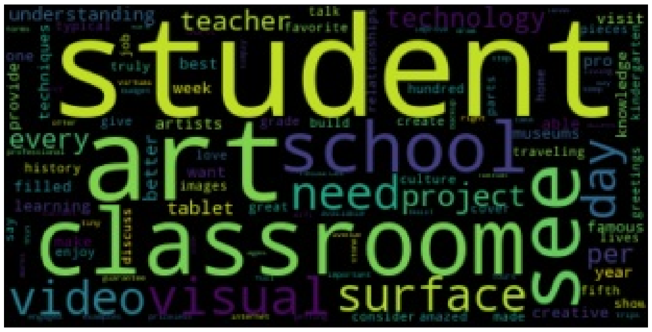
```
# considering the essay text vector for forming the word clou
d
essay_text = final_essay[index].values
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []
cluster4 = []
cluster5 = []
cluster6 = []
cluster7 = []
cluster8 = []

for i in range(kmeans.labels_.shape[0]):
    if kmeans.labels_[i] == 0:
        cluster1.append(essay_text[i])
    elif kmeans.labels_[i] == 1:
        cluster2.append(essay_text[i])
    elif kmeans.labels_[i] == 2:
        cluster3.append(essay_text[i])
    elif kmeans.labels_[i] == 3:
        cluster4.append(essay_text[i])
    elif kmeans.labels_[i] == 4:
        cluster4.append(essay_text[i])
    elif kmeans.labels_[i] == 5:
        cluster5.append(essay_text[i])
    elif kmeans.labels_[i] == 6:
        cluster6.append(essay_text[i])
    elif kmeans.labels_[i] == 7:
        cluster7.append(essay_text[i])
    else :
        cluster8.append(essay_text[i])

# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2))
print("\nNo. of reviews in Cluster-3 : ",len(cluster3))
print("\nNo. of reviews in Cluster-4 : ",len(cluster4))
print("\nNo. of reviews in Cluster-5 : ",len(cluster5))
print("\nNo. of reviews in Cluster-6 : ",len(cluster6))
print("\nNo. of reviews in Cluster-7 : ",len(cluster7))
print("\nNo. of reviews in Cluster-8 : ",len(cluster8))
```

```
No. of reviews in Cluster-1 :  1

No. of reviews in Cluster-2 :  73

No. of reviews in Cluster-3 :  1398

No. of reviews in Cluster-4 :  1717

No. of reviews in Cluster-5 :  1

No. of reviews in Cluster-6 :  6672

No. of reviews in Cluster-7 :  138

No. of reviews in Cluster-8 :  0
```

In [120]:

```
from wordcloud import WordCloud
essay_cluster=" ".join(essa for essa in cluster1)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 1")
plt.show()
```

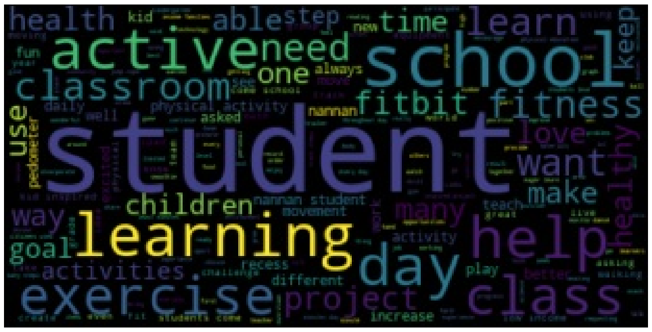
word cloud for cluster 1



In [121]:

```
from wordcloud import WordCloud
essay_cluster2=" ".join(essa for essa in cluster2)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster2)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 2")
plt.show()
```

word cloud for cluster 2

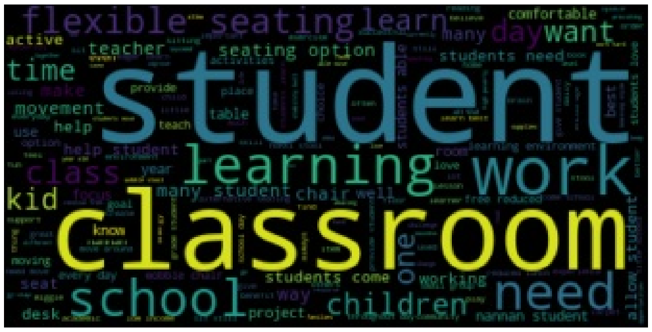


In [122]:

```
from wordcloud import WordCloud
essay_cluster3=" ".join(essa for essa in cluster3)
```

```
# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster3)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 3")
plt.show()
```

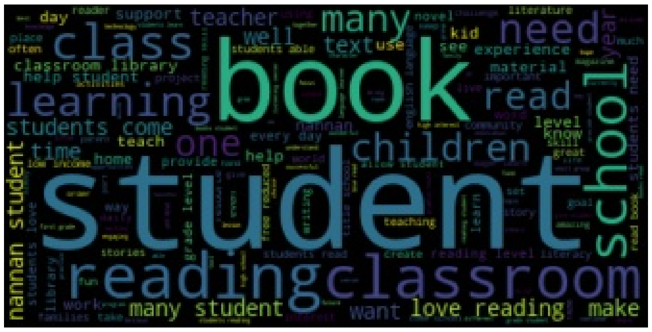
word cloud for cluster 3



In [123]:

```
from wordcloud import WordCloud
essay_cluster4=" ".join(essa for essa in cluster4)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster4)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 4")
plt.show()
```

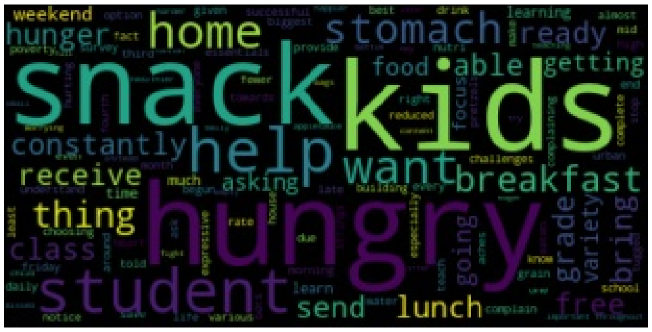
word cloud for cluster 4



In [124]:

```
from wordcloud import WordCloud
essay_cluster5=" ".join(essa for essa in cluster5)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster5)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 5")
plt.show()
```

word cloud for cluster 5

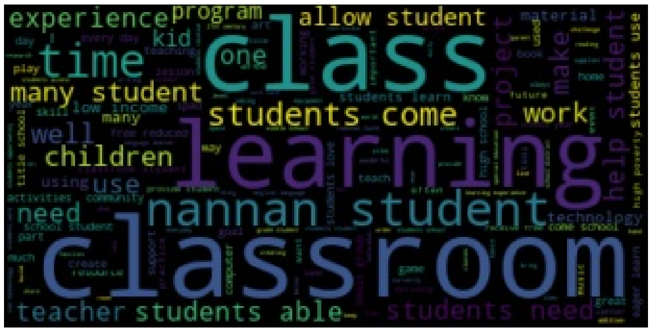


In [126]:

```
from wordcloud import WordCloud
essay_cluster6=" ".join(essa for essa in cluster6)
```

```
# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster6)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 6")
plt.show()
```

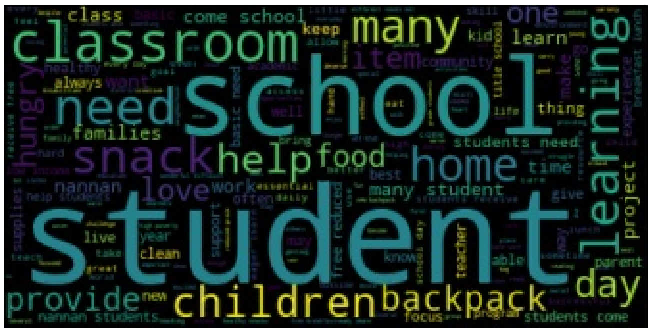
word cloud for cluster 6



In [127]:

```
from wordcloud import WordCloud
essay_cluster7=" ".join(essa for essa in cluster7)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster7)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 7")
plt.show()
```

word cloud for cluster 7



Inference: From the graph between sum of squared errors and no of clusters ,we can get 8 as the optimal clusters from the elbow method. we can observe that maximum no of points belong to cluster 6 with 6672 essays with almost constitute 66.72%

In cluster 1 we can see most frequent words like student ,Art,classroom ,need project it means they mostly describe about the students and need for project funding. In cluster 2 we can see most frequent words like student,learning,active,school,day which describes about the learning activities of the students. In cluster 3 ,we can see the most frequent words like student,learning ,work ,classroom which indicates the essence of project funding for the schoool. In cluster 4 ,we see most important words like book,student ,reading ,classroom etc from which we can conclude that it mainly describes about the activities of the student . In cluster 5 we can see most frequent words like snacks,hungry,kids ,breakfast which constitues most of the essays and it gives information about the need of snacks for students ,One may draw that the students belong to lower primary . In cluster 6 we can see most frequent words like classroom,class ,student which gives information about the necessary information about the stationary to the students. In cluster 7 we can see most frequent words like student,school,children,backpack ,snacks which gives inforamtion about the student much needs.

we can infer from the graph that the sum of squared errors with 8 clusters is 47313597 which have only considered top 5000 features with 10k random points from the whole data.

Please note that the final model varies depending on the increase of data points taken.

2.6 Apply AgglomerativeClustering

cluster 2

In [128]:

```
from sklearn.cluster import AgglomerativeClustering

model = AgglomerativeClustering(n_clusters=2).fit(X_vectors)

essay_text = final_essay[index].values

# Getting all the reviews in different clusters
cluster1_AC2 = []
cluster2_AC2 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1_AC2.append(essay_text[i])
    else :
        cluster2_AC2.append(essay_text[i])


# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1_AC2))
print("\nNo. of reviews in Cluster-2 : ",len(cluster2_AC2))
```

```
No. of reviews in Cluster-1 :  9994

No. of reviews in Cluster-2 :  6
```

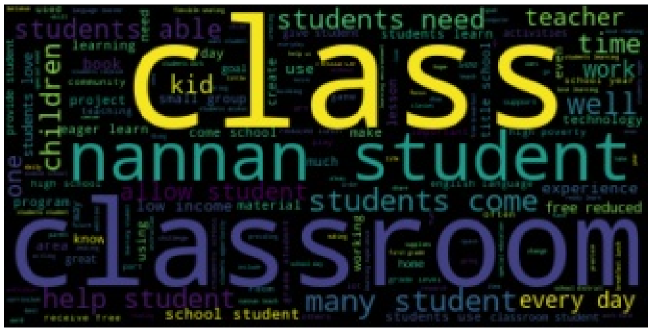
In [ ]:

```
import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10, 7))
plt.title("cluster 2 Dendograms")
dend = shc.dendrogram(shc.linkage(X_vectors, method='ward'))
```

In [129]:

```
from wordcloud import WordCloud
essay_cluster1_ac2=" ".join(essa for essa in cluster1_AC2)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster1_ac2)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 1 AC2")
plt.show()
```

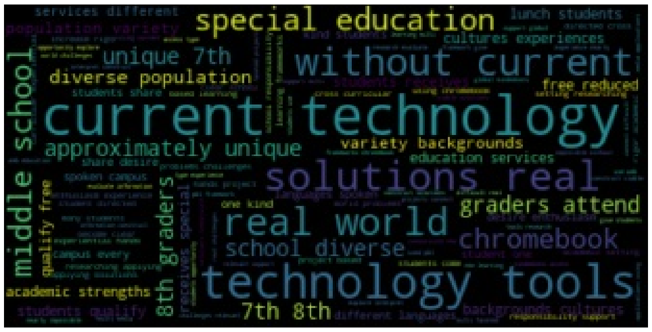
word cloud for cluster 1 AC2



In [133]:

```
from wordcloud import WordCloud
essay_cluster2_ac2=" ".join(essa for essa in cluster2_AC2)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster2_ac2)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 2 AC2")
plt.show()
```

word cloud for cluster 2 AC2



Inference: We can see that when 2 clusters are considered , the maximum no of points belong to cluser 1 with almost 99.96%. in cluster 1 we can see most frequent words like class,student ,classroom ,nanan In cluster 2 we can see most frequent words like technology,tools ,current,special edition.

In [131]:

```
len(essay_cluster2_ac2)
```

Out[131]:

```
10641
```

In [132]:

```
len(essay_cluster1_ac2)
```

Out[132]:

```
10603329
```

n clusters 5

In [134]:

```
from sklearn.cluster import AgglomerativeClustering

model = AgglomerativeClustering(n_clusters=5).fit(X_vectors)
```

In [141]:

```
essay_text = final_essay[index].values

# Getting all the reviews in different clusters
cluster1_AC5 = []
cluster2_AC5 = []
cluster3_AC5 = []
cluster4_AC5 = []
cluster5_AC5 = []

for i in range(model.labels_.shape[0]):
    if model.labels_[i] == 0:
        cluster1_AC5.append(essay_text[i])
    elif model.labels_[i] == 1:
        cluster2_AC5.append(essay_text[i])
    elif model.labels_[i] == 2:
        cluster3_AC5.append(essay_text[i])
    elif model.labels_[i] == 3:
        cluster4_AC5.append(essay_text[i])
    else :
        cluster5_AC5.append(essay_text[i])
```

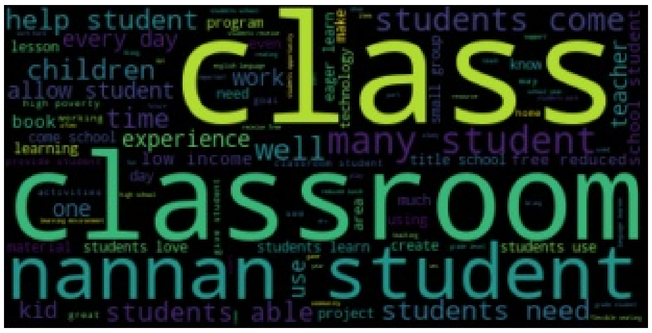
```python
# Number of reviews in different clusters
print("No. of reviews in Cluster-1 : ",len(cluster1_AC5))
print("No. of reviews in Cluster-2 : ",len(cluster2_AC5))
print("No. of reviews in Cluster-3 : ",len(cluster3_AC5))
print("No. of reviews in Cluster-4 : ",len(cluster4_AC5))
print("No. of reviews in Cluster-5: ",len(cluster5_AC5))
```

```
No. of reviews in Cluster-1 :  9975
No. of reviews in Cluster-2 :  6
No. of reviews in Cluster-3 :  13
No. of reviews in Cluster-4 :  5
No. of reviews in Cluster-5:  1
```

In [142]:

```python
from wordcloud import WordCloud
essay_cluster1_ac5=" ".join(essa for essa in cluster1_AC5)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster1_ac5)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 1 AC5")
plt.show()
```

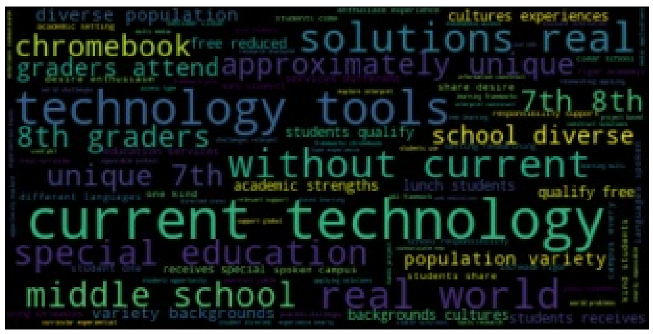
word cloud for cluster 1 AC5



In [143]:

```python
from wordcloud import WordCloud
essay_cluster2_ac5=" ".join(essa for essa in cluster2_AC5)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster2_ac5)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 2 AC5")
plt.show()
```

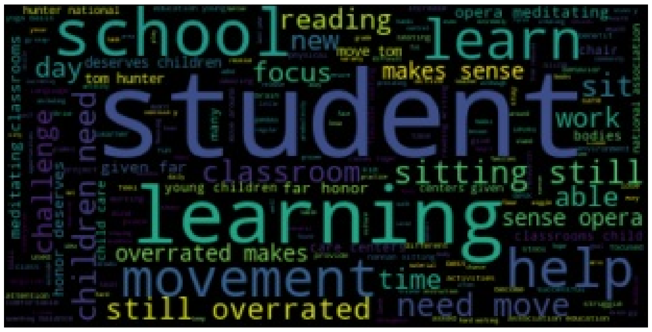
word cloud for cluster 2 AC5



In [144]:

```python
from wordcloud import WordCloud
essay_cluster3_ac5=" ".join(essa for essa in cluster3_AC5)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster3_ac5)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 3 AC5")
plt.show()
```

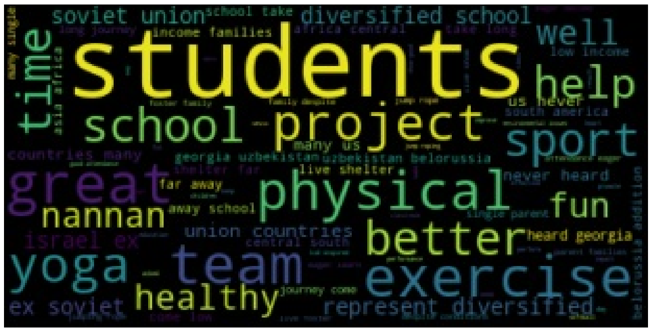
word cloud for cluster 3 AC5



In [145]:

```
from wordcloud import WordCloud
essay_cluster4_ac5=" ".join(essa for essa in cluster4_AC5)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster4_ac5)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 4 AC5")
plt.show()
```

word cloud for cluster 4 AC5

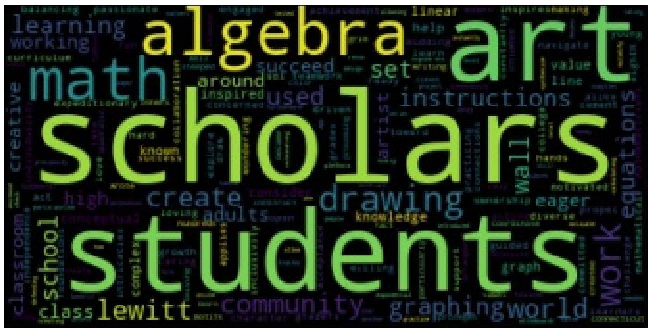


In [146]:

```
from wordcloud import WordCloud
essay_cluster5_ac5=" ".join(essa for essa in cluster5_AC5)
```

```python
# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster5_ac5)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 5 AC5")
plt.show()
```

word cloud for cluster 5 AC5



Inference: We can see that when 5 clusters are considered , the maximum no of points belong to cluster 1 with almost 99.75%. in cluster 1 we can see most frequent words like class,student ,classroom ,nanan it means they mostly describe about the students and classroom details. In cluster 2 we can see most frequent words like technology,tools ,current technology ,special edition which describes about the need for project funding. In cluster 3 ,we can see the most frequent words like student,school,learning,need overrated etc which indicates the essence of project funding for the schoool. In cluster 4 ,we see most important words like yoga ,exercises,physical ,better which indicateds the need for the physical activity in schools. in cluster 5 we can see most frequent words like algebra,art ,math,scholars,drawing which belong to the extracurricular actitities as well as subjects to bne incorporated or stressed among students. WE can alanyse the data better with increase in clusters.

2.7 Apply DBSCAN

for 5k features we need to take 5k as min points and lap is unable to execute the command so we take bow vectror with mindif 1000 and max features as 50

In [49]:

```
# We are considering only the words which appeared in at leas
t 1000 documents(rows or projects).
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_bow_essay = CountVectorizer(min_df=1000)
vectorizer_bow_essay.fit(process_columns['preprocessed_essays'
])

text_bow_db= vectorizer_bow_essay.transform(process_columns['
preprocessed_essays'])

print("Shape of  matrix after one hot encodig ",text_bow_db.s
hape)
```

```
Shape of  matrix after one hot encodig  (10924
8, 1766)
```

In [50]:

```
#combining categorical  numerical ,project_title(BOW)  and pr
eprocessed_essay (BOW)
set_db = hstack((cat_num, text_bow_db))
```

In [51]:

```
set_db.shape
```

Out[51]:

```
(109248, 1874)
```

In [52]:

```
data=set_db.toarray()
```

In [58]:

```
# taking random 10k samples from the data
index = np.random.choice(109248, 10000, replace=False)
data_10k=data[index,:]
y_10k=Y[index]
```

In [62]:

```
data_10k.shape
```

Out[62]:

```
(10000, 1874)
```

In [63]:

```
y_10k.shape
```

Out[63]:

```
(10000,)
```

In [59]:

```
#credit to https://github.com/PushpendraSinghChauhan/Amazon-F
ine-Food-Reviews/blob/master/Apply%20DBSCAN%20on%20Amazon%20F
ine%20Food%20Reviews.ipynb
# function to determinethe distance of nth-nearest neighbour
to all points in a multi-dimensional array
```

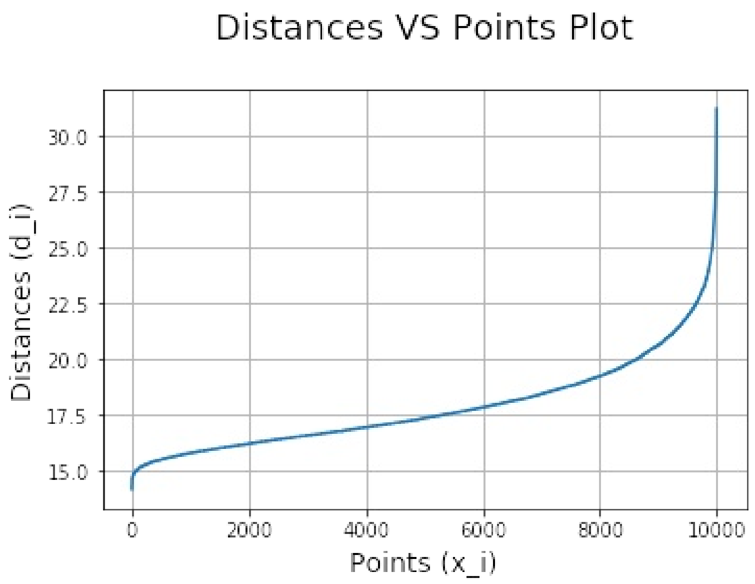
```
def n_neighbour(vectors , n):
    distance = []
    for point in vectors:
        temp = np.sort(np.sum((vectors-point)**2,axis=1),axis
=None)
        distance.append(temp[n])
    return np.sqrt(np.array(distance))
```

In [60]:

```
# Function definition for implementing DBSCAN
def dbscan(epsilon, samples, Data):
    from sklearn.cluster import DBSCAN
    db = DBSCAN(eps=epsilon, min_samples=samples, n_jobs=-1).
fit(Data)

    # Number of clusters in labels, ignoring noise(-1) if pre
sent.
    n_clusters = len(set(db.labels_))
    print("Number of clusters for MinPts = %d and Epsilon = %
f is : %d "%(samples,epsilon,n_clusters))
    print("Labels(-1 is for Noise) : ",set(db.labels_))
    print()
    return db
```

In [61]:

```
min_points = 1875

# Computing distances of nth-nearest neighbours
distances = n_neighbour(data_10k,min_points)
sorted_distance = np.sort(distances)
points = [i for i in range(data_10k.shape[0])]

# Draw distances(d_i) VS points(x_i) plot
plt.plot(points, sorted_distance)
plt.xlabel('Points (x_i)',size=14)
plt.ylabel('Distances (d_i)',size=14)
plt.title('Distances VS Points Plot\n',size=18)
plt.grid()
plt.show()
```



In [64]:

```
# Clustering with right epsilon
db1 = dbscan(18, min_points, data_10k)


# Clustering with  epsilon = 19
db2 = dbscan(19, min_points, data_10k)


# Clustering with epsilon = 20
db3 = dbscan(20, min_points, data_10k)


# Clustering with epsilon = 21
db4 = dbscan(21, min_points, data_10k)
```

```
Number of clusters for MinPts = 1875 and Epsil
on = 18.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}
```

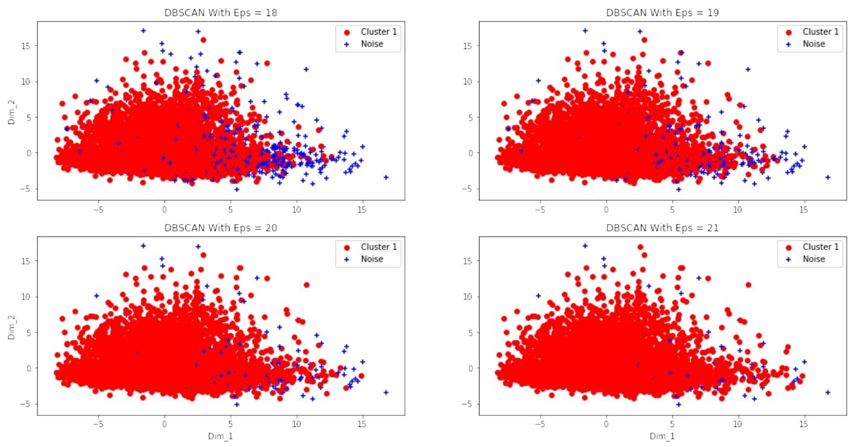
```
Number of clusters for MinPts = 1875 and Epsil
on = 19.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 1875 and Epsil
on = 20.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}

Number of clusters for MinPts = 1875 and Epsil
on = 21.000000 is : 2
Labels(-1 is for Noise) :  {0, -1}
```

In [65]:

```
from sklearn.decomposition import PCA
pca_2d = PCA(n_components=2).fit_transform(data_10k)
```

In [66]:

```
# Scatter plot for DBSCAN with Eps = 18
plt.figure(figsize=(18,9))
plt.subplot(221)
for i in range(0, pca_2d.shape[0]):
    if db1.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker
='o')
    elif db1.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker
='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 18')
plt.ylabel('Dim_2')

# Scatter plot for DBSCAN with Eps = 19
plt.subplot(222)
for i in range(0, pca_2d.shape[0]):
    if db2.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker
='o')
    elif db2.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker
='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 19')

# Scatter plot for DBSCAN with Eps = 20
plt.subplot(223)
for i in range(0, pca_2d.shape[0]):
    if db3.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker
='o')
    elif db3.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker
='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 20')
plt.ylabel('Dim_2')
plt.xlabel('Dim_1')

# Scatter plot for DBSCAN with Eps = 21
plt.subplot(224)
for i in range(0, pca_2d.shape[0]):
    if db4.labels_[i] == 0:
        c1 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='r',marker
='o')
    elif db4.labels_[i] == -1:
        c2 = plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker
='+')
plt.legend([c1, c2], ['Cluster 1', 'Noise'])
plt.title('DBSCAN With Eps = 21')
plt.xlabel('Dim_1')

plt.show()
```


DBSCAN With Eps = 18
DBSCAN With Eps = 19
DBSCAN With Eps = 20
DBSCAN With Eps = 21
● Cluster 1
+ Noise
Dim_1
Dim_2

forming word cloud

with epsilon value 18

In [69]:

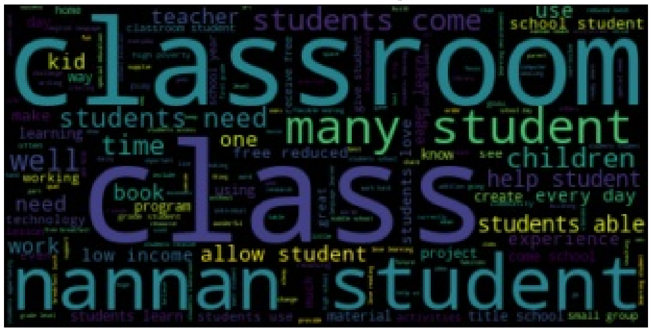
```
essay_text = final_essay[index].values

# Getting all the reviews in different clusters
cluster1_DB = []
cluster2_DB = []

for i in range(db1.labels_.shape[0]):
    if db1.labels_[i] == 0:
        cluster1_DB.append(essay_text[i])
    else :
        cluster2_DB.append(essay_text[i])



# Number of reviews in different clusters
print("No. of essays in Cluster-1 : ",len(cluster1_DB))
print("No. of essays in Cluster-2 : ",len(cluster2_DB))
```

```
No. of essays in Cluster-1 :  9299
No. of essays in Cluster-2 :  701
```

In [76]:

```
from wordcloud import WordCloud
essay_cluster1_DB=" ".join(essa for essa in cluster1_DB)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster1_DB)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 1 with epsilon 18 DB SCAN"
)
plt.show()
```

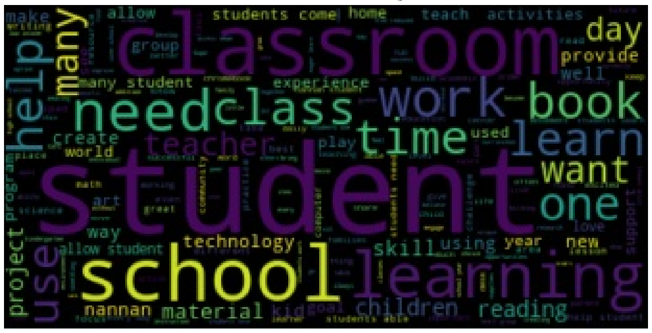
word cloud for cluster 1 with epsilon 18 DB SCAN



In [77]:

```
from wordcloud import WordCloud
essay_cluster2_DB=" ".join(essa for essa in cluster2_DB)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster2_DB)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 2 with epsilon 18 DB SCAN"
)
plt.show()
```

word cloud for cluster 2 with epsilon 18 DB SCAN



Inference:we could see most points belong to the cluster 1 which belong to corepoint cluster and consist of words like classroom,class ,student,many ,children and student interestly we got 701 out of 10k (7%)points who belong to noise cluster and consist of words like classroom, student, school .

We cannot consider this epsilon as the optimal value as it consist of more noise points .

with epsilon value 19

In [72]:

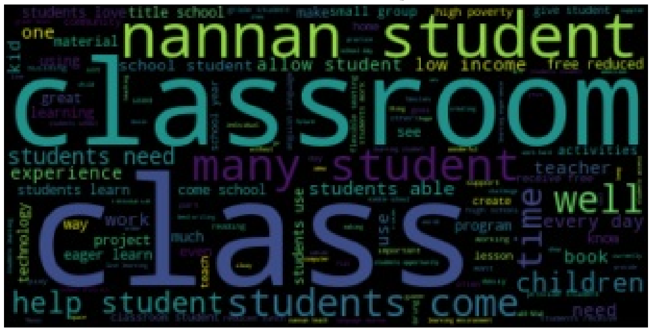
```
essay_text = final_essay[index].values

# Getting all the reviews in different clusters
cluster1_DB2 = []
cluster2_DB2 = []

for i in range(db2.labels_.shape[0]):
    if db2.labels_[i] == 0:
        cluster1_DB2.append(essay_text[i])
    else :
        cluster2_DB2.append(essay_text[i])



# Number of reviews in different clusters
print("No. of essays in Cluster-1 : ",len(cluster1_DB2))
print("No. of essays in Cluster-2 : ",len(cluster2_DB2))
```

```
No. of essays in Cluster-1 :  9644
No. of essays in Cluster-2 :  356
```

In [78]:

```
from wordcloud import WordCloud
essay_cluster1_DB2=" ".join(essa for essa in cluster1_DB2)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster1_DB2)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 1 with epsilon 19 DB SCAN"
)
plt.show()
```

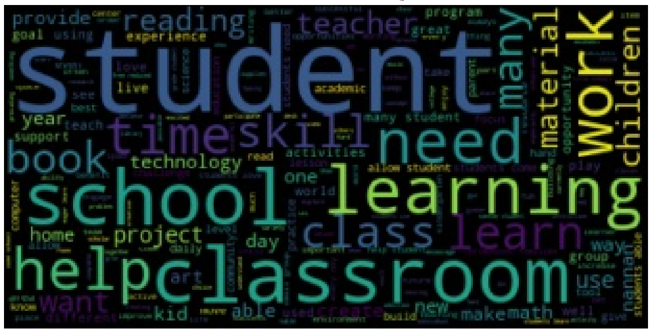
word cloud for cluster 1 with epsilon 19 DB SCAN



In [79]:

```
from wordcloud import WordCloud
essay_cluster2_DB2=" ".join(essa for essa in cluster2_DB2)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster2_DB2)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 2 with epsilon 19 DB SCAN"
)
plt.show()
```

word cloud for cluster 2 with epsilon 19 DB SCAN



Inference Cluster 1 which is corepoint cluster consist words like classroom class nanan student etc cluster 2 which consist of noise points almost constitute 3.56% and consist words like student ,school,help,classroom and learning etc WE see that as epsilon value increasing there is decrease in the noise points .

with epsilon value 20

In [80]:

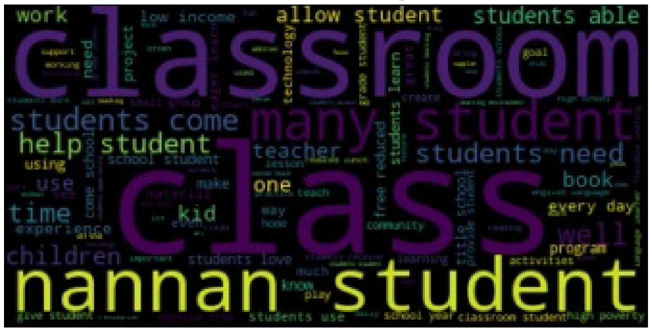
```
essay_text = final_essay[index].values

# Getting all the reviews in different clusters
cluster1_DB3 = []
cluster2_DB3 = []

for i in range(db3.labels_.shape[0]):
    if db3.labels_[i] == 0:
        cluster1_DB3.append(essay_text[i])
    else :
        cluster2_DB3.append(essay_text[i])



# Number of reviews in different clusters
print("No. of essays in Cluster-1 : ",len(cluster1_DB3))
print("No. of essays in Cluster-2 : ",len(cluster2_DB3))
```

```
No. of essays in Cluster-1 :  9824
No. of essays in Cluster-2 :  176
```

In [81]:

```
from wordcloud import WordCloud
essay_cluster1_DB3=" ".join(essa for essa in cluster1_DB3)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster1_DB3)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 1 with epsilon 20 DB SCAN"
)
plt.show()
```

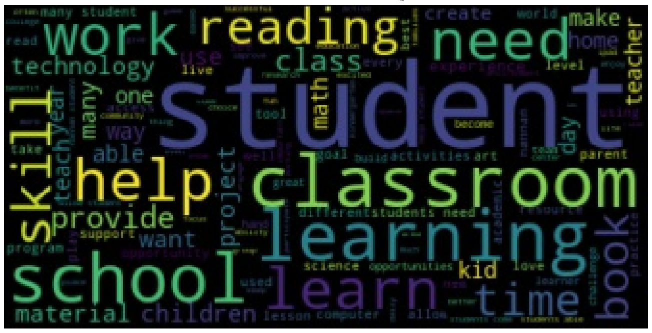
word cloud for cluster 1 with epsilon 20 DB SCAN



In [82]:

```
from wordcloud import WordCloud
essay_cluster2_DB3=" ".join(essa for essa in cluster2_DB3)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster2_DB3)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 2 with epsilon 20 DB SCAN"
)
plt.show()
```

word cloud for cluster 2 with epsilon 20 DB SCAN



Inference Cluster 1 which is corepoint cluster consist words like classroom class nanan student etc cluster 2 which consist of noise points almost constitute 1.76% and consist words like student ,school,help,classroom and learning etc WE see that as epsilon value increasing there is decrease in the noise points .

with epsilon value 21

In [83]:

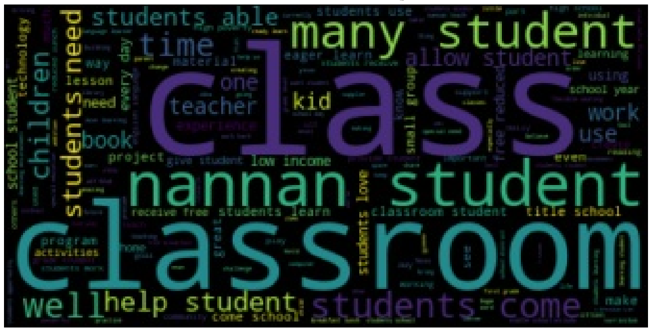
```
essay_text = final_essay[index].values

# Getting all the reviews in different clusters
cluster1_DB4 = []
cluster2_DB4 = []

for i in range(db4.labels_.shape[0]):
    if db4.labels_[i] == 0:
        cluster1_DB4.append(essay_text[i])
    else :
        cluster2_DB4.append(essay_text[i])



# Number of reviews in different clusters
print("No. of essays in Cluster-1 : ",len(cluster1_DB4))
print("No. of essays in Cluster-2 : ",len(cluster2_DB4))
```

```
No. of essays in Cluster-1 :  9923
No. of essays in Cluster-2 :  77
```

In [84]:

```
from wordcloud import WordCloud
essay_cluster1_DB4=" ".join(essa for essa in cluster1_DB4)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster1_DB4)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 1 with epsilon 21 DB SCAN"
)
plt.show()
```

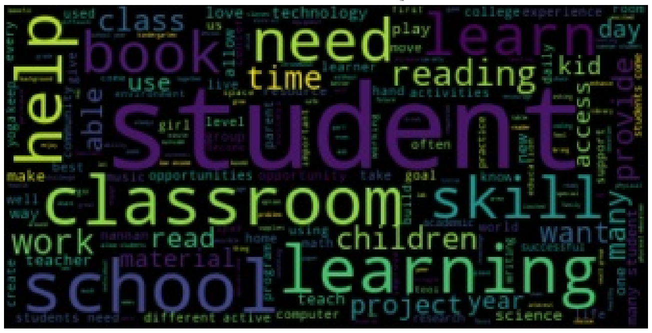
word cloud for cluster 1 with epsilon 21 DB SCAN



In [85]:

```
from wordcloud import WordCloud
essay_cluster2_DB4=" ".join(essa for essa in cluster2_DB4)

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(essay_cluster2_DB4)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title(" word cloud for cluster 2 with epsilon 21 DB SCAN"
)
plt.show()
```

word cloud for cluster 2 with epsilon 21 DB SCAN



Inference Cluster 1 which is corepoint cluster consist words like classroom class nanan student etc cluster 2 which consist of noise points almost constitute 0.77% and consist words like student ,school,help,classroom and learning etc WE see that as epsilon value increasing there is decrease in the noise points and this epsilon can be considered as the best value.

we can also infer that in all the clusters formed with different epsilons most of the words belong to same cluster like student belong to cluster 2 and class belong to cluster 1

procedure followed

1 took all the categorical and numerical data along with essay bow and title bow of donorchose dataset 2 since we have 109248 data we have taken only 10k data points as our laptop has only 8gb ram and it is getting struck with even 30k points 3 for Kmeans and hierarchial we have converted the features to 5000 using select k best features and for db scan we have only considered bow vector with only 50 features as to minimize the min points as we need to take atleast (D+1) points as the min minpoints . 4 K means

plotted (sse vs no of clusters) and got 8 as the clusters and formed wordcloud for each cluster with essay text

5 AgglomerativeClustering

took 2 and 5 clusters and formed wordcloud for each cluster with essay text

6 DB scan

After taking only bow vector with 50 most important features we got total 158 features so considered 316 as the min points and drawn plot with points on x axis and epsilon on y axis and got 11 as the best value

considered epsilon value 12,13 and 14 and formed wordcloud with each of these epsilon values and min points as 316 and formed the essay text values.

Conclusion

In [86]:

```
# http://zetcode.com/python/prettytable/

from prettytable import PrettyTable

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

x = PrettyTable()
x.field_names = ["algorithm", "clusters formed ", "Max point
cluster","no of points in max cluster"]

x.add_row(["K-Means", 8, 6, 6672])
x.add_row(["AgglomerativeClustering", 2, 1, 9994])
x.add_row(["AgglomerativeClustering", 5, 1, 9975])

x.add_row(["DBSCAN eps 18", 2, 1, 9229])
x.add_row(["DBSCAN eps 19", 2, 1, 9644])
x.add_row(["DBSCAN eps 20", 2, 1, 9824])
x.add_row(["DBSCAN eps 21", 2, 1, 9923])



print(x)
```

```
+--------------------------+-------------------+
-------------------+--------------------------
---+
|         algorithm        | clusters formed  |
 Max point cluster | no of points in max clust
er |
+--------------------------+-------------------+
------------------+-------------------------
---+
|         K-Means         |        8         |
         6         |             6672
   |
| AgglomerativeClustering |        2         |
         1         |             9994
   |
| AgglomerativeClustering |        5         |
         1         |             9975
   |
|      DBSCAN eps 18      |        2         |
         1         |             9229
   |
|      DBSCAN eps 19      |        2         |
         1         |             9644
   |
|      DBSCAN eps 20      |        2         |
         1         |             9824
   |
|      DBSCAN eps 21      |        2         |
         1         |             9923
   |
+-------------------------+-------------------+
------------------+-------------------------
---+
```