1. Importing Packages

In [1]:

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

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

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer
import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

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

from tqdm import tqdm
import os

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

2. Loading Data

In [145]:

```
project_data = pd.read_csv('D:\\train_data.csv',nrows = 10000)
resource_data = pd.read_csv('D:\\resources.csv')
```

In [3]:

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

```
Number of data points in train data (10000, 17)
==================================================
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_prefix' 'school_state'
 'project_submitted_datetime' 'project_grade_category'
 'project_subject_categories' 'project_subject_subcategories'
 'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects' 'project_is_approved']
```

In [4]:

```
print("Number of data points in resources data", resource_data.shape)
print(resource_data.columns.values)
```

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

In [5]:

```
resource_data.head()
```

Out[5]:

	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
2	p069063	Cory Stories: A Kid's Book About Living With Adhd	1	8.45
3	p069063	Dixon Ticonderoga Wood-Cased #2 HB Pencils, Bo...	2	13.59
4	p069063	EDUCATIONAL INSIGHTS FLUORESCENT LIGHT FILTERS...	3	24.95

In [6]:

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

Out[6]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_grade_cate
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	Grades P
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Grade

In [7]:

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

Out[7]:

	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 [8]:

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

Out[8]:

	id	price	quantity
0	p000001	459.56	7
1	p000002	515.89	21

In [9]:

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

In [10]:

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

Out[10]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_grade_cate
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	Grades P
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Grade

In [11]:

```
project_data['teacher_prefix'] = project_data['teacher_prefix'].replace(np.NaN,'Mrs.')
```

3. Text Preprocessing

3.1. Concatenating all essay text

In [12]:

```
# 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)
```

3.2. Preprocessing Essay text

In [13]:

```
# printing some random essays.
print(project_data['essay'].values[0])
print("="*50)
print(project_data['essay'].values[150])
print("="*50)
print(project_data['essay'].values[1000])
print("="*50)
print(project_data['essay'].values[9999])
print("="*50)
```

```
My students are English learners that are working on English as their second or third languages. W
e are a melting pot of refugees, immigrants, and native-born Americans bringing the gift of langua
ge to our school. \r\n\r\n We have over 24 languages represented in our English Learner program wi
th students at every level of mastery.  We also have over 40 countries represented with the
families within our school.  Each student brings a wealth of knowledge and experiences to us that
open our eyes to new cultures, beliefs, and respect.\"The limits of your language are the limits o
f your world.\"-Ludwig Wittgenstein  Our English learner's have a strong support system at home th
at begs for more resources.  Many times our parents are learning to read and speak English along s
ide of their children.  Sometimes this creates barriers for parents to be able to help their child
learn phonetics, letter recognition, and other reading skills.\r\n\r\nBy providing these dvd's and
players, students are able to continue their mastery of the English language even if no one at hom
```

players, students are able to continue their mastery of the English language even if no one at hom
e is able to assist. All families with students within the Level 1 proficiency status, will be a
offered to be a part of this program. These educational videos will be specially chosen by the En
glish Learner Teacher and will be sent home regularly to watch. The videos are to help the child
develop early reading skills.\r\n\r\nParents that do not have access to a dvd player will have the
opportunity to check out a dvd player to use for the year. The plan is to use these videos and ed
ucational dvd's for the years to come for other EL students.\r\nnannan
==========
The 51 fifth grade students that will cycle through my classroom this year all love learning, at l
east most of the time. At our school, 97.3% of the students receive free or reduced price lunch. O
f the 560 students, 97.3% are minority students. \r\nThe school has a vibrant community that loves
to get together and celebrate. Around Halloween there is a whole school parade to show off the bea
utiful costumes that students wear. On Cinco de Mayo we put on a big festival with crafts made by
the students, dances, and games. At the end of the year the school hosts a carnival to celebrate t
he hard work put in during the school year, with a dunk tank being the most popular activity.My st
udents will use these five brightly colored Hokki stools in place of regular, stationary, 4-legged
chairs. As I will only have a total of ten in the classroom and not enough for each student to hav
e an individual one, they will be used in a variety of ways. During independent reading time they
will be used as special chairs students will each use on occasion. I will utilize them in place of
chairs at my small group tables during math and reading times. The rest of the day they will be us
ed by the students who need the highest amount of movement in their life in order to stay focused
on school.\r\n\r\nWhenever asked what the classroom is missing, my students always say more Hokki
Stools. They can't get their fill of the 5 stools we already have. When the students are sitting i
n group with me on the Hokki Stools, they are always moving, but at the same time doing their
work. Anytime the students get to pick where they can sit, the Hokki Stools are the first to be ta
ken. There are always students who head over to the kidney table to get one of the stools who are
disappointed as there are not enough of them. \r\n\r\nWe ask a lot of students to sit for 7 hours
a day. The Hokki stools will be a compromise that allow my students to do desk work and move at th
e same time. These stools will help students to meet their 60 minutes a day of movement by
allowing them to activate their core muscles for balance while they sit. For many of my students,
these chairs will take away the barrier that exists in schools for a child who can't sit
still.nannan
==========
How do you remember your days of school? Was it in a sterile environment with plain walls, rows of
desks, and a teacher in front of the room? A typical day in our room is nothing like that. I work
hard to create a warm inviting themed room for my students look forward to coming to each day.\r\n
\r\nMy class is made up of 28 wonderfully unique boys and girls of mixed races in
Arkansas.\r\nThey attend a Title I school, which means there is a high enough percentage of free a
nd reduced-price lunch to qualify. Our school is an \"open classroom\" concept, which is very uniq
ue as there are no walls separating the classrooms. These 9 and 10 year-old students are very eage
r learners; they are like sponges, absorbing all the information and experiences and keep on wanti
ng more.With these resources such as the comfy red throw pillows and the whimsical nautical hangin
g decor and the blue fish nets, I will be able to help create the mood in our classroom setting to
be one of a themed nautical environment. Creating a classroom environment is very important in the
success in each and every child's education. The nautical photo props will be used with each child
as they step foot into our classroom for the first time on Meet the Teacher evening. I'll take pic
tures of each child with them, have them developed, and then hung in our classroom ready for their
first day of 4th grade. This kind gesture will set the tone before even the first day of school!
The nautical thank you cards will be used throughout the year by the students as they create thank
you cards to their team groups.\r\n\r\nYour generous donations will help me to help make our
classroom a fun, inviting, learning environment from day one.\r\n\r\nIt costs lost of money out of
my own pocket on resources to get our classroom ready. Please consider helping with this project t
o make our new school year a very successful one. Thank you!nannan
==========
\"Creative Greatness\" is this school year's mantra to inspire my students to reach for the stars.
I'm excited about ushering in an enthusiasm and passion for growth in the visual arts department a
nd inspiring students to consider and apply the purpose of art outside of the classroom.
\r\n\r\nMy art students and art club members are not just \"taking\" art class, but are using thei
r creativity to engage in school-wide beautification projects and community initiatives. Help us t
o explore a greater variety of art media and technology in my Art 1 classes to ignite student's i
nterest in furthering their studies in art. Our large student body limits funding to the arts, so
charitable donations are crucial to our growth into Advanced Placement and College and Career Read
iness programs in the arts.Our class will create personalized and unique interactive notebooks to
encourage the development of independent learners and writers. Interactive notebooks are not just
used for class notes, but also for daily learning activities that require students to process the
information presented in class and then organize the content in a manner that will reinforce their
learning. \r\nInteractive Notebooks are a cross curricular tool that supports literacy in all cont
ent areas. In our art class, these notebooks are used not only as an affordable sketchbook option,
but also as an \"all things art\" guide that students can continue to reference throughout the sch
ool year and as they continue studies of more advanced art courses. We use our interactive noteboo
ks to write art critiques in response to viewing the works of famous artists and to write art stat
ements in response to the student's personal artwork. We also use interactive notebooks to build v
ocabulary skills with engaging activities to learn about the elements and principles of art to go
far beyond just defining the terms. Students are required to chose thinking maps that best organi
ze the information presented in the lesson to teach lifelong skills of literacy and note-taking. \
r\nStudents' interest in using interactive notebooks is positively impacted when they are able to
be creative and personalize the look of their notebooks. Engagement will no doubt be dramatically
increased with fun and colorful notebook covers and pages for each lesson. With this note-taking p

increased with fun and colorful notebook covers and pages for each lesson. With this note-taking p
rocess, students will learn organization, color coding, summarizing, and other important skills wh
ile creating personalized portfolios of their individual learning that they can reference
throughout the year.nannan
==

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[9999])
print(sent)
print("="*50)
```

\"Creative Greatness\" is this school year is mantra to inspire my students to reach for the stars
. I am excited about ushering in an enthusiasm and passion for growth in the visual arts
department and inspiring students to consider and apply the purpose of art outside of the classroo
m. \r\n\r\nMy art students and art club members are not just \"taking\" art class, but are using t
heir creativity to engage in school-wide beautification projects and community initiatives. Help u
s to explore a greater variety of art media and technology in my Art 1 classes to ignite student
is interest in furthering their studies in art. Our large student body limits funding to the arts,
so charitable donations are crucial to our growth into Advanced Placement and College and Career R
eadiness programs in the arts.Our class will create personalized and unique interactive notebooks
to encourage the development of independent learners and writers. Interactive notebooks are not ju
st used for class notes, but also for daily learning activities that require students to process t
he information presented in class and then organize the content in a manner that will reinforce th
eir learning. \r\nInteractive Notebooks are a cross curricular tool that supports literacy in all
content areas. In our art class, these notebooks are used not only as an affordable sketchbook opt
ion, but also as an \"all things art\" guide that students can continue to reference throughout th
e school year and as they continue studies of more advanced art courses. We use our interactive no
tebooks to write art critiques in response to viewing the works of famous artists and to write art
statements in response to the student is personal artwork. We also use interactive notebooks to bu
ild vocabulary skills with engaging activities to learn about the elements and principles of art t
o go far beyond just defining the terms. Students are required to chose thinking maps that best o
rganize the information presented in the lesson to teach lifelong skills of literacy and note-taki
ng. \r\nStudents' interest in using interactive notebooks is positively impacted when they are abl
e to be creative and personalize the look of their notebooks. Engagement will no doubt be
dramatically increased with fun and colorful notebook covers and pages for each lesson. With this
note-taking process, students will learn organization, color coding, summarizing, and other
important skills while creating personalized portfolios of their individual learning that they can
reference throughout the year.nannan
==

In [16]:

```
# \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)
```

 Creative Greatness is this school year is mantra to inspire my students to reach for the stars.
I am excited about ushering in an enthusiasm and passion for growth in the visual arts department
and inspiring students to consider and apply the purpose of art outside of the classroom.   My a

```
rt students and art club members are not just  taking  art class, but are using their creativity t
o engage in school-wide beautification projects and community initiatives. Help us to explore a gr
eater variety of art media and technology in my Art 1  classes to ignite student is interest in fu
rthering their studies in art. Our large student body limits funding to the arts, so charitable
donations are crucial to our growth into Advanced Placement and College and Career Readiness progr
ams in the arts.Our class will create personalized and unique interactive notebooks to encourage t
he development of independent learners and writers. Interactive notebooks are not just used for cl
ass notes, but also for daily learning activities that require students to process the information
presented in class and then organize the content in a manner that will reinforce their learning.
Interactive Notebooks are a cross curricular tool that supports literacy in all content areas. In
our art class, these notebooks are used not only as an affordable sketchbook option, but also as a
n  all things art  guide that students can continue to reference throughout the school year and as
they continue studies of more advanced art courses. We use our interactive notebooks to write art
critiques in response to viewing the works of famous artists and to write art statements in
response to the student is personal artwork. We also use interactive notebooks to build vocabulary
skills with engaging activities to learn about the elements and principles of art to go far beyond
just defining the terms.  Students are required to chose thinking maps that best organize the info
rmation presented in the lesson to teach lifelong skills of literacy and note-taking.   Students'
interest in using interactive notebooks is positively impacted when they are able to be creative a
nd personalize the look of their notebooks. Engagement will no doubt be dramatically increased
with fun and colorful notebook covers and pages for each lesson. With this note-taking process, st
udents will learn organization, color coding, summarizing, and other important skills while
creating personalized portfolios of their individual learning that they can reference throughout t
he year.nannan
```

In [17]:

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

```
 Creative Greatness is this school year is mantra to inspire my students to reach for the stars I
am excited about ushering in an enthusiasm and passion for growth in the visual arts department an
d inspiring students to consider and apply the purpose of art outside of the classroom My art stud
ents and art club members are not just taking art class but are using their creativity to engage i
n school wide beautification projects and community initiatives Help us to explore a greater varie
ty of art media and technology in my Art 1 classes to ignite student is interest in furthering the
ir studies in art Our large student body limits funding to the arts so charitable donations are cr
ucial to our growth into Advanced Placement and College and Career Readiness programs in the arts
Our class will create personalized and unique interactive notebooks to encourage the development o
f independent learners and writers Interactive notebooks are not just used for class notes but als
o for daily learning activities that require students to process the information presented in clas
s and then organize the content in a manner that will reinforce their learning Interactive Noteboo
ks are a cross curricular tool that supports literacy in all content areas In our art class these
notebooks are used not only as an affordable sketchbook option but also as an all things art guide
that students can continue to reference throughout the school year and as they continue studies of
more advanced art courses We use our interactive notebooks to write art critiques in response to v
iewing the works of famous artists and to write art statements in response to the student is
personal artwork We also use interactive notebooks to build vocabulary skills with engaging
activities to learn about the elements and principles of art to go far beyond just defining the te
rms Students are required to chose thinking maps that best organize the information presented in t
he lesson to teach lifelong skills of literacy and note taking Students interest in using interact
ive notebooks is positively impacted when they are able to be creative and personalize the look of
their notebooks Engagement will no doubt be dramatically increased with fun and colorful notebook
covers and pages for each lesson With this note taking process students will learn organization co
lor coding summarizing and other important skills while creating personalized portfolios of their
individual learning that they can reference throughout the year nannan
```

In [18]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've",
\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his',
'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them',
'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll",
'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having',
'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', '
while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',
```

```
'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',
'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under'
, 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'e
ach', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll'
, 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "d
esn'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 [19]:

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

```
100%|████████████████████████████████████████████████████████████████| 10000/10000
[00:09<00:00, 1092.64it/s]
```

In [20]:

```
# after preprocesing
preprocessed_essays[9999]
```

Out[20]:

```
'creative greatness school year mantra inspire students reach stars excited ushering enthusiasm pa
ssion growth visual arts department inspiring students consider apply purpose art outside
classroom art students art club members not taking art class using creativity engage school wide b
eautification projects community initiatives help us explore greater variety art media technology
art 1 classes ignite student interest furthering studies art large student body limits funding art
s charitable donations crucial growth advanced placement college career readiness programs arts cl
ass create personalized unique interactive notebooks encourage development independent learners wr
iters interactive notebooks not used class notes also daily learning activities require students p
rocess information presented class organize content manner reinforce learning interactive
notebooks cross curricular tool supports literacy content areas art class notebooks used not affor
dable sketchbook option also things art guide students continue reference throughout school year c
ontinue studies advanced art courses use interactive notebooks write art critiques response viewin
g works famous artists write art statements response student personal artwork also use interactive
notebooks build vocabulary skills engaging activities learn elements principles art go far beyond
defining terms students required chose thinking maps best organize information presented lesson te
ach lifelong skills literacy note taking students interest using interactive notebooks positively
impacted able creative personalize look notebooks engagement no doubt dramatically increased fun c
olorful notebook covers pages lesson note taking process students learn organization color coding
summarizing important skills creating personalized portfolios individual learning reference
throughout year nannan'
```

In [21]:

```
project_data['preprocessed_essays'] = preprocessed_essays
project_data.drop(['essay'], axis=1, inplace=True)
project_data.head(2)
```

Out[21]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_grade_cate
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	Grades P
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Grade

3.3. Preprocessing Title text

In [22]:

```
# printing some random essays.
print(project_data['project_title'].values[0])
print("="*50)
print(project_data['project_title'].values[150])
print("="*50)
print(project_data['project_title'].values[1000])
print("="*50)
print(project_data['project_title'].values[5000])
print("="*50)
print(project_data['project_title'].values[9999])
print("="*50)
```

```
Educational Support for English Learners at Home
==================================================
More Movement with Hokki Stools
==================================================
Sailing Into a Super 4th Grade Year
==================================================
Bouncing Our Wiggles and Worries Away!
==================================================
Note Your Ordinary Notebook!
==================================================
```

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]:

```
title = decontracted(project_data['project_title'].values[9999])
print(title)
print("="*50)
```

```
Note Your Ordinary Notebook!
==================================================
```

In [25]:

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

```
Note Your Ordinary Notebook!
```

In [26]:

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

```
Note Your Ordinary Notebook
```

In [27]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've",
\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his',
'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them',
'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll",
'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having',
'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', '
while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',
'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under'
, 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'e
ach', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll'
, 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "d
esn'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 [28]:

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

```
100%|██████████████████████████████████████████████████████████████████████████| 10000/10000
[00:00<00:00, 21056.01it/s]
```

In [29]:

```
# after preprocesing
preprocessed_titles[9999]
```

Out[29]:

```
'note ordinary notebook'
```

In [30]:

```
project_data['preprocessed_titles'] = preprocessed_titles
project_data.drop(['project_title'], axis=1, inplace=True)
project_data.head(2)
```

Out[30]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_grade_cate
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	Grades P
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Grade

In [31]:

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

Out[31]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_grade_cate
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	Grades P
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Grade

4. Preprocessing of Categorical data

4.1. Preprocessing project_grade_category

In [32]:

```
project_grade_clean_category = []

for i in range(len(project_data)):
    a = project_data["project_grade_category"][i].replace(" ", "_").replace("-", "_")
    project_grade_clean_category.append(a)
```

In [33]:

```
project_grade_clean_category[0:5]
```

Out[33]:

```
['Grades_PreK_2', 'Grades_6_8', 'Grades_6_8', 'Grades_PreK_2', 'Grades_PreK_2']
```

In [34]:

```
project_data['project_grade_clean_category'] = project_grade_clean_category
```

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

Out[34]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_subject_ca
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	Literacy & L
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	History & Civics,

4.2. Preprocessing project_subject_categories

In [35]:

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

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

In [36]:

```
cat_list[0:5]
```

Out[36]:

```
['Literacy_Language',
 'History_Civics Health_Sports',
 'Health_Sports',
 'Literacy_Language Math_Science',
 'Math_Science']
```

In [37]:

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

Out[37]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_subject_su
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Civics & Gover

Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_subject_su

4.3. Preprocessing project_subject_subcategories

In [38]:

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

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

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

In [39]:

```
sub_cat_list[0:5]
```

Out[39]:

```
['ESL Literacy',
 'Civics_Government TeamSports',
 'Health_Wellness TeamSports',
 'Literacy Mathematics',
 'Mathematics']
```

In [40]:

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

Out[40]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_essay_1
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	My students are English learners that are work...
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Our students arrive to our school eager to lea...

In [41]:

```
project_data.head()
```

Out[41]:

Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_essay_1

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_essay_1
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	My students are English learners that are work...
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Our students arrive to our school eager to lea...
2	21895	p182444	3465aaf82da834c0582ebd0ef8040ca0	Ms.	AZ	2016-08-31 12:03:56	\r\n\"True champions aren't always the ones th...
3	45	p246581	f3cb9bffbba169bef1a77b243e620b60	Mrs.	KY	2016-10-06 21:16:17	I work at a unique school filled with both ESL...
4	172407	p104768	be1f7507a41f8479dc06f047086a39ec	Mrs.	TX	2016-07-11 01:10:09	Our second grade classroom next year will be m...

5. Counting number of words in the combine essays

In [42]:

```
essay_count = []
for word in project_data['preprocessed_essays']:
    a = len(word.split())
    b = str(a)
    essay_count.append(b)
```

In [43]:

```
essay_count[9999]
```

Out[43]:

```
'236'
```

In [44]:

```
project_data['number_of_words_in_essays'] = essay_count
project_data.head(2)
```

Out[44]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_essay_1
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	My students are English learners that are work...
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Our students arrive to our school eager to lea...

2 rows × 21 columns

6. Counting number of words in the title

In [45]:

```
title_count = []
for word in project_data['preprocessed_titles']:
    a = len(word.split())
    b = str(a)
    title_count.append(b)
```

In [46]:

```
title_count[9999]
```

Out[46]:

```
'3'
```

In [47]:

```
project_data['number_of_words_in_the_title'] = title_count
project_data.head(2)
```

Out[47]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_submitted_datetime	project_essay_1
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:43:57	My students are English learners that are work...
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:22:10	Our students arrive to our school eager to lea...

2 rows × 22 columns

7. Splitting data

In [48]:

```
Xproject_data= project_data
print(Xproject_data.shape)
Yproject_data = Xproject_data['project_is_approved']
print(Yproject_data.shape)
Xproject_data.drop(['project_is_approved'], axis=1, inplace=True)
print(Xproject_data.shape)
```

```
(10000, 22)
(10000,)
(10000, 21)
```

8. Encoding Categorigal data

8.1. One Hot Encoding of clean_categories

In [49]:

```
# we use count vectorizer to convert the values into one hot encoded features
from sklearn.feature_extraction.text import CountVectorizer
vectorizer1 = CountVectorizer(lowercase=False, binary=True)
vectorizer1.fit(Xproject_data['clean_categories'].values)
print(vectorizer1.get_feature_names())


categories_one_hot_encode = vectorizer1.transform(Xproject_data['clean_categories'].values)
print("Shape of matrix after one hot encodig ",categories_one_hot_encode.shape)
```

```
['AppliedLearning', 'Care_Hunger', 'Health_Sports', 'History_Civics', 'Literacy_Language',
'Math_Science', 'Music_Arts', 'SpecialNeeds', 'Warmth']
Shape of matrix after one hot encodig  (10000, 9)
```

8.2. One Hot Encoding of clean_subcategories

In [50]:

```
# we use count vectorizer to convert the values into one hot encoded features
vectorizer2 = CountVectorizer(lowercase=False, binary=True)
vectorizer2.fit(Xproject_data['clean_subcategories'].values)
print(vectorizer2.get_feature_names())


sub_categories_one_hot_encode = vectorizer2.transform(Xproject_data['clean_subcategories'].values)


print("Shape of matrix after one hot encoding ",sub_categories_one_hot_encode.shape)
```

```
['AppliedSciences', 'Care_Hunger', 'CharacterEducation', 'Civics_Government',
'College_CareerPrep', 'CommunityService', 'ESL', 'EarlyDevelopment', 'Economics',
'EnvironmentalScience', 'Extracurricular', 'FinancialLiteracy', 'ForeignLanguages', 'Gym_Fitness',
'Health_LifeScience', 'Health_Wellness', 'History_Geography', 'Literacy', 'Literature_Writing', 'M
athematics', 'Music', 'NutritionEducation', 'Other', 'ParentInvolvement', 'PerformingArts', 'Socia
lSciences', 'SpecialNeeds', 'TeamSports', 'VisualArts', 'Warmth']
Shape of matrix after one hot encoding  (10000, 30)
```

8.3. One Hot Encoding of school_state

In [51]:

```
# we use count vectorizer to convert the values into one hot encoded features
vectorizer3 = CountVectorizer(lowercase=False, binary=True)
vectorizer3.fit(Xproject_data['school_state'].values)
print(vectorizer3.get_feature_names())


school_state_one_hot_encode = vectorizer3.transform(Xproject_data['school_state'].values)


print("Shape of matrix after one hot encoding ",school_state_one_hot_encode.shape)
```

```
['AK', 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA', 'HI', 'IA', 'ID', 'IL', 'IN', 'K
S', '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 encoding  (10000, 51)
```

8.4. One Hot Encoding of teacher_prefix

In [52]:

```
# we use count vectorizer to convert the values into one hot encoded features
vectorizer4 = CountVectorizer(lowercase=False, binary=True)
vectorizer4.fit(Xproject_data['teacher_prefix'].values)
print(vectorizer4.get_feature_names())

teacher_prefix_one_hot_encode = vectorizer4.transform(Xproject_data['teacher_prefix'].values)


print("Shape of matrix after one hot encoding ",teacher_prefix_one_hot_encode.shape)
```

```
['Mr', 'Mrs', 'Ms', 'Teacher']
Shape of matrix after one hot encoding  (10000, 4)
```

8.5. One Hot Encoding of project_grade_clean_category

In [53]:

```
# we use count vectorizer to convert the values into one hot encoded features
```

```
vectorizer5 = CountVectorizer(lowercase=False, binary=True)
vectorizer5.fit(Xproject_data['project_grade_clean_category'].values)
print(vectorizer5.get_feature_names())


grade_one_hot_encode = vectorizer5.transform(Xproject_data['project_grade_clean_category'].values)

print("Shape of matrix after one hot encoding ",grade_one_hot_encode.shape)
```

```
['Grades_3_5', 'Grades_6_8', 'Grades_9_12', 'Grades_PreK_2']
Shape of matrix after one hot encoding  (10000, 4)
```

9. Encoding of Text Data

9.1. BOW Encoding on preprocessed_essays

In [54]:

```
# We are considering only the words which appeared in at least 10 documents(rows or projects).
vectorizer6 = CountVectorizer(min_df=10)
text_bow = vectorizer6.fit_transform(Xproject_data['preprocessed_essays'].values)
print("Shape of matrix after one hot encodig ",text_bow.shape)
```

```
Shape of matrix after one hot encodig  (10000, 6127)
```

9.2. BOW Encoding on preprocessed_titles

In [55]:

```
# We are considering only the words which appeared in at least 10 documents(rows or projects).
vectorizer7 = CountVectorizer(min_df=10)
title_bow = vectorizer7.fit_transform(Xproject_data['preprocessed_titles'].values)
print("Shape of matrix after one hot encodig ",title_bow.shape)
```

```
Shape of matrix after one hot encodig  (10000, 607)
```

10. Encoding of Numerical Data

10.1. Encoding of price data

In [58]:

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


scalar = MinMaxScaler()

price_standardized = scalar.fit_transform(Xproject_data['price'].values.reshape(-1, 1))
print(price_standardized.shape)
```

```
(10000, 1)
```

In [59]:

```
price_standardized
```

Out[59]:

```
array([[0.01531974],
```

```
       [0.02976326],
       [0.05155358],
       ...,
       [0.03520959],
       [0.03266397],
       [0.02109015]])
```

10.2. Encoding of quantity data

In [60]:

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

scalar = MinMaxScaler()

quantity_standardized = scalar.fit_transform(Xproject_data['quantity'].values.reshape(-1, 1))
print(quantity_standardized.shape)
```

```
(10000, 1)
```

In [61]:

```
quantity_standardized
```

Out[61]:

```
array([[0.03793103],
       [0.        ],
       [0.0362069 ],
       ...,
       [0.02758621],
       [0.00172414],
       [0.06034483]])
```

10.3. Encoding of teacher_number_of_previously_posted_projects data

In [62]:

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

scalar = MinMaxScaler()

# Now standardize the data with above maen and variance.
teacher_number_of_previously_posted_projects_standardized =
scalar.fit_transform(Xproject_data['teacher_number_of_previously_posted_projects'].values.reshape(
-1, 1))
print(teacher_number_of_previously_posted_projects_standardized.shape)
```

```
(10000, 1)
```

In [63]:

```
teacher_number_of_previously_posted_projects_standardized
```

Out[63]:

```
array([[0.        ],
       [0.01851852],
       [0.0026455 ],
       ...,
       [0.01058201],
       [0.01587302],
```

```
       [0.01322751]])
```

10.4. Encoding of number_of_words_in_essays data

In [64]:

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

scalar = MinMaxScaler()

number_of_words_in_essays_standardized =
scalar.fit_transform(Xproject_data['number_of_words_in_essays'].values.reshape(-1, 1))
print(number_of_words_in_essays_standardized.shape)
```

```
(10000, 1)
```

In [65]:

```
number_of_words_in_essays_standardized
```

Out[65]:

```
array([[0.33617021],
       [0.12340426],
       [0.50212766],
       ...,
       [0.37021277],
       [0.17446809],
       [0.70638298]])
```

10.5. Encoding of number_of_words_in_the_title data

In [66]:

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

scalar = MinMaxScaler()

number_of_words_in_the_title_standardized =
scalar.fit_transform(Xproject_data['number_of_words_in_the_title'].values.reshape(-1, 1))
print(number_of_words_in_the_title_standardized.shape)
```

```
(10000, 1)
```

In [67]:

```
number_of_words_in_the_title_standardized
```

Out[67]:

```
array([[0.45454545],
       [0.36363636],
       [0.54545455],
       ...,
       [0.45454545],
       [0.18181818],
       [0.27272727]])
```

11. Printing Dimensions of all Preprocessed Data

In [68]:

```
print(categories_one_hot_encode.shape)
print(sub_categories_one_hot_encode.shape)
print(school_state_one_hot_encode.shape)
print(teacher_prefix_one_hot_encode.shape)
print(grade_one_hot_encode.shape)
print(text_bow.shape)
print(title_bow.shape)
print(price_standardized.shape)
print(quantity_standardized.shape)
print(teacher_number_of_previously_posted_projects_standardized.shape)
print(number_of_words_in_essays_standardized.shape)
print(number_of_words_in_the_title_standardized.shape)
```

```
(10000, 9)
(10000, 30)
(10000, 51)
(10000, 4)
(10000, 4)
(10000, 6127)
(10000, 607)
(10000, 6127)
(10000, 607)
(10000, 1)
(10000, 1)
(10000, 1)
(10000, 1)
(10000, 1)
```

Set 1: categorical, numerical features + project_title(BOW) +preprocessed_eassay

In [69]:

```
from scipy.sparse import hstack
# with the same hstack function we are concatinating a sparse matrix and a dense matirx :)
Xset1 =
hstack((categories_one_hot_encode,sub_categories_one_hot_encode,school_state_one_hot_encode,teache
r_prefix_one_hot_encode,grade_one_hot_encode,price_standardized,quantity_standardized,teacher_numbe
r_of_previously_posted_projects_standardized,text_bow,title_bow)).tocsr()

print(Xset1.shape,Yproject_data.shape)
```

```
(10000, 6835) (10000,)
```

In [70]:

```
type(Xset1)
```

Out[70]:

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

12. Selecting top 5K features

In [72]:

```
from sklearn.feature_selection import SelectKBest, chi2
selector1 =  SelectKBest(chi2, k=5000)
X_set1_new = selector1.fit_transform(Xset1, Yproject_data)
```

In [73]:

```
selector1.get_support(indices=True)
```

Out[73]:

```
array([   0,    1,    3, ..., 6831, 6832, 6833], dtype=int64)
```

In [74]:

```
X_set1_new_dataframe = pd.DataFrame(Xset1.toarray())
```

In [75]:

```
X_set1_new_dataframe.shape
```

Out[75]:

```
(10000, 6835)
```

In [76]:

```
# https://stackoverflow.com/questions/36967666/transform-scipy-sparse-csr-to-pandas
names = X_set1_new_dataframe.columns.values[selector1.get_support()]
scores = selector1.scores_[selector1.get_support()]
names_scores = list(zip(names, scores))
ns_df = pd.DataFrame(data = names_scores, columns=['Feature_names', 'Feature_Scores'])
#Sort the dataframe for better visualization
ns_df_sorted = ns_df.sort_values(['Feature_Scores', 'Feature_names'], ascending = [False, True])
ns_df_sorted.head()
```

Out[76]:

	Feature_names	Feature_Scores
2583	3583	136.413734
576	805	130.505981
3948	5401	111.390002
788	1105	65.114507
3344	4595	64.766623

In [77]:

```
print(ns_df_sorted.shape)
```

```
(5000, 2)
```

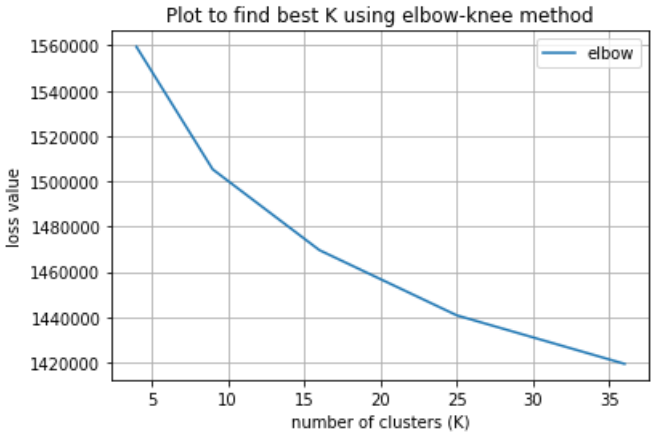
13. Kmeans

In [80]:

```
from sklearn.cluster import KMeans
def getLosses(hyperparameter, data_matrix):
    loss = []

    for k in K:
        kmeans = KMeans(n_clusters=k, random_state=0, n_jobs=-1).fit(data_matrix)
        loss.append(kmeans.inertia_)

    return loss

def plotGraph(hyperparameter, loss):
    plt.plot(hyperparameter, loss, label='elbow','-+')

    plt.title("Plot to find best K using elbow-knee method")
    plt.xlabel('number of clusters (K)')
    plt.ylabel('loss value')
    plt.legend()
    plt.grid(True)
```

In [81]:

```
K = [4, 9, 16, 25, 36]
loss_value = getLosses(K, X_set1_new)
plotGraph(K, loss_value)
```



14. Fitting the kmeans with selected best K(n_clusters)

In [82]:

```
kmeans = KMeans(n_clusters = 16, random_state=0, n_jobs=-1).fit(X_set1_new)
```

In [83]:

```
FN = project_data['preprocessed_essays'].values
```

In [89]:

```
# Getting all the reviews in different clusters
cluster1 = []
cluster2 = []
cluster3 = []
cluster4 = []
cluster5 = []
cluster6 = []
cluster7 = []
cluster8 = []
cluster9 = []
cluster10 = []
cluster11 = []
cluster12 = []
cluster13 = []
cluster14 = []
cluster15 = []
cluster16 = []


for i in range(kmeans.labels_.shape[0]):
    if kmeans.labels_[i] == 0:
        cluster1.append(FN[i])
    elif kmeans.labels_[i] == 1:
        cluster2.append(FN[i])
    elif kmeans.labels_[i] == 2:
        cluster3.append(FN[i])
    elif kmeans.labels_[i] == 3:
        cluster4.append(FN[i])
    elif kmeans.labels_[i] == 4:
        cluster5.append(FN[i])
    elif kmeans.labels_[i] == 5:
        cluster6.append(FN[i])
    elif kmeans.labels_[i] == 6:
        cluster7.append(FN[i])
    elif kmeans.labels_[i] == 7:
        cluster8.append(FN[i])
    elif kmeans.labels_[i] == 8:
```

```
        cluster9.append(FN[i])
    elif kmeans.labels_[i] == 9:
        cluster10.append(FN[i])
    elif kmeans.labels_[i] == 10:
        cluster11.append(FN[i])
    elif kmeans.labels_[i] == 11:
        cluster12.append(FN[i])
    elif kmeans.labels_[i] == 12:
        cluster13.append(FN[i])
    elif kmeans.labels_[i] == 13:
        cluster14.append(FN[i])
    elif kmeans.labels_[i] == 14:
        cluster15.append(FN[i])

    else :
        cluster16.append(FN[i])

# Number of reviews in different clusters
print("\nNo. of essay in Cluster-1  : ",len(cluster1))
print("\nNo. of essay in Cluster-2  : ",len(cluster2))
print("\nNo. of essay in Cluster-3  : ",len(cluster3))
print("\nNo. of essay in Cluster-4  : ",len(cluster4))
print("\nNo. of essay in Cluster-5  : ",len(cluster5))
print("\nNo. of essay in Cluster-6  : ",len(cluster6))
print("\nNo. of essay in Cluster-7  : ",len(cluster7))
print("\nNo. of essay in Cluster-8  : ",len(cluster8))
print("\nNo. of essay in Cluster-9  : ",len(cluster9))
print("\nNo. of essay in Cluster-10 : ",len(cluster10))
print("\nNo. of essay in Cluster-11 : ",len(cluster11))
print("\nNo. of essay in Cluster-12 : ",len(cluster12))
print("\nNo. of essay in Cluster-13 : ",len(cluster13))
print("\nNo. of essay in Cluster-14 : ",len(cluster14))
print("\nNo. of essay in Cluster-15 : ",len(cluster15))
print("\nNo. of essay in Cluster-16 : ",len(cluster16))
```

```
No. of essay in Cluster-1  :  289

No. of essay in Cluster-2  :  677

No. of essay in Cluster-3  :  598

No. of essay in Cluster-4  :  510

No. of essay in Cluster-5  :  684

No. of essay in Cluster-6  :  279

No. of essay in Cluster-7  :  1096

No. of essay in Cluster-8  :  168

No. of essay in Cluster-9  :  883

No. of essay in Cluster-10 :  1

No. of essay in Cluster-11 :  1648

No. of essay in Cluster-12 :  257

No. of essay in Cluster-13 :  1436

No. of essay in Cluster-14 :  542

No. of essay in Cluster-15 :  380

No. of essay in Cluster-16 :  552
```

In [90]:

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
stopwords = set(STOPWORDS)

def show_wordcloud(data, title = None):
    wordcloud = WordCloud(
        background_color='white',
```

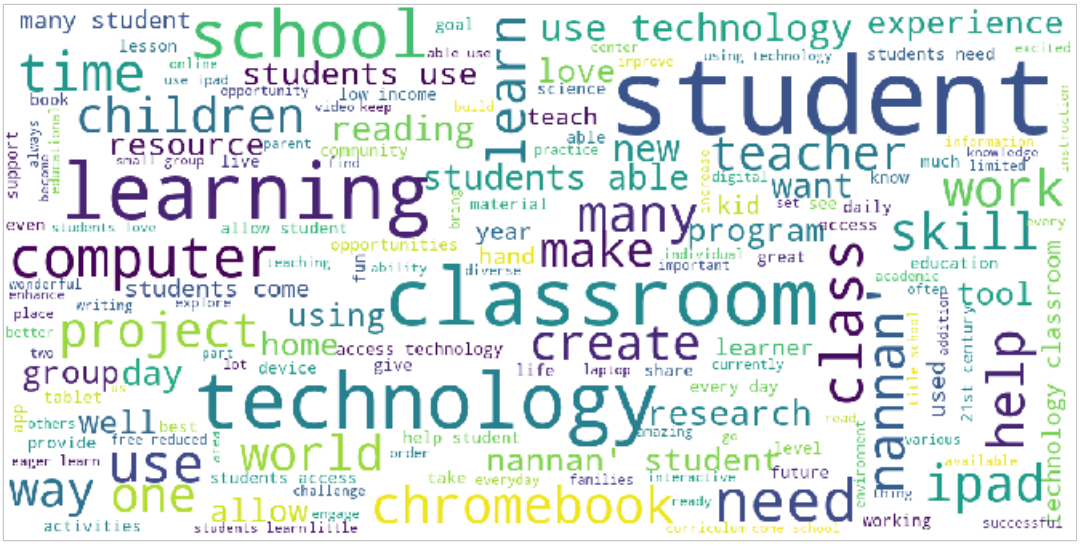
```
        stopwords=stopwords,
        max_words=200,
        max_font_size=40,
        scale=3,
        random_state=1 # chosen at random by flipping a coin; it was heads
    ).generate(str(data))

    fig = plt.figure(1, figsize=(12, 12))
    plt.axis('off')


    if title:
        fig.suptitle(title, fontsize=20)
        fig.subplots_adjust(top=2.3)

    plt.imshow(wordcloud)
    plt.show()
```

In [91]:

```
show_wordcloud(cluster1)
```

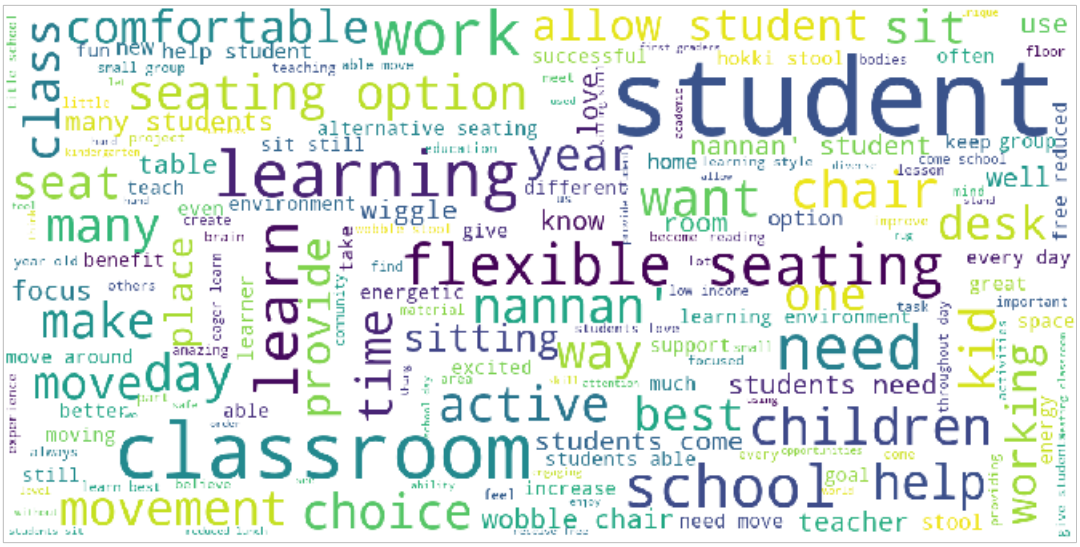
In [92]:

```
show_wordcloud(cluster2)
```



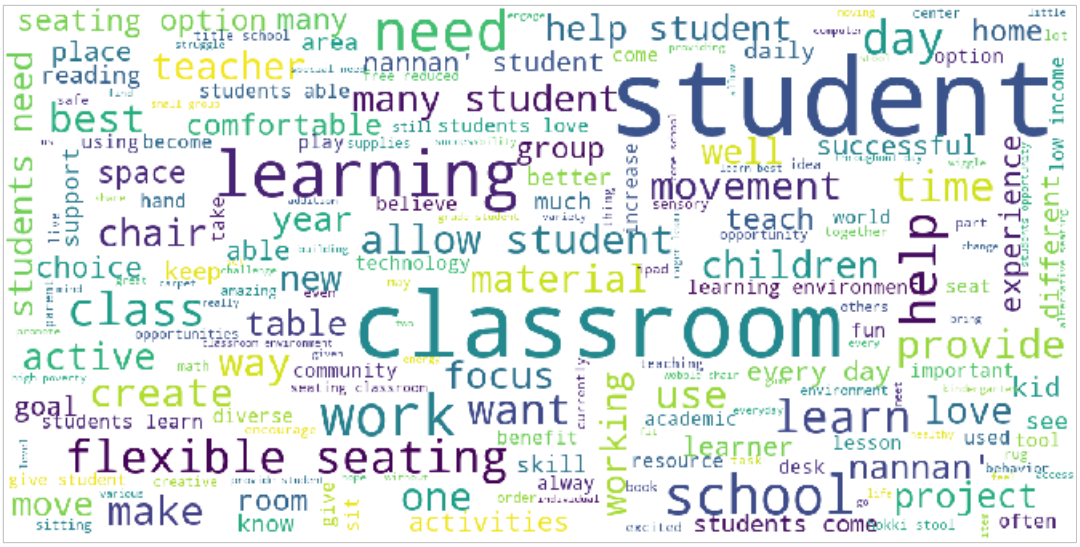
In [93]:

```
show_wordcloud(cluster3)
```



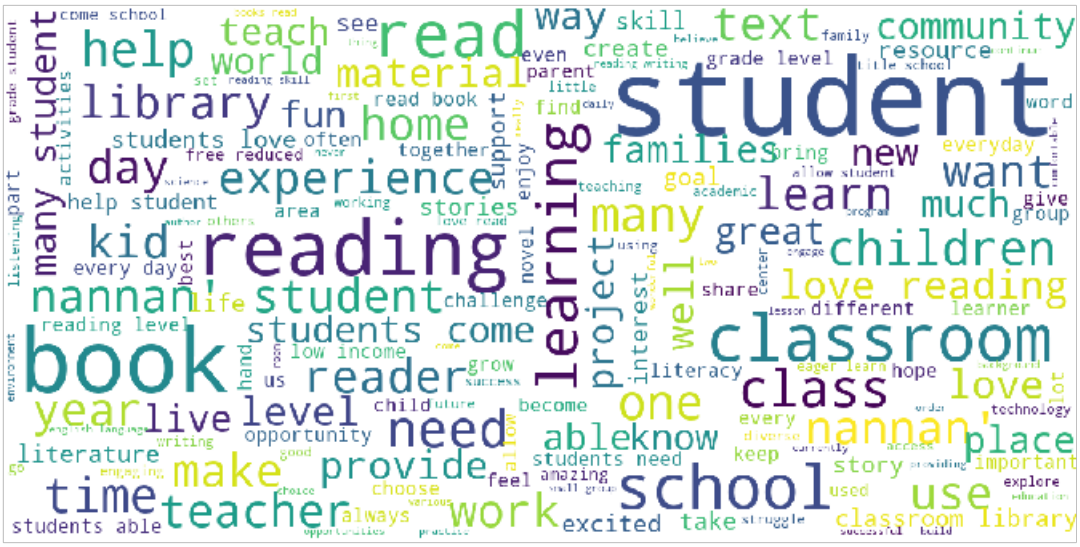
In [94]:

```
show_wordcloud(cluster4)
```



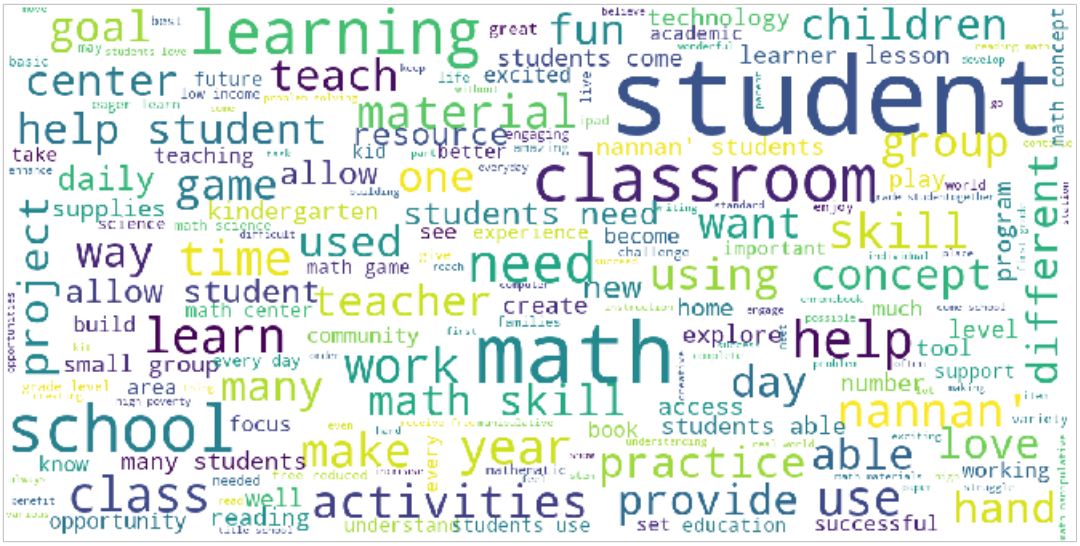
In [95]:

```
show_wordcloud(cluster5)
```



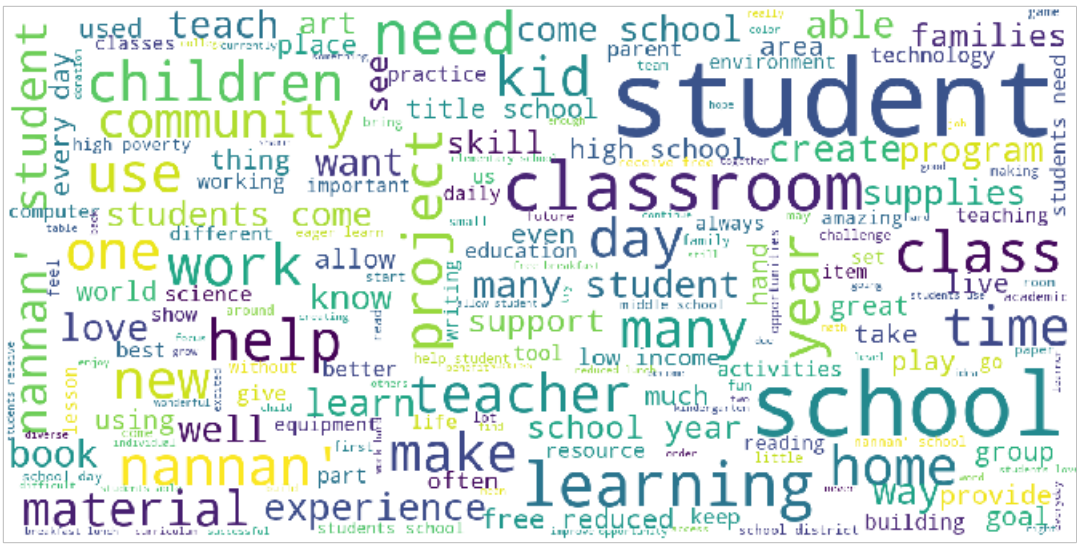
In [96]:

```
show_wordcloud(cluster6)
```



In [97]:

```
show_wordcloud(cluster7)
```

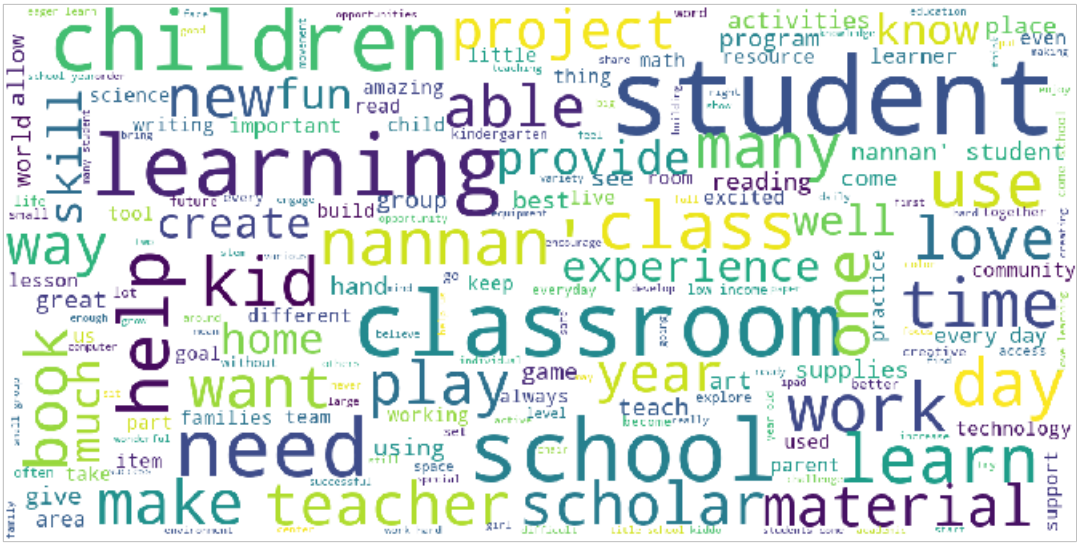


In [98]:

```
show_wordcloud(cluster8)
```

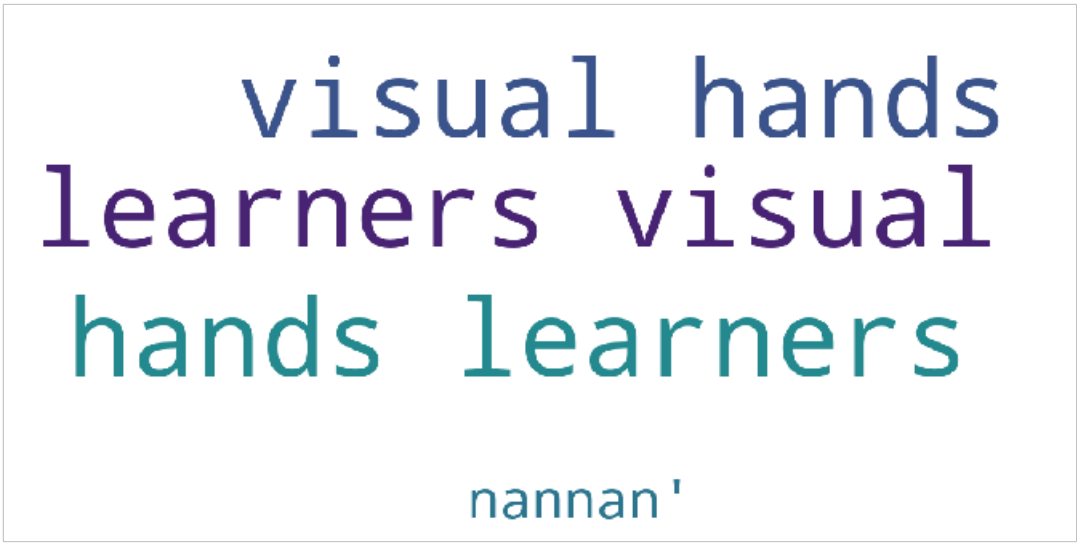
In [99]:

```
show_wordcloud(cluster9)
```



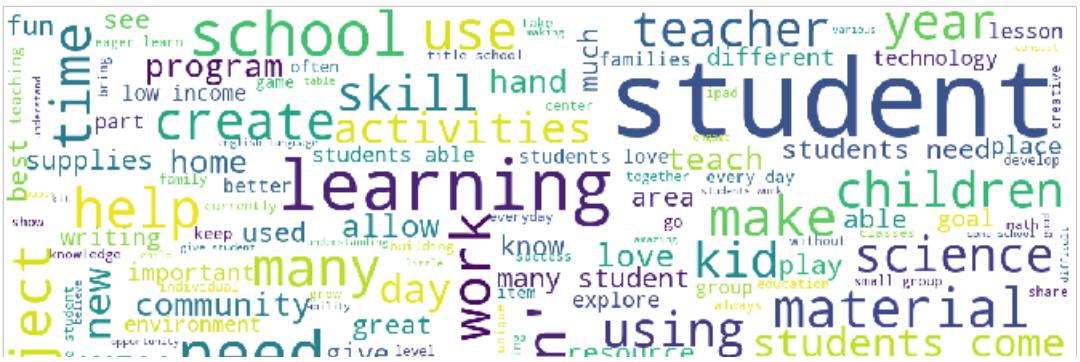
In [100]:

```
show_wordcloud(cluster10)
```



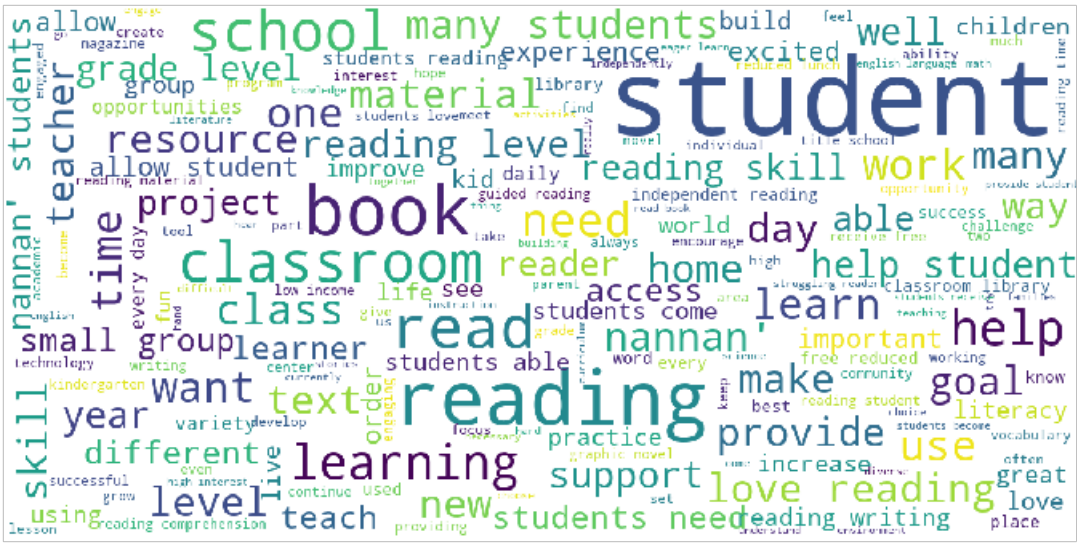
In [101]:

```
show_wordcloud(cluster11)
```



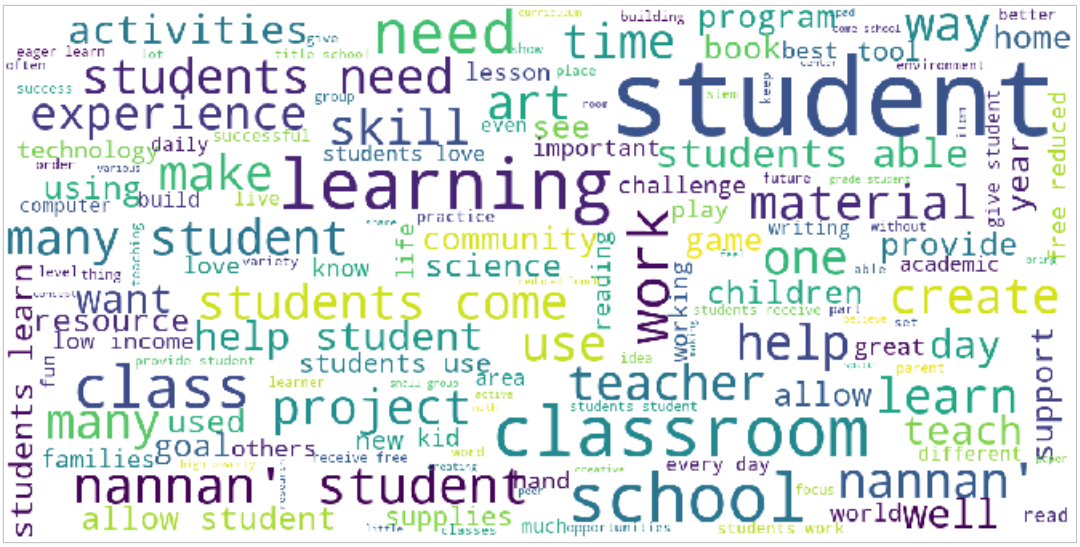
In [102]:

```
show_wordcloud(cluster12)
```



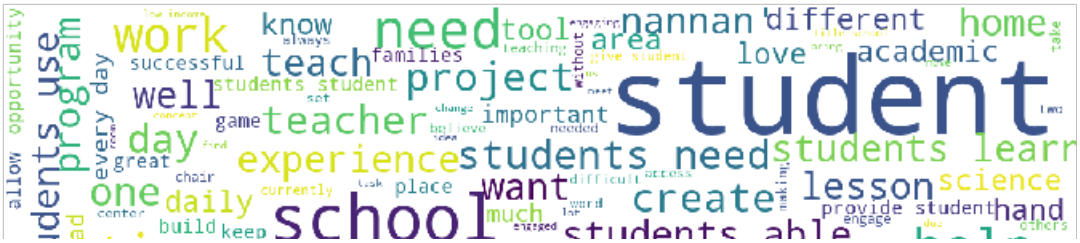
In [103]:

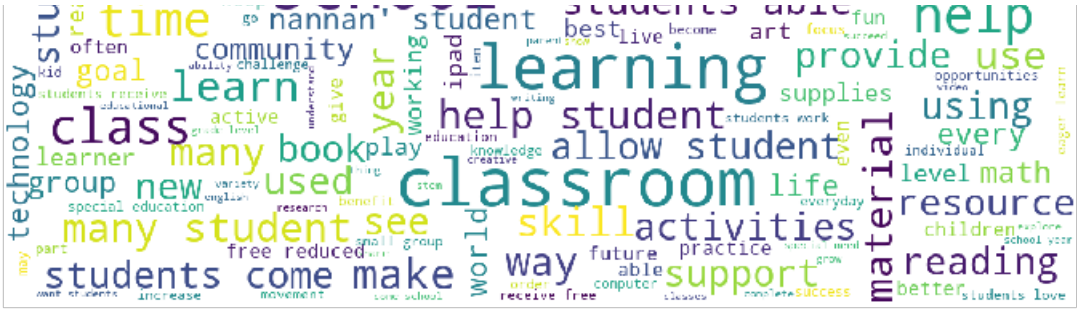
```
show_wordcloud(cluster13)
```



In [104]:

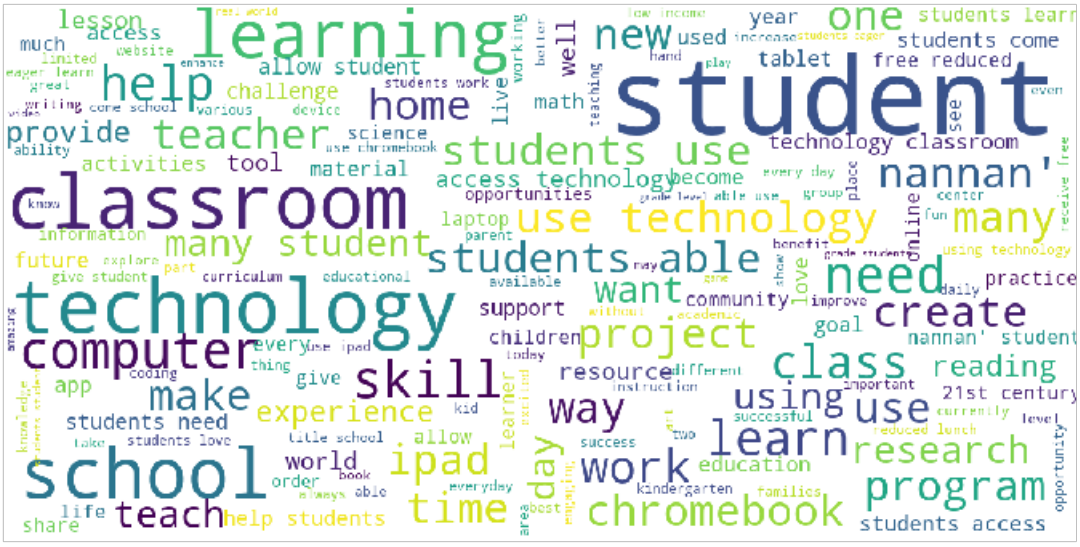
```
show_wordcloud(cluster14)
```





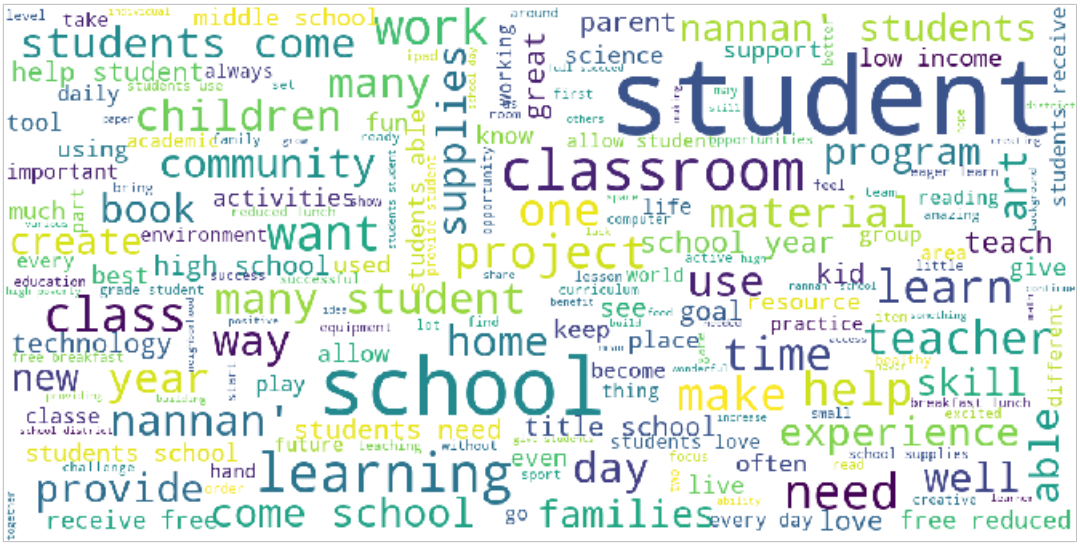
In [105]:

```
show_wordcloud(cluster15)
```



In [106]:

```
show_wordcloud(cluster16)
```

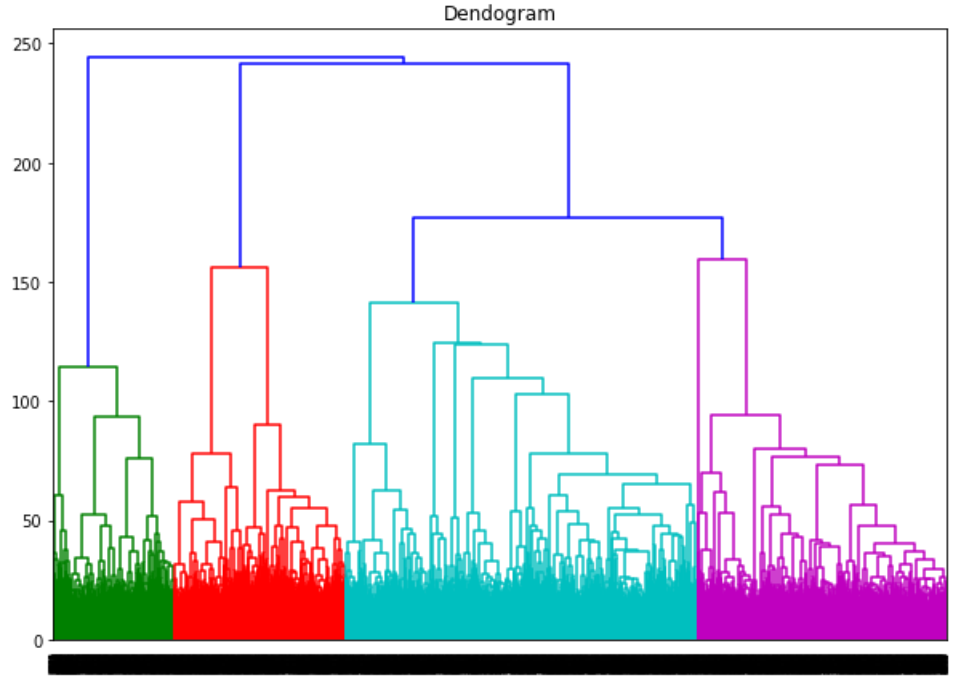


15. Agglomerative

In [107]:

```
import scipy.cluster.hierarchy as sch

X_set1_new_dense = X_set1_new.todense()
```

```
plt.figure(figsize=(10, 7))
plt.title("Dendogram")
dendo = sch.hierarchy.dendrogram(sch.linkage(X_set1_new_dense, method='ward'))
```



In [109]:

```
from sklearn.cluster import AgglomerativeClustering
Agglo = AgglomerativeClustering(n_clusters = 4).fit(X_set1_new_dense)
```

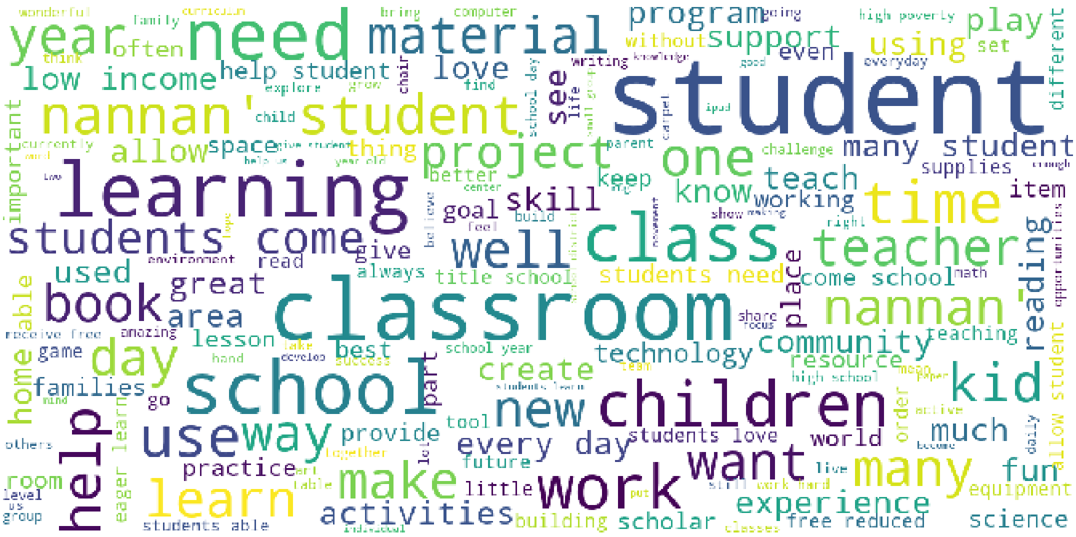
In [111]:

```
# Getting all the reviews in different clusters
Acluster1 = []
Acluster2 = []
Acluster3 = []
Acluster4 = []



for i in range(Agglo.labels_.shape[0]):
    if Agglo.labels_[i] == 0:
        Acluster1.append(FN[i])
    elif Agglo.labels_[i] == 1:
        Acluster2.append(FN[i])
    elif Agglo.labels_[i] == 2:
        Acluster3.append(FN[i])
    else :
        Acluster4.append(FN[i])

# Number of reviews in different clusters
print("\nNo. of essay in Cluster-1  : ",len(Acluster1))
print("\nNo. of essay in Cluster-2  : ",len(Acluster2))
print("\nNo. of essay in Cluster-3  : ",len(Acluster3))
print("\nNo. of essay in Cluster-4  : ",len(Acluster4))
```

```
No. of essay in Cluster-1  :  2790

No. of essay in Cluster-2  :  3935

No. of essay in Cluster-3  :  1917

No. of essay in Cluster-4  :  1358
```

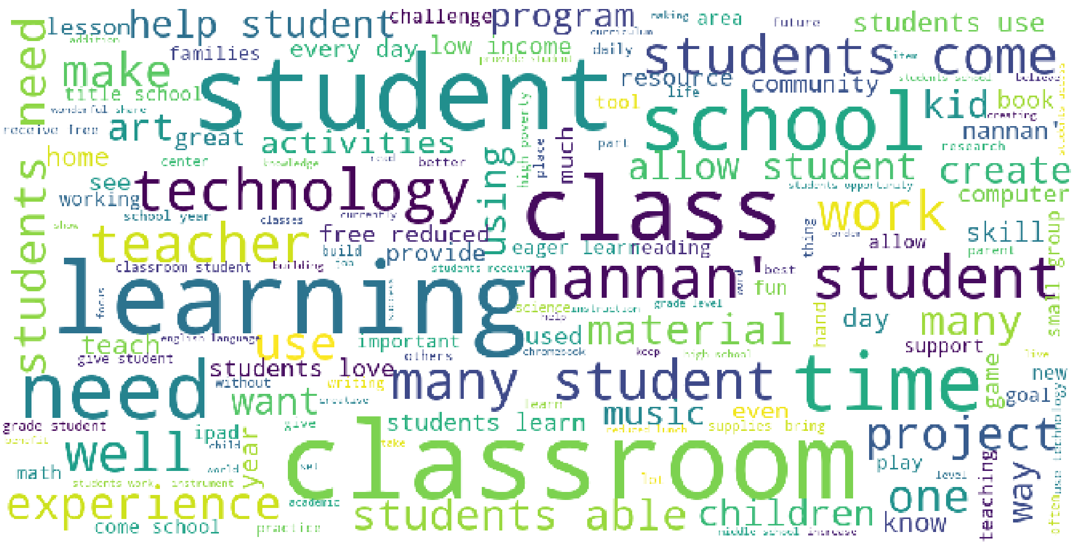
In [112]:

```
show_wordcloud(Acluster1)
```



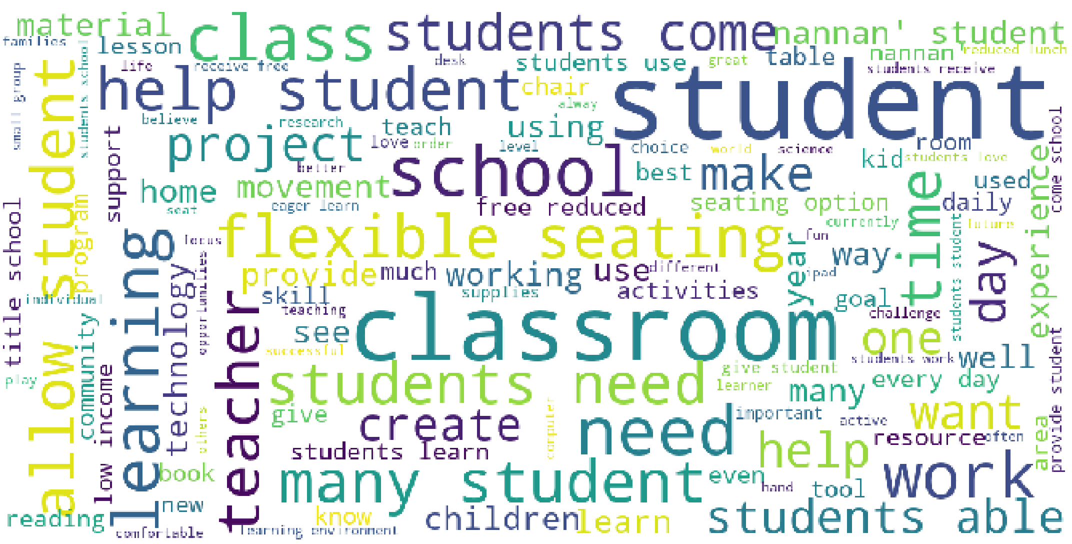
In [113]:

```
show_wordcloud(Acluster2)
```



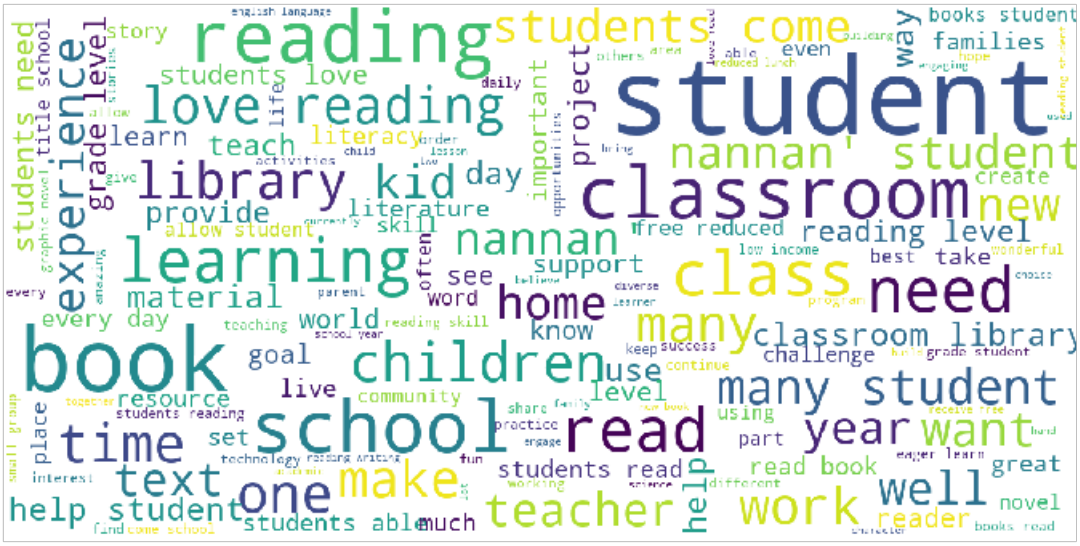
In [114]:

```
show_wordcloud(Acluster3)
```



In [115]:

```
show_wordcloud(Acluster4)
```



16. DBSCAN

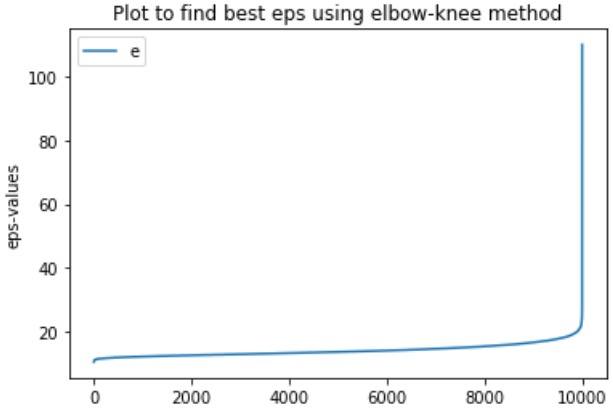
In [118]:

```
from sklearn.neighbors import KDTree

algo_title = 'DBSCAN Clustering'
minPts = 10                          #minPts:ln(n),n = 10000
tree = KDTree(X_set1_new_dense)

idx = 0
eps = []
for x_i in tqdm(X_set1_new_dense):
    eps.append(tree.query(X_set1_new_dense[idx], return_distance=True, k=minPts)[0][0][-1])
    idx += 1
eps.sort()

plt.plot(range(0,10000), eps[:10000])
plt.title("Plot to find best eps using elbow-knee method")
plt.xlabel('Integers')
plt.ylabel('eps-values')
plt.legend('elbow')
```

```
100%|██████████████████████████████████████████████████████████████████████████| 10000/10000 [14
:40<00:00, 11.36it/s]
```

Out[118]:

```
<matplotlib.legend.Legend at 0x1a3fd581898>
```



Integers

In [128]:

```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps = 19, min_samples=10, metric='euclidean').fit(X_set1_new_dense)
```

In [129]:

```
a = dbscan.labels_.tolist()
```

In [137]:

```
a[0:10]
```

Out[137]:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [144]:

```
a[140:160]
```

Out[144]:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1]
```

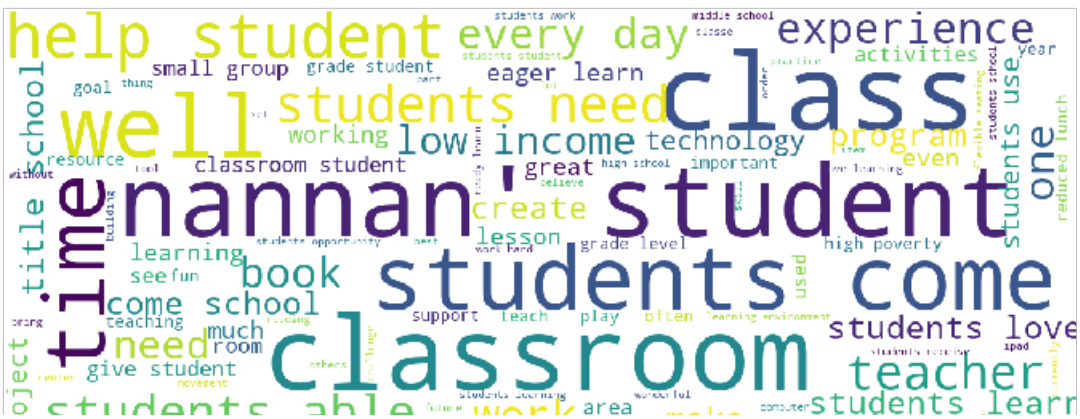
In [131]:

```
# Getting all the reviews in different clusters
dbcluster1 = []
dbcluster2 = []



for i in range(Agglo.labels_.shape[0]):
    if dbscan.labels_[i] == 0:
        dbcluster1.append(FN[i])
    else :
        dbcluster2.append(FN[i])

# Number of reviews in different clusters
print("\nNo. of essay in Cluster-1  : ",len(dbcluster1))
print("\nNo. of essay in Cluster-2  : ",len(dbcluster2))
```

```
No. of essay in Cluster-1  :  9879

No. of essay in Cluster-2  :  121
```

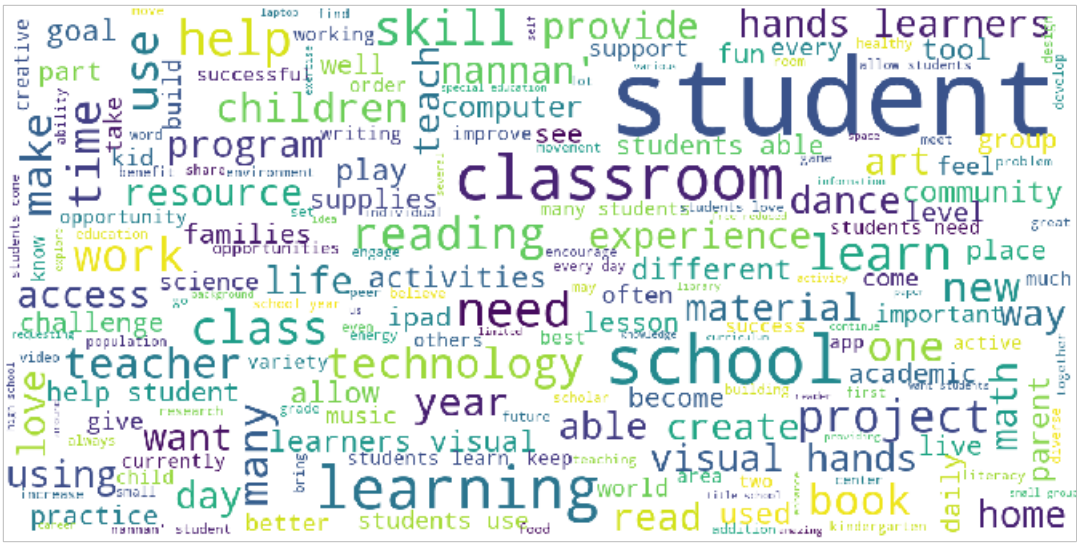
In [132]:

```
show_wordcloud(dbcluster1)
```



In [133]:

```
show_wordcloud(dbcluster2)
```



Observations

1. Here,we have taken only 10k datapoints due to computation issue prepreccessed the data and vectorized the data.

2. After vectorized stacked up the data using hstack and applied kmeans,Agglomerative and Dbscan algorithms on the data.

3. When applied kmeans got 16 clusters out of which the 10th cluster has nly one datapoint which implies or can be considered as an outier.

4. When applied Aglomerative got 4 clusters after visualizing dendogram and worked preety well.

5. When applied DBSCAN got two clusters 0 and -1,where -1 implies an outlier points.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: