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	Descri
project_id	A unique identifier for the proposed project. **Example:** `p03`
project_title	Title of the project. **Exam** • `Art Will Make You Ha` • `First Grade`
project_grade_category	Grade level of students for which the project is targeted. One of the foll enumerated va • `Grades Pr` • `Grades` • `Grades` • `Grades`
project_subject_categories	One or more (comma-separated) subject categories for the project fro following enumerated list of va • `Applied Lear` • `Care & Hu` • `Health & Sp` • `History & Ci` • `Literacy & Lang` • `Math & Sci` • `Music & The` • `Special N` • `Wa` **Exam** • `Music & The` • `Literacy & Language, Math & Sci`
school_state	State where school is located (Two-letter U.S. postal (https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations#Postal_co **Example**
project_subject_subcategories	One or more (comma-separated) subject subcategories for the pr **Exam** • `Lite` • `Literature & Writing, Social Scie`
project_resource_summary	An explanation of the resources needed for the project. **Exan** • `My students need hands on literacy materials to mar sensory ne`
project_essay_1	First application e
project_essay_2	Second application e
project_essay_3	Third application e
project_essay_4	Fourth application e

Feature	Descri
project_submitted_datetime	Datetime when project application was submitted. **Example:** 2016-04 12:43:56
teacher_id	A unique identifier for the teacher of the proposed project. **Exa** bdf8baa8fedef6bfeec7ae4ff1c1
teacher_prefix	Teacher's title. One of the following enumerated va • • • • • • Teac
teacher_number_of_previously_posted_projects	Number of project applications previously submitted by the same tea **Exampl**

* 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 [0]:
```
%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
```

```
D:\installed\Anaconda3\lib\site-packages\gensim\utils.py:1197: UserWarning: d
etected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

1.1 Reading Data

In [0]:
```
project_data = pd.read_csv('train_data.csv')
resource_data = pd.read_csv('resources.csv')
```

In [0]:
```
print("Number of data points in train data", project_data.shape)
print('-'*50)
print("The attributes of data :", project_data.columns.values)
```

```
Number of data points in train data (109248, 17)
--------------------------------------------------
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_prefix' 'sc
hool_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 [0]:
```
print("Number of data points in train data", resource_data.shape)
print(resource_data.columns.values)
resource_data.head(2)
```

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

Out[0]:

	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 [0]:
```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflo
w.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-fr
om-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-strin
g-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 & Scienc
e", "Warmth", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on
space "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to
replace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(emp
ty) ex:"Math & Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the tra
iling spaces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

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

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

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

1.3 preprocessing of `project_subject_subcategories`

In [0]:
```
sub_catogories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflo
w.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-fr
om-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-strin
g-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 & Scienc
e", "Warmth", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on
space "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to
replace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(emp
ty) ex:"Math & Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the tra
iling spaces
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())

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

# count of all the words in corpus python: https://stackoverflow.com/a/2289859
5/4084039
my_counter = Counter()
for word in project_data['clean_subcategories'].values:
    my_counter.update(word.split())

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

1.3 Text preprocessing

In [0]:
```
# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) +\
                        project_data["project_essay_2"].map(str) + \
                        project_data["project_essay_3"].map(str) + \
                        project_data["project_essay_4"].map(str)
```

In [0]:
```
project_data.head(2)
```

Out[0]:

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

In [0]:
```
#### 1.4.2.3 Using Pretrained Models: TFIDF weighted W2V
```

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

My students are English learners that are working on English as their second or third languages. We are a melting pot of refugees, immigrants, and native-born Americans bringing the gift of language to our school. \r\n\r\n We have over 24 languages represented in our English Learner program with students at every level of mastery. We also have over 40 countries represented with the families within our school. Each student brings a wealth of knowledge and ex periences to us that open our eyes to new cultures, beliefs, and respect.\"Th e limits of your language are the limits of your world.\"-Ludwig Wittgenstein Our English learner's have a strong support system at home that begs for more resources. Many times our parents are learning to read and speak English alo ng side of their children. Sometimes this creates barriers for parents to be able to help their child learn phonetics, letter recognition, and other readi ng 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 home is able to assist. All families with students within the Level 1 proficiency status, will be a offered to be a part of this program. These educational videos wil l be specially chosen by the English Learner Teacher and will be sent home re gularly to watch. The videos are to help the child develop early reading ski lls.\r\n\r\nParents that do not have access to a dvd player will have the opp ortunity to check out a dvd player to use for the year. The plan is to use t hese videos and educational dvd's for the years to come for other EL student s.\r\nnannan

==========

The 51 fifth grade students that will cycle through my classroom this year al l love learning, at least most of the time. At our school, 97.3% of the stude nts receive free or reduced price lunch. Of the 560 students, 97.3% are minor ity students. \r\nThe school has a vibrant community that loves to get togeth er and celebrate. Around Halloween there is a whole school parade to show off the beautiful costumes that students wear. On Cinco de Mayo we put on a big f estival with crafts made by the students, dances, and games. At the end of th e year the school hosts a carnival to celebrate the hard work put in during t he school year, with a dunk tank being the most popular activity.My students will use these five brightly colored Hokki stools in place of regular, statio nary, 4-legged chairs. As I will only have a total of ten in the classroom an d not enough for each student to have an individual one, they will be used in a variety of ways. During independent reading time they will be used as speci al 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 th e day they will be used by the students who need the highest amount of moveme nt in their life in order to stay focused on school.\r\n\r\nWhenever asked wh at the classroom is missing, my students always say more Hokki Stools. They c an't get their fill of the 5 stools we already have. When the students are si tting in group with me on the Hokki Stools, they are always moving, but at th e same time doing their work. Anytime the students get to pick where they can sit, the Hokki Stools are the first to be taken. There are always students wh o 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 t o do desk work and move at the same time. These stools will help students to meet their 60 minutes a day of movement by allowing them to activate their co re 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 s till.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 theme

d 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 and reduced-price lunch to qualify. Our school is an \"open classroom\" concept, which is very unique as there are no walls separating the classrooms. These 9 and 10 year-old students are very eager learners; they are like sponges, absorbing all the information and experiences and keep on wanting more.With these resources such as the comfy red throw pillows and the whimsical nautical hanging decor and the blue fish nets, I will be able to help create the mood in our classroom 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 pictures of each child with them, have them developed, and then hung in our classroom ready for their first day of 4th grade. This kind gesture will set the tone before even the first day of school! The nautical thank you cards will be used throughout the year by the students as they create thank you 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 to make our new school year a very successful one. Thank you!nannan

==========

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

==========

The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires. -William A. Ward\r\n\r\nMy school has 803 students which is makeup is 97.6% African-American, making up the largest segment of the student body. A typical school in Dallas is made up of 23.2% African-American students. Most of the students are on free or reduced lunch. We aren't receiving doctors, lawyers, or engineers children from rich backgrounds or neighborhoods. As an educator I am inspiring minds of young children and we focus not only on academics but one smart, effective, efficient, and disciplined students with good character.In our classroom we can utilize the Bluetooth for swift transitions during class. I use a speaker which doesn't amplify the sound enough to receive the message. Due to the volume of my speaker my students can't hear videos or books clearly and it isn't making the lessons as meaningful. But with the bluetooth speaker my students will be able to hear and I can stop, pause and replay it at any time.\r\nThe cart will allow me to have more room for storage of things that are needed for the day and has an extra part to it I can use. The table top chart has all of the letter, words and pictures for students to learn about different letters and it

```
is more accessible.nannan
==================================================
```

In [0]:
```
# 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 [0]:
```
sent = decontracted(project_data['essay'].values[20000])
print(sent)
print("="*50)
```

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

In [0]:
```
# \r \n \t remove from string python: http://texthandler.com/info/remove-line-breaks-python/
sent = sent.replace('\\r', ' ')
sent = sent.replace('\\"', ' ')
sent = sent.replace('\\n', ' ')
print(sent)
```

```
My kindergarten students have varied disabilities ranging from speech and lan
guage delays, cognitive delays, gross/fine motor delays, to autism. They are
eager beavers and always strive to work their hardest working past their limi
tations.     The materials we have are the ones I seek out for my students. I
teach in a Title I school where most of the students receive free or reduced
price lunch.  Despite their disabilities and limitations, my students love co
ming to school and come eager to learn and explore.Have you ever felt like yo
u had ants in your pants and you needed to groove and move as you were in a m
eeting? This is how my kids feel all the time. The want to be able to move as
they learn or so they say.Wobble chairs are the answer and I love then becaus
e they develop their core, which enhances gross motor and in Turn fine motor
skills.   They also want to learn through games, my kids do not want to sit a
nd do worksheets. They want to learn to count by jumping and playing. Physica
l engagement is the key to our success. The number toss and color and shape m
ats can make that happen. My students will forget they are doing work and jus
t have the fun a 6 year old deserves.nannan
```

In [0]:
```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
print(sent)
```

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

In [0]:
```
# 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', 'it
self', 'they', 'them', 'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 't
hat', "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', 'becau
se', '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', 'a
ll', 'any', 'both', 'each', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'tha
n', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "shoul
d've", 'now', 'd', 'll', 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'm
a', 'mightn', "mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shoul
dn't", 'wasn', "wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [0]:
```
# Combining all the above stundents
from tqdm import tqdm
preprocessed_essays = []
# tqdm is for printing the status bar
for sentance in tqdm(project_data['essay'].values):
    sent = decontracted(sentance)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    # https://gist.github.com/sebleier/554280
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    preprocessed_essays.append(sent.lower().strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████
██| 109248/109248 [01:53<00:00, 963.53it/s]
```

In [0]:
```
# after preprocesing
preprocessed_essays[20000]
```

Out[0]:
```
'my kindergarten students varied disabilities ranging speech language delays
cognitive delays gross fine motor delays autism they eager beavers always str
ive work hardest working past limitations the materials ones i seek students
i teach title i school students receive free reduced price lunch despite disa
bilities limitations students love coming school come eager learn explore hav
e ever felt like ants pants needed groove move meeting this kids feel time th
e want able move learn say wobble chairs answer i love develop core enhances
gross motor turn fine motor skills they also want learn games kids not want s
it worksheets they want learn count jumping playing physical engagement key s
uccess the number toss color shape mats make happen my students forget work f
un 6 year old deserves nannan'
```

1.4 Preprocessing of `project_title`

In [0]:
```
# similarly you can preprocess the titles also
```

1.5 Preparing data for models

In [0]:
```
project_data.columns
```

Out[0]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', 'project_grade_category', 'project_titl
e',
       'project_essay_1', 'project_essay_2', 'project_essay_3',
       'project_essay_4', 'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay'],
      dtype='object')
```

we are going to consider

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

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

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

1.5.1 Vectorizing Categorical data

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

In [0]:
```
# we use count vectorizer to convert the values into one
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()), lowercase=False, binary=True)
categories_one_hot = vectorizer.fit_transform(project_data['clean_categories'].values)
print(vectorizer.get_feature_names())
print("Shape of matrix after one hot encodig ",categories_one_hot.shape)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearning', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language']
Shape of matrix after one hot encodig  (109248, 9)
```

In [0]:
```
# we use count vectorizer to convert the values into one
vectorizer = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys()), lowe
rcase=False, binary=True)
sub_categories_one_hot = vectorizer.fit_transform(project_data['clean_subcateg
ories'].values)
print(vectorizer.get_feature_names())
print("Shape of matrix after one hot encodig ",sub_categories_one_hot.shape)
```

```
['Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement',
'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'NutritionEducati
on', 'Warmth', 'Care_Hunger', 'SocialSciences', 'PerformingArts', 'CharacterE
ducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Music', 'History_Geo
graphy', 'Health_LifeScience', 'EarlyDevelopment', 'ESL', 'Gym_Fitness', 'Env
ironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedSciences', 'Spec
ialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy']
Shape of matrix after one hot encodig  (109248, 30)
```

In [0]:
```
# you can do the similar thing with state, teacher_prefix and project_grade_ca
tegory also
```

1.5.2 Vectorizing Text data

1.5.2.1 Bag of words

In [0]:
```
# We are considering only the words which appeared in at least 10 documents(ro
ws or projects).
vectorizer = CountVectorizer(min_df=10)
text_bow = vectorizer.fit_transform(preprocessed_essays)
print("Shape of matrix after one hot encodig ",text_bow.shape)
```

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

In [0]:
```
# you can vectorize the title also
# before you vectorize the title make sure you preprocess it
```

1.5.2.2 TFIDF vectorizer

In [0]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=10)
text_tfidf = vectorizer.fit_transform(preprocessed_essays)
print("Shape of matrix after one hot encodig ",text_tfidf.shape)
```

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

1.5.2.3 Using Pretrained Models: Avg W2V

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

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

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

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

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

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

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

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


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

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

Out[0]:
```
'''\n# Reading glove vectors in python: https://stackoverflow.com/a/38230349/40
84039\ndef loadGloveModel(gloveFile):\n    print ("Loading Glove Model")\n
f = open(gloveFile,\'r\', encoding="utf8")\n    model = {}\n    for line in t
qdm(f):\n        splitLine = line.split()\n        word = splitLine[0]\n
embedding = np.array([float(val) for val in splitLine[1:]])\n        model[wo
rd] = embedding\n    print ("Done.",len(model)," words loaded!")\n    return
model\nmodel = loadGloveModel(\'glove.42B.300d.txt\')\n\n# ================
==========\nOutput:\n    \nLoading Glove Model\n1917495it [06:32, 4879.69it/
s]\nDone. 1917495  words loaded!\n\n# ============================\n\nwords =
[]\nfor i in preproced_texts:\n    words.extend(i.split(\' \'))\n\nfor i in p
reproced_titles:\n    words.extend(i.split(\' \'))\nprint("all the words in t
he coupus", len(words))\nwords = set(words)\nprint("the unique words in the c
oupus", len(words))\n\ninter_words = set(model.keys()).intersection(words)\np
rint("The number of words that are present in both glove vectors and our coup
us",       len(inter_words),"(",np.round(len(inter_words)/len(words)*100,
3),"%)")\n\nwords_courpus = {}\nwords_glove = set(model.keys())\nfor i in wor
ds:\n    if i in words_glove:\n        words_courpus[i] = model[i]\nprint("wo
rd 2 vec length", len(words_courpus))\n\n\n# stronging variables into pickle
files python: http://www.jessicayung.com/how-to-use-pickle-to-save-and-load-v
ariables-in-python/\n\nimport pickle\nwith open(\'glove_vectors\', \'wb\') as
f:\n    pickle.dump(words_courpus, f)\n\n\n'
```

In [0]:
```
# stronging variables into pickle files python: http://www.jessicayung.com/how
-to-use-pickle-to-save-and-load-variables-in-python/
# make sure you have the glove_vectors file
with open('glove_vectors', 'rb') as f:
    model = pickle.load(f)
    glove_words =  set(model.keys())
```

In [0]:
```
# average Word2Vec
# compute average word2vec for each review.
avg_w2v_vectors = []; # the avg-w2v for each sentence/review is stored in this
list
for sentence in tqdm(preprocessed_essays): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors.append(vector)

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

```
100%|██████████████████████████████████████████████████████████████████████
| 109248/109248 [00:59<00:00, 1823.16it/s]

109248
300
```

1.5.2.3 Using Pretrained Models: TFIDF weighted W2V

In [0]:
```
# S = ["abc def pqr", "def def def abc", "pqr pqr def"]
tfidf_model = TfidfVectorizer()
tfidf_model.fit(preprocessed_essays)
# we are converting a dictionary with word as a key, and the idf as a value
dictionary = dict(zip(tfidf_model.get_feature_names(), list(tfidf_model.idf_
)))
tfidf_words = set(tfidf_model.get_feature_names())
```

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

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

```
100%|██████████████████████████████████████████████████████████████████████
██| 109248/109248 [07:32<00:00, 241.33it/s]

109248
300
```

In [0]:
```
# Similarly you can vectorize for title also
```

1.5.3 Vectorizing Numerical features

In [0]:
```
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'
}).reset_index()
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

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

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

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

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

In [0]:
```
price_standardized
```

Out[0]:
```
array([[0.00098843, 0.00191166, 0.00330448, ..., 0.00153418, 0.00046704,
        0.00070265]])
```

1.5.4 Merging all the above features

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

In [0]:
```
print(categories_one_hot.shape)
print(sub_categories_one_hot.shape)
print(text_bow.shape)
print(price_standardized.shape)
```

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

In [0]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
# with the same hstack function we are concatinating a sparse matrix and a dense matirx :)
X = hstack((categories_one_hot, sub_categories_one_hot, text_bow, price_standardized))
X.shape
```

Out[0]:
```
(109248, 16663)
```

In [0]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the rea
der
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

Computing Sentiment Scores

In [0]:
```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

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

sid = SentimentIntensityAnalyzer()

for_sentiment = 'a person is a person no matter how small dr seuss i teach the
smallest students with the biggest enthusiasm \
for learning my students learn in many different ways using all of our senses
 and multiple intelligences i use a wide range\
of techniques to help all my students succeed students in my class come from a
variety of different backgrounds which makes\
for wonderful sharing of experiences and cultures including native americans o
ur school is a caring community of successful \
learners which can be seen through collaborative student project based learnin
g in and out of the classroom kindergarteners \
in my class love to work with hands on materials and have many different oppor
tunities to practice a skill before it is\
mastered having the social skills to work cooperatively with friends is a cruc
ial aspect of the kindergarten curriculum\
montana is the perfect place to learn about agriculture and nutrition my stude
nts love to role play in our pretend kitchen\
in the early childhood classroom i have had several kids ask me can we try coo
king with real food i will take their idea \
and create common core cooking lessons where we learn important math and writi
ng concepts while cooking delicious healthy \
food for snack time my students will have a grounded appreciation for the work
that went into making the food and knowledge \
of where the ingredients came from as well as how it is healthy for their bodi
es this project would expand our learning of \
nutrition and agricultural cooking recipes by having us peel our own apples to
make homemade applesauce make our own bread \
and mix up healthy plants from our classroom garden in the spring we will also
create our own cookbooks to be printed and \
shared with families students will gain math and literature skills as well as
 a life long enjoyment for healthy cooking \
nannan'
ss = sid.polarity_scores(for_sentiment)

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

# we can use these 4 things as features/attributes (neg, neu, pos, compound)
# neg: 0.0, neu: 0.753, pos: 0.247, compound: 0.93
```

```
D:\installed\Anaconda3\lib\site-packages\nltk\twitter\__init__.py:20: UserWar
ning:

The twython library has not been installed. Some functionality from the twitt
er package will not be available.


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

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 (https://scikit-learn.org/stable/modules/feature_selection.html)/reduction algorithms (https://scikit-learn.org/stable/modules/decomposition.html) ex: selectkbest features, pretrained word vectors, model based feature selection etc and reduce the number of features to 5k features
- step 3: Apply all three kmeans, Agglomerative clustering, DBSCAN
  - **K-Means Clustering:**
    ● Find the best 'k' using the elbow-knee method (plot k vs inertia_)
  - **Agglomerative Clustering:**
    ● Apply agglomerative algorithm (https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/) and try a different number of clusters like 2,5 etc.
    ● 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 (https://stackoverflow.com/a/48558030/4084039).
    ● 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

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.preprocessing import normalize

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_notebook as tqdm1
from tqdm import tqdm
import time
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

from sklearn.model_selection import train_test_split
```

```
C:\Users\LENOVO\Anaconda3\lib\site-packages\gensim\utils.py:1197: UserWarnin
g: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

In [2]:
```
project_data = pd.read_csv('train_data.csv')
resource_data = pd.read_csv('resources.csv')
```

In [3]:
```
print("Number of data points in train data", project_data.shape)
print('-'*50)
print("The attributes of data :", project_data.columns.values)
```

```
Number of data points in train data (109248, 17)
--------------------------------------------------
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_prefix' 'sc
hool_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']
```

Text preprocessing(1)

In [4]:
```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflo
w.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-fr
om-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-strin
g-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 & Scienc
e", "Warmth", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on
space "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to
replace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(emp
ty) ex:"Math & Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the tra
iling spaces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())
```

In [5]:
```
project_data['clean_categories'] = cat_list
project_data.drop(['project_subject_categories'], axis=1, inplace=True)
project_data.head(5)
```

Out[5]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	
2	21895	p182444	3465aaf82da834c0582ebd0ef8040ca0	Ms.	AZ	
3	45	p246581	f3cb9bffbba169bef1a77b243e620b60	Mrs.	KY	
4	172407	p104768	be1f7507a41f8479dc06f047086a39ec	Mrs.	TX	

In [6]:
```
# count of all the words in corpus python: https://stackoverflow.com/a/22898595/4084039
from collections import Counter
my_counter = Counter()
for word in project_data['clean_categories'].values:
    my_counter.update(word.split())
my_counter
```

Out[6]:
```
Counter({'Literacy_Language': 52239,
         'History_Civics': 5914,
         'Health_Sports': 14223,
         'Math_Science': 41421,
         'SpecialNeeds': 13642,
         'AppliedLearning': 12135,
         'Music_Arts': 10293,
         'Warmth': 1388,
         'Care_Hunger': 1388})
```

In [7]:
```
# dict sort by value python: https://stackoverflow.com/a/613218/4084039
cat_dict = dict(my_counter)
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))


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

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

In [8]:
```
sub_catogories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflo
w.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-fr
om-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-strin
g-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 & Scienc
e", "Warmth", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on
space "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to
replace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(emp
ty) ex:"Math & Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the tra
iling spaces
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())
```

In [9]:
```
project_data['clean_subcategories'] = sub_cat_list
project_data.drop(['project_subject_subcategories'], axis=1, inplace=True)
project_data.head(2)
```

Out[9]:

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

In [10]:
```
# count of all the words in corpus python: https://stackoverflow.com/a/2289859
5/4084039
from collections import Counter
my_counter = Counter()
for word in project_data['clean_subcategories'].values:
    my_counter.update(word.split())
```

In [11]:
```
# dict sort by value python: https://stackoverflow.com/a/613218/4084039
sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv: kv[1]))


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

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

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)
```

In [13]:
```
# 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[13]:

	id	price	quantity
0	p000001	459.56	7
1	p000002	515.89	21

In [14]:
```
# join two dataframes in python:
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

In [15]:
```
#presence of the numerical digits in a strings with numeric : https://stackove
rflow.com/a/19859308/8089731
def hasNumbers(inputString):
    return any(i.isdigit() for i in inputString)
p1 = project_data[['id','project_resource_summary']]
p1 = pd.DataFrame(data=p1)
p1.columns = ['id','digits_in_summary']
p1['digits_in_summary'] =  p1['digits_in_summary'].map(hasNumbers)
# https://stackoverflow.com/a/17383325/8089731
p1['digits_in_summary'] = p1['digits_in_summary'].astype(int)
project_data = pd.merge(project_data, p1, on='id', how='left')
project_data.head(5)
```

Out[15]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	project_
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	
2	21895	p182444	3465aaf82da834c0582ebd0ef8040ca0	Ms.	AZ	
3	45	p246581	f3cb9bffbba169bef1a77b243e620b60	Mrs.	KY	
4	172407	p104768	be1f7507a41f8479dc06f047086a39ec	Mrs.	TX	

5 rows × 21 columns

Text preprocessing(2)

In [16]:
```
# 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 [17]:
```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you'
, "you're", "you've",\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he'
, 'him', 'his', 'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'it
self', 'they', 'them', 'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 't
hat', "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', 'becau
se', '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', 'a
ll', 'any', 'both', 'each', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'tha
n', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "shoul
d've", 'now', 'd', 'll', 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'm
a', 'mightn', "mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shoul
dn't", 'wasn', "wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [18]:
```
# 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)
    sent = re.sub('nannan', '', sent)
    # https://gist.github.com/sebleier/554280
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    preprocessed_essays.append(sent.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████████████████████
| 109248/109248 [01:04<00:00, 1683.73it/s]
```

In [19]:
```
from tqdm import tqdm
preprocessed_titles = []
# tqdm is for printing the status bar
for title in tqdm(project_data['project_title'].values):
    _title = decontracted(title)
    _title = _title.replace('\\r', ' ')
    _title = _title.replace('\\"', ' ')
    _title = _title.replace('\\n', ' ')
    _title = re.sub('[^A-Za-z0-9]+', ' ', _title)
    # https://gist.github.com/sebleier/554280
    _title = ' '.join(e for e in _title.split() if e not in stopwords)
    preprocessed_titles.append(_title.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████████████████████|
109248/109248 [00:02<00:00, 37670.36it/s]
```

In [20]:
```
preprocessed_titles[1000]
```

Out[20]:
```
'sailing into super 4th grade year'
```

In [21]:
```
project_grade_catogories = list(project_data['project_grade_category'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47301924/4084039

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

project_grade_cat_list = []
for i in tqdm1(project_grade_catogories):
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Science", "Warmth", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on space "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to replace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(empty) ex:"Math & Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the trailing spaces
        temp = temp.replace('&','_')
    project_grade_cat_list.append(temp.strip())
```

In [22]:
```
project_data['clean_project_grade_category'] = project_grade_cat_list
project_data.drop(['project_grade_category'], axis=1, inplace=True)
project_data.head(2)
```

Out[22]:

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

2 rows × 21 columns

In [23]:
```
project_data.drop(['project_essay_1','project_essay_2','project_essay_3','proj
ect_essay_4'], axis=1, inplace=True)
project_data.head(2)
```

Out[23]:

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

In [24]:
```
#Replacing Nan's with maximum occured value: https://stackoverflow.com/a/51053
916/8089731
project_data['teacher_prefix'].value_counts().argmax()
project_data.fillna(value=project_data['teacher_prefix'].value_counts().argmax
(),axis=1,inplace=True)
```

In [25]:
```
project_data['preprocessed_essays'] = preprocessed_essays
project_data['preprocessed_titles'] = preprocessed_titles
```

In [26]:
```
project_data.columns
```

Out[26]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', 'project_title',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay', 'price', 'quantit
y',
       'digits_in_summary', 'clean_project_grade_category',
       'preprocessed_essays', 'preprocessed_titles'],
      dtype='object')
```

In [27]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the rea
der
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

2.2 Make Data Model Ready: encoding numerical, categorical features

In [28]:
```
X_train, X_test, y_train, y_test = train_test_split(project_data,project_data[
'project_is_approved'], test_size=0.33, stratify = project_data['project_is_ap
proved'])
# X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=
0.33, stratify=y_train)

X_train.drop(['project_is_approved'], axis=1, inplace=True)
X_test.drop(['project_is_approved'], axis=1, inplace=True)
# X_cv.drop(['project_is_approved'], axis=1, inplace=True)
print(X_train.shape)
print(X_test.shape)
```

```
(73196, 18)
(36052, 18)
```

1.4.1 Vectorizing Categorical data

In [29]:
```
# we use count vectorizer to convert the values into one hot encoded features
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_cat = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()), lowe
rcase=False, binary=True)
vectorizer_cat.fit(X_train['clean_categories'].values)
print(vectorizer_cat.get_feature_names())


categories_one_hot_train = vectorizer_cat.transform(X_train['clean_categories'
].values)
# categories_one_hot_cv = vectorizer_cat.transform(X_cv['clean_categories'].va
lues)
categories_one_hot_test = vectorizer_cat.transform(X_test['clean_categories'].
values)
print("Shape of matrix after one hot encodig_train ",categories_one_hot_train.
shape)
# print("Shape of matrix after one hot encodig_cv ",categories_one_hot_cv.shap
e)
print("Shape of matrix after one hot encodig_test ",categories_one_hot_test.sh
ape)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearning',
'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language']
Shape of matrix after one hot encodig_train  (73196, 9)
Shape of matrix after one hot encodig_test  (36052, 9)
```

In [30]:
```
# we use count vectorizer to convert the values into one hot encoded features
vectorizer_sub_cat = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys
()), lowercase=False, binary=True)
vectorizer_sub_cat.fit(X_train['clean_subcategories'].values)
print(vectorizer_sub_cat.get_feature_names())


sub_categories_one_hot_train = vectorizer_sub_cat.transform(X_train['clean_sub
categories'].values)
# sub_categories_one_hot_cv = vectorizer_sub_cat.transform(X_cv['clean_subcate
gories'].values)
sub_categories_one_hot_test = vectorizer_sub_cat.transform(X_test['clean_subca
tegories'].values)
print("Shape of matrix after one hot encodig_train ",sub_categories_one_hot_tr
ain.shape)
# print("Shape of matrix after one hot encodig_cv ",sub_categories_one_hot_cv.
shape)
print("Shape of matrix after one hot encodig_test ",sub_categories_one_hot_tes
t.shape)
```

```
['Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement',
'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'NutritionEducati
on', 'Warmth', 'Care_Hunger', 'SocialSciences', 'PerformingArts', 'CharacterE
ducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Music', 'History_Geo
graphy', 'Health_LifeScience', 'EarlyDevelopment', 'ESL', 'Gym_Fitness', 'Env
ironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedSciences', 'Spec
ialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy']
Shape of matrix after one hot encodig_train  (73196, 30)
Shape of matrix after one hot encodig_test  (36052, 30)
```

In [31]:
```
# we use count vectorizer to convert the values into one hot encoded features
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_state = CountVectorizer( lowercase=False, binary=True)
vectorizer_state.fit(X_train['school_state'].values)
print(vectorizer_state.get_feature_names())


school_state_one_hot_train = vectorizer_state.transform(X_train['school_state'
].values)
# school_state_one_hot_cv = vectorizer_state.transform(X_cv['school_state'].va
lues)
school_state_one_hot_test = vectorizer_state.transform(X_test['school_state'].
values)
print("Shape of matrix after one hot encodig_train ",school_state_one_hot_trai
n.shape)
# print("Shape of matrix after one hot encodig_cv ",school_state_one_hot_cv.sh
ape)
print("Shape of matrix after one hot encodig_test ",school_state_one_hot_test.
shape)
```

```
['AK', 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA', 'HI', 'I
A', '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_train  (73196, 51)
Shape of matrix after one hot encodig_test  (36052, 51)
```

In [32]:
```
# we use count vectorizer to convert the values into one hot encoded features
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_teacherprefix = CountVectorizer( lowercase=False, binary=True)
vectorizer_teacherprefix.fit(X_train['teacher_prefix'].values.astype('U'))
print(vectorizer_teacherprefix.get_feature_names())

#https://stackoverflow.com/a/39308809/8089731
teacher_prefix_one_hot_train = vectorizer_teacherprefix.transform(X_train['tea
cher_prefix'].values.astype('U'))
# teacher_prefix_one_hot_cv = vectorizer_teacherprefix.transform(X_cv['teacher
_prefix'].values.astype('U'))
teacher_prefix_one_hot_test = vectorizer_teacherprefix.transform(X_test['teach
er_prefix'].values.astype('U'))
print("Shape of matrix after one hot encodig_train ",teacher_prefix_one_hot_tr
ain.shape)
# print("Shape of matrix after one hot encodig_cv ",teacher_prefix_one_hot_cv.
shape)
print("Shape of matrix after one hot encodig_test ",teacher_prefix_one_hot_tes
t[:5,:])
# print(X_train['teacher_prefix'].value_counts())
```

```
['Dr', 'Mr', 'Mrs', 'Ms', 'Teacher']
Shape of matrix after one hot encodig_train  (73196, 5)
Shape of matrix after one hot encodig_test    (0, 3)    1
  (1, 2)        1
  (2, 3)        1
  (3, 4)        1
  (4, 1)        1
```

In [33]:
```
print(project_data['clean_project_grade_category'].unique())# we use count vec
torizer to convert the values into one hot encoded features
from sklearn.feature_extraction.text import CountVectorizer
# https://stackoverflow.com/a/38161028/8089731
pattern = "(?u)\\b[\\w-]+\\b"
vectorizer_projectgrade = CountVectorizer(token_pattern=pattern, lowercase=Fal
se, binary=True)
vectorizer_projectgrade.fit(X_train['clean_project_grade_category'].values)
print(vectorizer_projectgrade.get_feature_names())

#https://stackoverflow.com/a/39308809/8089731
project_grade_category_one_hot_train = vectorizer_projectgrade.transform(X_tra
in['clean_project_grade_category'].values)
# project_grade_category_one_hot_cv = vectorizer_projectgrade.transform(X_cv
['clean_project_grade_category'].values)
project_grade_category_one_hot_test = vectorizer_projectgrade.transform(X_test
['clean_project_grade_category'].values)
print("Shape of matrix after one hot encodig_train ",project_grade_category_on
e_hot_train.shape)
# print("Shape of matrix after one hot encodig_cv ",project_grade_category_one
_hot_cv.shape)
print("Shape of matrix after one hot encodig_test ",project_grade_category_one
_hot_test[:5,:])
```

```
['GradesPreK-2' 'Grades6-8' 'Grades3-5' 'Grades9-12']
['Grades3-5', 'Grades6-8', 'Grades9-12', 'GradesPreK-2']
Shape of matrix after one hot encodig_train  (73196, 4)
Shape of matrix after one hot encodig_test    (0, 3)    1
  (1, 0)        1
  (2, 3)        1
  (3, 1)        1
  (4, 2)        1
```

Vectorizing Numerical features

In [34]:
```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
# from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import normalize

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

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

# train_text_feature_onehotCoding = normalize(train_text_feature_onehotCoding, axis=0)
# Now standardize the data with above maen and variance.
price_standardized_train = normalize(X_train['price'].values.reshape(-1, 1),axis=0)
# price_standardized_cv = price_scalar.transform(X_cv['price'].values.reshape(-1, 1))
price_standardized_test = normalize(X_test['price'].values.reshape(-1, 1),axis=0)
print(price_standardized_train.shape)
# print(price_standardized_cv.shape)
print(price_standardized_test.shape)
```

```
(73196, 1)
(36052, 1)
```

In [35]:
```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
# from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import normalize

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

# Now standardize the data with above maen and variance.
quantity_standardized_train = normalize(X_train['quantity'].values.reshape(-1, 1),axis=0)
# quantity_standardized_cv = quantity_scalar.transform(X_cv['quantity'].values.reshape(-1, 1))
quantity_standardized_test = normalize(X_test['quantity'].values.reshape(-1, 1),axis=0)
print(quantity_standardized_train.shape)
# print(quantity_standardized_cv.shape)
print(quantity_standardized_test.shape)
```

```
(73196, 1)
(36052, 1)
```

In [36]:
```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
# from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import normalize

# 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)

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

# Now standardize the data with above maen and variance.
teacher_number_of_previously_posted_projects_standardized_train = normalize(X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1, 1),axis=0)
# teacher_number_of_previously_posted_projects_standardized_cv = teacher_number_of_previously_posted_projects_scalar.transform(X_cv['teacher_number_of_previously_posted_projects'].values.reshape(-1, 1))
teacher_number_of_previously_posted_projects_standardized_test = normalize(X_test['teacher_number_of_previously_posted_projects'].values.reshape(-1, 1),axis=0)
print(teacher_number_of_previously_posted_projects_standardized_train.shape)
# print(teacher_number_of_previously_posted_projects_standardized_cv.shape)
print(teacher_number_of_previously_posted_projects_standardized_test.shape)
```

```
(73196, 1)
(36052, 1)
```

In [ ]:
```
```

In [37]:
```
# please write all the code with proper documentation, and proper titles for each subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debugging your code
# make sure you featurize train and test data separatly

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

2.3 Make Data Model Ready: encoding eassay, and project_title

In [38]:
```
X_train.head(2)
```

Out[38]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	pro
4279	136257	p120950	18b42a21b08dda1e0788525af1163ab4	Ms.	MN	
17572	100630	p037368	678f3b6ae314bb90fa888de642eed8b6	Mrs.	UT	

TFIDF Vectorizer on `project_TEXT/ESSAYS` (Train,Cv,Test)

In [40]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer_tfidf_essays = TfidfVectorizer(min_df=10,max_features=5000,ngram_range=(1,2))
vectorizer_tfidf_essays.fit(X_train['preprocessed_essays'])

text_tfidf_train = vectorizer_tfidf_essays.transform(X_train['preprocessed_essays'])
# text_tfidf_cv = vectorizer_tfidf_essays.transform(X_cv['preprocessed_essays'])
text_tfidf_test = vectorizer_tfidf_essays.transform(X_test['preprocessed_essays'])
print("Shape of matrix after tfidf_text_train ",text_tfidf_train.shape)
# print("Shape of matrix after tfidf_text_cv ",text_tfidf_cv.shape)
print("Shape of matrix after tfidf_text_test ",text_tfidf_test.shape)
```

```
Shape of matrix after tfidf_text_train  (73196, 5000)
Shape of matrix after tfidf_text_test  (36052, 5000)
```

TFIDF Vectorizer on `project_title` (Train,Cv,Test)

In [41]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer_tfidf_title = TfidfVectorizer(min_df=10)
vectorizer_tfidf_title.fit(X_train['preprocessed_titles'])

title_tfidf_train = vectorizer_tfidf_title.transform(X_train['preprocessed_tit
les'])
# title_tfidf_cv = vectorizer_tfidf_title.transform(X_cv['preprocessed_title
s'])
title_tfidf_test = vectorizer_tfidf_title.transform(X_test['preprocessed_title
s'])
print("Shape of matrix after tfidf_title_train ",title_tfidf_train.shape)
# print("Shape of matrix after tfidf_title_cv ",title_tfidf_cv.shape)
print("Shape of matrix after tfidf_title_test ",title_tfidf_test.shape)
```

```
Shape of matrix after tfidf_title_train  (73196, 2643)
Shape of matrix after tfidf_title_test  (36052, 2643)
```

In [1]:
```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import dill
# dill.dump_session('notebook_env.db')
dill.load_session('notebook_env.db')
```

```
C:\Users\LENOVO\Anaconda3\lib\site-packages\gensim\utils.py:1197: UserWarnin
g: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

In [2]:
```
project_data.columns
```

Out[2]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', 'project_title',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay', 'price', 'quantit
y',
       'digits_in_summary', 'clean_project_grade_category',
       'preprocessed_essays', 'preprocessed_titles'],
      dtype='object')
```

In [ ]:
```

```

In [3]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr = hstack((categories_one_hot_train,sub_categories_one_hot_train,school_st
ate_one_hot_train,teacher_prefix_one_hot_train
               ,project_grade_category_one_hot_train,price_standardized_train,
quantity_standardized_train
               ,teacher_number_of_previously_posted_projects_standardized_trai
n,text_tfidf_train,title_tfidf_train))
# X_cr = hstack((categories_one_hot_cv,sub_categories_one_hot_cv,school_state_
one_hot_cv,teacher_prefix_one_hot_cv
#                ,project_grade_category_one_hot_cv,price_standardized_cv,quan
tity_standardized_cv
#                ,teacher_number_of_previously_posted_projects_standardized_c
v,text_tfidf_cv,title_tfidf_cv)).tocsr()
X_te = hstack((categories_one_hot_test,sub_categories_one_hot_test,school_stat
e_one_hot_test,teacher_prefix_one_hot_test
               ,project_grade_category_one_hot_test,price_standardized_test,qu
antity_standardized_test
               ,teacher_number_of_previously_posted_projects_standardized_test
,text_tfidf_test,title_tfidf_test))

print("Final Data matrix on TFIDF")
print(X_tr.shape, y_train.shape)
# print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix on TFIDF
(73196, 7745) (73196,)
(36052, 7745) (36052,)
====================================================================================
======================
```

In [4]:
```
X_te.shape
```

Out[4]:
```
(36052, 7745)
```

In [5]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code
# make sure you featurize train and test data separatly

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

2.4 Dimensionality Reduction on the selected features

In [6]:
```
###########################################################################
# from sklearn.preprocessing import MaxAbsScaler
# scaler = MaxAbsScaler()
# X_tr = scaler.fit_transform(X_tr,y_train)
# X_te = scaler.transform(X_te)
###########################################################################
from sklearn.feature_selection import SelectKBest, chi2
t = SelectKBest(chi2,k=5000).fit(X_tr, y_train)
X_tr = t.transform(X_tr)
X_te = t.transform(X_te)
###########################################################################
print("Final Data matrix on TFIDF")
print(X_tr.shape, y_train.shape)
# print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

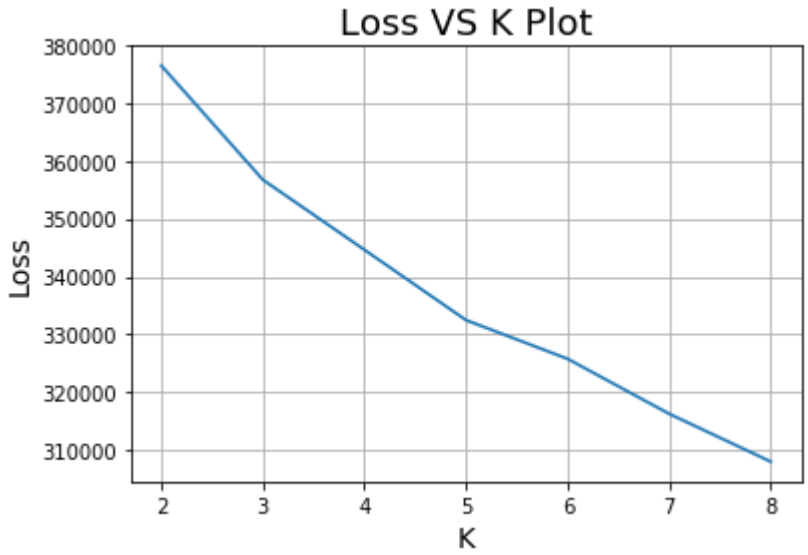
```
Final Data matrix on TFIDF
(73196, 5000) (73196,)
(36052, 5000) (36052,)
====================================================================================================
```

In [7]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the rea
der
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

2.5 Apply Kmeans

In [8]:
```
from sklearn.cluster import KMeans

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

In [9]:
```
plt.plot(k_values, loss)
plt.xlabel('K',size=14)
plt.ylabel('Loss',size=14)
plt.title('Loss VS K Plot',size=18)
plt.grid()
plt.show()
```



In [10]:
```
optimal_k = 6

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

In [ ]:
```

```

In [11]:
```
essays = X_train['preprocessed_essays'].values

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

In [ ]:

In [12]:
```
for i in range(3):
    print('%s\n'%(cluster1[i]))
```

```
within class i great diversity learners i never school including preschool ki
ndergarten second time finding ways differentiate bit difficult i want make s
ure students challenged not discouraged every student learns pace need suppor
t class although student different background different learning style amazin
g support they love come class knowing opportunity learn help teach one anoth
er the magic boards practice cards language learning centers math folder game
s students practice word building number recognition the sand alphabet letter
s allows students practice letter recognition fun time the parachute large di
ce alphabet bean bags give class opportunity build team building skills well
learning abc 123 your generous donation allow expand knowledge sweet kinderga
rtners build teamwork skills with support yacker tracker i able teach student
s manage voices working groups i also able provide extra one one practice stu
dent help become confident learning

we small rural school east texas high poverty rate low ses there no one way a
pproach learning message teachers try demonstrate daily students the greatest
challenge provide opportunities students reach greatest learning potential wi
thin allotted budget our team teaching approach math ela instruction strives
present intermediate students motivating modes learning school optimizing eng
agement in multi disciplinary classroom learning styles dictate students shor
t periods large group instruction several small even independent tasks made a
vailable student choice this ipad mini 4 shared among students small groups b
ecause passed around young students also needing protective case device on ip
ad student access folders containing apps address specific academic needs rea
ding math writing these skills closely monitor grade level teks highly motiva
ting it certainly improve engagement make learning fun

belle hall awesome school full wonderful learners over 600 students come comp
uter lab week we recently received 35 purposed laptops created new lab since
not 1 1 school means access technology students this new lab allow students a
ccess computer technology three times often our students technologically prep
ared move forward education our students need effectively use many computer p
rograms the copy paste feature touch pad much difficult use mouse young stude
nts students belle hall already comfortable mouse usage having pieces equipme
nt lab lab makes students less likely get frustrated computer without mice us
age lab probably not grow even though many students using touch screen device
s outside school not types devices belle hall mice must
```

In [13]:
```
for i in range(3):
    print('%s\n'%(cluster2[i]))
```

```
my classroom slc consist low function students see world differently us they
face challenges everyday hard time functioning others we title 1 school diffi
cult get materials need keep minds hands busy my students live neighborhood l
ow income rates not necessities nor materials need help better understand wor
ld my students need different materials regular my scholars learn best able e
xploder see world offer whether hands computer they however learn differently
see world different light my students not grade levels different learning lev
els which challenging times i believe students necessities others students un
fortunately title 1 school unable supply us materials need i would like teach
students everyday living your donations greatly appreciated thank

my students special many reasons there genuine thirst learning among students
despite not even basic materials class they inspiration every day the student
s i teach inspire daily basis not allowing shortage resources prevent learnin
g it would easy complain rather make best little my students eager learners e
njoy demonstrating creative touch comes learning social studies we work toget
her financially challenged school district need simple basic tables would rep
lace older poorly maintained furniture affording students opportunity work ar
ea clean sturdy organized go long way helping students achieve full potential
in current situation little no room students work show creativity demonstrate
comprehension lesson my students great kids heart deserve least basic needs c
lassroom there no doubt mind climate culture would vastly improved addition n
ew tables thank much taking time read project

in class multiply disabled students camden new jersey daily learning overcome
obstacles break barriers many little experience world outside tiny socioecono
mically depressed neighborhoods with proper technological resources sky limit
camden poorest one dangerous cities america many students lack stable home en
vironments school provides safe know my students little experience world outs
ide tiny socioeconomically depressed neighborhoods with proper technological
resources learning stations expand classroom experience outside walls classro
om my students work lessons integrated across content differentiated meet spe
cific needs interests learning styles that proper academic material important
with proper supplies resources expand classroom experience outside walls clas
sroom my students work lessons integrated across content differentiated meet
specific needs interests learning styles technological literacy essential stu
dents competitive 21st century global economy as result integrating technolog
y classroom instruction one top priorities good educator access technology do
cucam would allow students work cooperatively active participants learning pr
ocess rather passive recipients information would ensure develop better depth
knowledge concepts learning it also allows differentiation not basis academic
levels interests well the round table chairs help provide setting conducive p
roject based instruction using technology supplies collaborative learning env
ironment researching writing mathematical application skills rather simple dr
ills practice helps foster student creativity collaboration communication ski
lls many employers demand 21st century workforce these items allow instructio
n highly customized students work independently active participants learning
process rather passive recipients information
```

In [14]:
```
for i in range(3):
    print('%s\n'%(cluster3[i]))
```

students come school eager learn explore they excited talk friends teacher at young age children acquire new skills every day my biggest challenge getting students attention keeping engaged activity play often talked relief serious learning but children play serious learning play really work childhood fred r ogers i teach 24 wonderful four five year old children love learning enjoy co ming school every day they happy inquisitive curious energetic love learn pla y i want students look forward coming school every day fun learn i want first experience school positive watch wonder class four year olds release butterfl y each child unique live adult butterfly release children learn life cycle bu tterfly listen stories butterflies look books completing puzzle butterfly inc rease spatial awareness fine motor skills the class collaboratively work colo ring illustrations the very hungry caterpillar having child take part process coloring illustrations give ownership child class make book meaningful childr en get paint paper butterfly watch color spreads water added diffusing paper finally children get eat caterpillar shaped cookie this fun filled butterfly themed day incorporate science reading math art it day children remember fore ver please help us bring project life

as educator goal provide students positive learning experiences safe comforta ble brain friendly learning environment rich opportunities personal expressio n i believe giving students choices learn teaching reflective learners proble ms solvers my students learn responsibility empathy compassion better readers writers mathematicians learn play together classroom we support learn work to gether ensure every student classroom able achieve personal best i extremely excited project i cannot wait give fantastic first graders opportunity use le gos learn grow subject areas while many people may see legos think play i see project way provide students important learning tools enhance learning classr oom big way with amazing materials students play games legos reinforce additi on subtraction fluency they create settings characters problems solutions wri te animate stories they become better readers engineers following included in struction cards lego idea book guide creations inspire creativity as complete community unit students opportunity build community using information learn i n science able support science standards creating animals demonstrating anima l adaptations creating habitats through structured activities free play stem challenges students opportunity collaborate learners become better readers wr iters scientists engineers problem solvers reluctant learners surely drawn en gaging lessons incorporate hands interactive activities involving one favorit e toys please consider supporting project provide students many opportunities learn grow

my students hispanic mainly central america asian mainly vietnam african amer ican white multi racial students i work public elementary school i work stude nts regardless racial ethnic background learning mental medical needs the bar riers exist families students include poverty not finishing high school reuni fication children parents not known unstable home environments parents hardwo rking individuals working 2 3 jobs leave little time spend children all indiv iduals overcome barrier lifetime whether struggle making maintaining relation ships believing one self i work students daily basis building resiliency the build anger anxiety stress sadness bullied manifest students lacking self con trol self regulation self confidence ability forgive students held back barri ers unable overcome also feel social isolation behavior concerns feel envious towards others the various lego books supplies used students fifth grade coun seling groups students struggle peer relationships positive leaders confidenc e the excitement students see use legos amazing students often use legos buil d something legos used help students build skills interacting others communic ation conflict resolution leadership team work in eight week counseling group students identify personal strengths areas would like improve creating lego c

haracter practice listening communication skills trying replicate structure d esigned another student student describes partners switch turns components te amwork leadership conflict resolution students work together assigned roles b uild structure

In [15]:
```
for i in range(3):
    print('%s\n'%(cluster4[i]))
```

our upper elementary school serves 3rd 5th graders south carolina our student s come high poverty social economic status 98 receiving free lunch a good maj ority students receive backpack lunches weekends half students speak another language home my students read 1 2 reading levels 3rd grade status my student s lack background knowledge home resources education the majority class males attention deficits girls need constant movement the kore wobble chairs allow students move around never leave seat the chair helps develop balance control the kore wobble chair life changer active kids kids adhd autistic kids aggres sive wobbling kore chair increases blood flow brain thereby quiets requiremen t fidgety kids move kids learn better productive the kid may considered un te achable able learn flexible seating wobble chairs allow students work learn b est students able choose chair desk works best

i one two pe teachers title 1 school located florence sc our students come us various backgrounds environments some two parent homes single family foster m any raised grandparents even living hotels the positive environment provide s chool positive thing many students see day everyday try greet kids smile posi tive attitude when students enter gym ready move as pe teachers job encourage set good example health wellness our students sit day play video games aftern oon need encourage stay active the students classes tune today technology inc luding fitbits they want monitor steps techy way better way encourage fitbit zip technology made kids excited active want encourage way possible the kids feel accomplished proud concrete way show fruits labor they excited show step numbers compare compete friends steps many parents cannot afford latest techn ology donation allow students opportunity try make lasting effect health well we would use fitbits lead example track physical activity reward accomplishme nts

our students come across city philadelphia entire school qualifies free lunch transportation school based fact majority students live poverty level despite issues facing students every day come school every day persevere mostly every student attending college post graduation our students need support available school district often cannot provide resources necessary students compete hig h schools region my students discussed needs fitness room spinning machine on e first ideas students they felt would help reach cardio goals given fact sch ool not gym we limited space fitness room mainly weights not idea students an spinning machine would meet every students needs allow students reach activit y goals daily basis developing fitness room goal students since i hired schoo l the student body given several goals wish accomplish development fitness ce nter make accessible appealing students school

In [16]:
```
for i in range(3):
    print('%s\n'%(cluster5[i]))
```

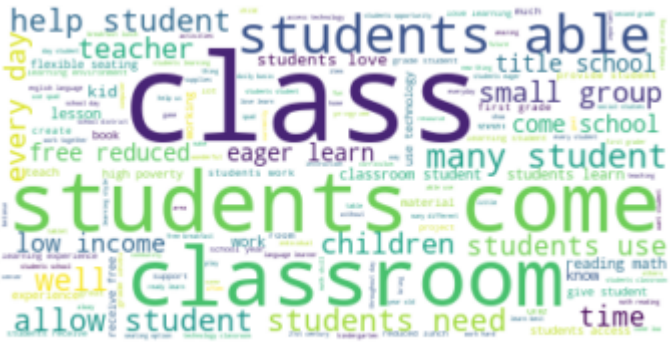
```
my students diverse socioeconomic backgrounds ethnicities they hardworking wa
nt best i want provide stimulating environment encourages respect innovation
creativeness collaboration many students come school wanting learn want conti
nue enthusiastic learning our school reaches families family fun nights trave
ling neighborhoods tutoring sessions fun nights school build sense community
all children deserve learn based learning styles we several wiggly first grad
ers could benefit using wobble chairs i feel would engaged projects able wobb
le move around learning first graders spend much day participating literacy a
ctivities hence sit much this challenging young children i feel wobble chairs
help kids excited different seating options reading writing learning i excite
d get 21st century seating classroom thank considering donation

as educator every experience classes smart literally extra homework keep our
kindergarten classes done year year able grasp many first grade concepts lite
racy math most kindergarten kids live urban area baltimore yet despite realit
y oblivious condition they love love books i mean love books new math concept
s introduced see brains calculating working when lightbulb aha moment happens
i know i helped click learning however i not take credit introduced primed ri
ght love learning colleague pre k teacher many years if able meet chat wonder
kids would definitely 100 amazed love kids remember poem all i needed to know
i learned kindergarten sandwiched milk cookies taking nap key unlocking poten
tial arts integrated learning live balanced life learn think draw paint sing
dance play work every day the document camera chromebook combo donation allow
learners chance not visualize follow step step directions needed create origi
nal raps dances 3 d drawing painting projects new feature arts integrated exp
loration station

our school located rural community every student needs something different wa
lk classroom my job provide safe place students grow learners no matter backg
round coming about 60 students receive free reduced lunch our students come d
iver backgrounds they different life experiences these students going achieve
whatever set minds go far places life i love watching grow learners in first
grade kids like move lot having stability discs sit allow continue chairs cus
hions use would prefer students also like sit around classroom learning the r
ug would provide area students sit whole group lessons area students work ind
ependently i excited see student use new learning space helps improve learnin
g focus thank taking look first project
```

In [17]:
```
for i in range(3):
    print('%s\n'%(cluster6[i]))
```

```
welcome intellectually gifted classroom i teach highly gifted disabled gifted
we class culturally income diverse all students uniquely special i blessed in
credible students my students hardworking big hearts they always giving back
school community they come new school year rip roaring ready go i excited see
students accomplish new school year at beginning year i start gifted class al
l about me project students get express tell research name family origins del
ve discovering individual natural talents this unit always breaks ice student
s i get know new students personal level as part unit i lead students abstrac
t art project includes initial my students love creating personal art pieces
i display school constantly ask take home they not love creation process cher
ish art pieces this meets gifted teacher standards learn fun process this i l
ove teaching quest

i work low income elementary school limited funding comes technology my stude
nts talented continually wish grow my students kind curious eager learn their
love reading keeps library recess lunch the staff sees thirst education works
hard every student comes because students not get chance use computers outsid
e computer lab set computers library would excellent way broaden computer use
inquisitive students this lenovo ideapad chromebook rotated throughout librar
y students need access online catalog this promote independent searching mate
rials when working small groups i also able deliver mini lessons keyboarding
research using additional technology the practice students using computers sa
fely responsibly better as students get comfortable using technology independ
ently i would like open lab time library school allow students continue resea
rch classroom projects reports students community rely heavily computer time
school limited no access home

our school inclusive high school located city one fastest growing areas north
ern california the school stands large bustling city historic farmland demons
trating recent shift agricultural suburban community some people may see chan
ge challenge school diversity strength it provides us opportunity learn many
backgrounds people walk halls i fortunate part unique community feels like fa
mily school contemporary art programs use digital imagery assist students cre
ating storing artwork our school computers equipped date photo editing softwa
re however rendered useless without modern digital cameras support wealth pho
tographs taken 130 students our current digital cameras date no longer servic
eable feature breaks they also not carry speed precision needed capture color
s lines textures student composition new nikon d3200 dslr camera equipment al
lows students quickly capture images share others when paired sony bdps3700 s
treaming blu ray disc player wi fi sony mdrrf985rk wireless rf headphones abl
e create digital videos along digital photos many today university programs r
equire portfolios students projects part college admissions process having di
rect access new technology allow students create successful digital portfolio
presentation saved organized sent directly admissions office
```

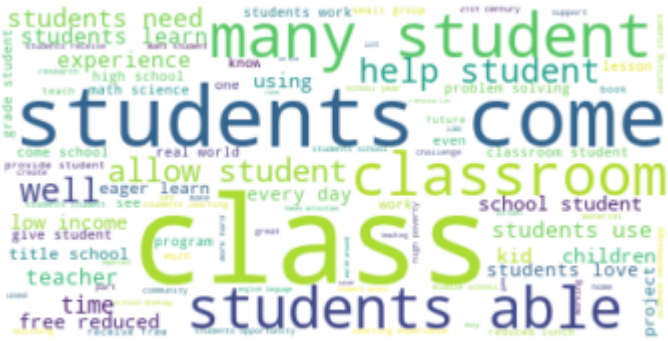
In [18]:
```
# for i in cluster1:
#     print(i.split())
# cluster1.values()
```

In [19]:
```
#cluster 1
words=''
for i in cluster1:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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

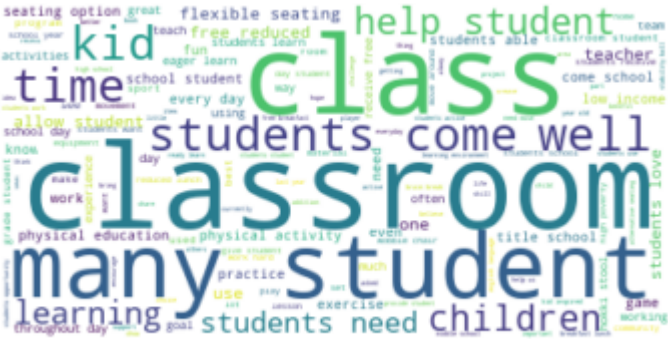


In [20]:
```
#cluster 2
words=''
for i in cluster2:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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

In [21]:
```
#cluster 3
words=''
for i in cluster3:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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

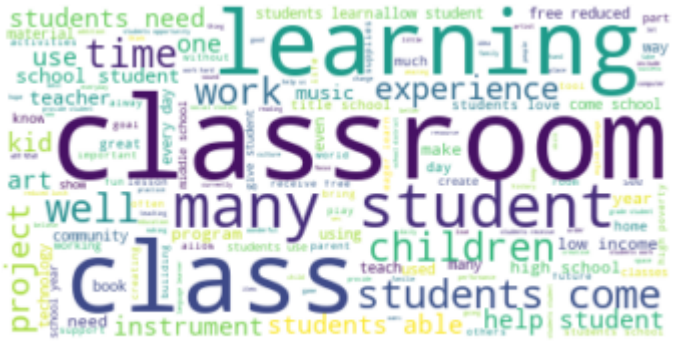


In [22]:
```
#cluster 4
words=''
for i in cluster4:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



In [23]:
```
#cluster 5
words=''
for i in cluster5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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

In [24]:
```
#cluster 6
words=''
for i in cluster6:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



In [0]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the rea
der
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

2.6 Apply AgglomerativeClustering

In [2]:
```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import dill
# dill.dump_session('notebook_env.db')
dill.load_session('notebook_env.db')
```

```
C:\Users\LENOVO\Anaconda3\lib\site-packages\gensim\utils.py:1197: UserWarnin
g: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

In [3]:
```
project_data.columns
```

Out[3]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', 'project_title',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay', 'price', 'quantit
y',
       'digits_in_summary', 'clean_project_grade_category',
       'preprocessed_essays', 'preprocessed_titles'],
      dtype='object')
```

In [ ]:
```

```

In [4]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr = hstack((categories_one_hot_train,sub_categories_one_hot_train,school_st
ate_one_hot_train,teacher_prefix_one_hot_train
               ,project_grade_category_one_hot_train,price_standardized_train,
quantity_standardized_train
               ,teacher_number_of_previously_posted_projects_standardized_trai
n,text_tfidf_train,title_tfidf_train))
# X_cr = hstack((categories_one_hot_cv,sub_categories_one_hot_cv,school_state_
one_hot_cv,teacher_prefix_one_hot_cv
#                ,project_grade_category_one_hot_cv,price_standardized_cv,quan
tity_standardized_cv
#                ,teacher_number_of_previously_posted_projects_standardized_c
v,text_tfidf_cv,title_tfidf_cv)).tocsr()
X_te = hstack((categories_one_hot_test,sub_categories_one_hot_test,school_stat
e_one_hot_test,teacher_prefix_one_hot_test
               ,project_grade_category_one_hot_test,price_standardized_test,qu
antity_standardized_test
               ,teacher_number_of_previously_posted_projects_standardized_test
,text_tfidf_test,title_tfidf_test))

print("Final Data matrix on TFIDF")
print(X_tr.shape, y_train.shape)
# print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix on TFIDF
(73196, 7745) (73196,)
(36052, 7745) (36052,)
====================================================================================
======================
```

In [5]:
```
X_te.shape
```

Out[5]:
```
(36052, 7745)
```

In [6]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code
# make sure you featurize train and test data separatly

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

2.4 Dimensionality Reduction on the selected features

In [7]:
```
###########################################################################
# from sklearn.preprocessing import MaxAbsScaler
# scaler = MaxAbsScaler()
# X_tr = scaler.fit_transform(X_tr,y_train)
# X_te = scaler.transform(X_te)
###########################################################################
from sklearn.feature_selection import SelectKBest, chi2
t = SelectKBest(chi2,k=5000).fit(X_tr, y_train)
X_tr = t.transform(X_tr)
X_te = t.transform(X_te)
###########################################################################
print("Final Data matrix on TFIDF")
print(X_tr.shape, y_train.shape)
# print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix on TFIDF
(73196, 5000) (73196,)
(36052, 5000) (36052,)
====================================================================================================
```

In [9]:
```
X_tr = X_tr[:5000]
X_train = X_train[:5000]
```

In [10]:
```
X_tr.shape
```

Out[10]:
```
(5000, 5000)
```

for k=2

In [33]:
```
from sklearn.cluster import AgglomerativeClustering

aggcl=AgglomerativeClustering(n_clusters=2).fit(X_tr.toarray())
```

In [75]:
```
cluster1=[]
cluster2=[]
essays = X_train['preprocessed_essays'].values
for i in range(aggcl.labels_.shape[0]):
    if aggcl.labels_[i] == 0:
        cluster1.append(essays[i])
    elif aggcl.labels_[i] == 1:
        cluster2.append(essays[i])
```

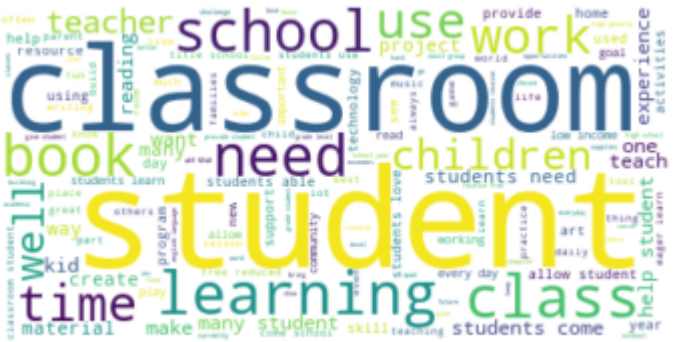
In [35]:
```
for i in range(3):
    print('%s\n'%(cluster1[i]))
```

```
my students diverse socioeconomic backgrounds ethnicities they hardworking wa
nt best i want provide stimulating environment encourages respect innovation
creativeness collaboration many students come school wanting learn want conti
nue enthusiastic learning our school reaches families family fun nights trave
ling neighborhoods tutoring sessions fun nights school build sense community
all children deserve learn based learning styles we several wiggly first grad
ers could benefit using wobble chairs i feel would engaged projects able wobb
le move around learning first graders spend much day participating literacy a
ctivities hence sit much this challenging young children i feel wobble chairs
help kids excited different seating options reading writing learning i excite
d get 21st century seating classroom thank considering donation

as educator every experience classes smart literally extra homework keep our
kindergarten classes done year year able grasp many first grade concepts lite
racy math most kindergarten kids live urban area baltimore yet despite realit
y oblivious condition they love love books i mean love books new math concept
s introduced see brains calculating working when lightbulb aha moment happens
i know i helped click learning however i not take credit introduced primed ri
ght love learning colleague pre k teacher many years if able meet chat wonder
kids would definitely 100 amazed love kids remember poem all i needed to know
i learned kindergarten sandwiched milk cookies taking nap key unlocking poten
tial arts integrated learning live balanced life learn think draw paint sing
dance play work every day the document camera chromebook combo donation allow
learners chance not visualize follow step step directions needed create origi
nal raps dances 3 d drawing painting projects new feature arts integrated exp
loration station

our upper elementary school serves 3rd 5th graders south carolina our student
s come high poverty social economic status 98 receiving free lunch a good maj
ority students receive backpack lunches weekends half students speak another
language home my students read 1 2 reading levels 3rd grade status my student
s lack background knowledge home resources education the majority class males
attention deficits girls need constant movement the kore wobble chairs allow
students move around never leave seat the chair helps develop balance control
the kore wobble chair life changer active kids kids adhd autistic kids aggres
sive wobbling kore chair increases blood flow brain thereby quiets requiremen
t fidgety kids move kids learn better productive the kid may considered un te
achable able learn flexible seating wobble chairs allow students work learn b
est students able choose chair desk works best
```

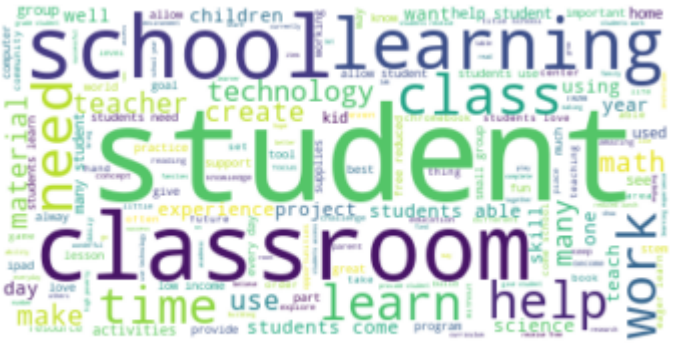
In [36]:
```
for i in range(3):
    print('%s\n'%(cluster2[i]))
```

```
within class i great diversity learners i never school including preschool ki
ndergarten second time finding ways differentiate bit difficult i want make s
ure students challenged not discouraged every student learns pace need suppor
t class although student different background different learning style amazin
g support they love come class knowing opportunity learn help teach one anoth
er the magic boards practice cards language learning centers math folder game
s students practice word building number recognition the sand alphabet letter
s allows students practice letter recognition fun time the parachute large di
ce alphabet bean bags give class opportunity build team building skills well
learning abc 123 your generous donation allow expand knowledge sweet kinderga
rtners build teamwork skills with support yacker tracker i able teach student
s manage voices working groups i also able provide extra one one practice stu
dent help become confident learning

we small rural school east texas high poverty rate low ses there no one way a
pproach learning message teachers try demonstrate daily students the greatest
challenge provide opportunities students reach greatest learning potential wi
thin allotted budget our team teaching approach math ela instruction strives
present intermediate students motivating modes learning school optimizing eng
agement in multi disciplinary classroom learning styles dictate students shor
t periods large group instruction several small even independent tasks made a
vailable student choice this ipad mini 4 shared among students small groups b
ecause passed around young students also needing protective case device on ip
ad student access folders containing apps address specific academic needs rea
ding math writing these skills closely monitor grade level teks highly motiva
ting it certainly improve engagement make learning fun

belle hall awesome school full wonderful learners over 600 students come comp
uter lab week we recently received 35 purposed laptops created new lab since
not 1 1 school means access technology students this new lab allow students a
ccess computer technology three times often our students technologically prep
ared move forward education our students need effectively use many computer p
rograms the copy paste feature touch pad much difficult use mouse young stude
nts students belle hall already comfortable mouse usage having pieces equipme
nt lab lab makes students less likely get frustrated computer without mice us
age lab probably not grow even though many students using touch screen device
s outside school not types devices belle hall mice must
```

In [37]:
```
#cluster 1
words=''
for i in cluster1:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



In [38]:
```
#cluster 2
words=''
for i in cluster2:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



for k=5

In [17]:
```
from sklearn.cluster import AgglomerativeClustering

aggcl=AgglomerativeClustering(n_clusters=5).fit(X_tr.toarray())
```

In [20]:
```
cluster1=[]
cluster2=[]
cluster3=[]
cluster4=[]
cluster5=[]
essays = X_train['preprocessed_essays'].values
for i in range(aggcl.labels_.shape[0]):
    if aggcl.labels_[i] == 0:
        cluster1.append(essays[i])
    elif aggcl.labels_[i] == 1:
        cluster2.append(essays[i])
    elif aggcl.labels_[i] == 2:
        cluster3.append(essays[i])
    elif  aggcl.labels_[i] == 3:
        cluster4.append(essays[i])
    elif  aggcl.labels_[i] == 4:
        cluster5.append(essays[i])
```

In [23]:
```
for i in range(3):
    print('%s\n'%(cluster1[i]))
```

```
within class i great diversity learners i never school including preschool ki
ndergarten second time finding ways differentiate bit difficult i want make s
ure students challenged not discouraged every student learns pace need suppor
t class although student different background different learning style amazin
g support they love come class knowing opportunity learn help teach one anoth
er the magic boards practice cards language learning centers math folder game
s students practice word building number recognition the sand alphabet letter
s allows students practice letter recognition fun time the parachute large di
ce alphabet bean bags give class opportunity build team building skills well
learning abc 123 your generous donation allow expand knowledge sweet kinderga
rtners build teamwork skills with support yacker tracker i able teach student
s manage voices working groups i also able provide extra one one practice stu
dent help become confident learning

we small rural school east texas high poverty rate low ses there no one way a
pproach learning message teachers try demonstrate daily students the greatest
challenge provide opportunities students reach greatest learning potential wi
thin allotted budget our team teaching approach math ela instruction strives
present intermediate students motivating modes learning school optimizing eng
agement in multi disciplinary classroom learning styles dictate students shor
t periods large group instruction several small even independent tasks made a
vailable student choice this ipad mini 4 shared among students small groups b
ecause passed around young students also needing protective case device on ip
ad student access folders containing apps address specific academic needs rea
ding math writing these skills closely monitor grade level teks highly motiva
ting it certainly improve engagement make learning fun

belle hall awesome school full wonderful learners over 600 students come comp
uter lab week we recently received 35 purposed laptops created new lab since
not 1 1 school means access technology students this new lab allow students a
ccess computer technology three times often our students technologically prep
ared move forward education our students need effectively use many computer p
rograms the copy paste feature touch pad much difficult use mouse young stude
nts students belle hall already comfortable mouse usage having pieces equipme
nt lab lab makes students less likely get frustrated computer without mice us
age lab probably not grow even though many students using touch screen device
s outside school not types devices belle hall mice must
```

In [24]:
```
for i in range(3):
    print('%s\n'%(cluster2[i]))
```

welcome intellectually gifted classroom i teach highly gifted disabled gifted we class culturally income diverse all students uniquely special i blessed in credible students my students hardworking big hearts they always giving back school community they come new school year rip roaring ready go i excited see students accomplish new school year at beginning year i start gifted class al l about me project students get express tell research name family origins del ve discovering individual natural talents this unit always breaks ice student s i get know new students personal level as part unit i lead students abstrac t art project includes initial my students love creating personal art pieces i display school constantly ask take home they not love creation process cher ish art pieces this meets gifted teacher standards learn fun process this i l ove teaching quest

i work low income elementary school limited funding comes technology my stude nts talented continually wish grow my students kind curious eager learn their love reading keeps library recess lunch the staff sees thirst education works hard every student comes because students not get chance use computers outsid e computer lab set computers library would excellent way broaden computer use inquisitive students this lenovo ideapad chromebook rotated throughout librar y students need access online catalog this promote independent searching mate rials when working small groups i also able deliver mini lessons keyboarding research using additional technology the practice students using computers sa fely responsibly better as students get comfortable using technology independ ently i would like open lab time library school allow students continue resea rch classroom projects reports students community rely heavily computer time school limited no access home

students come school eager learn explore they excited talk friends teacher at young age children acquire new skills every day my biggest challenge getting students attention keeping engaged activity play often talked relief serious learning but children play serious learning play really work childhood fred r ogers i teach 24 wonderful four five year old children love learning enjoy co ming school every day they happy inquisitive curious energetic love learn pla y i want students look forward coming school every day fun learn i want first experience school positive watch wonder class four year olds release butterfl y each child unique live adult butterfly release children learn life cycle bu tterfly listen stories butterflies look books completing puzzle butterfly inc rease spatial awareness fine motor skills the class collaboratively work colo ring illustrations the very hungry caterpillar having child take part process coloring illustrations give ownership child class make book meaningful childr en get paint paper butterfly watch color spreads water added diffusing paper finally children get eat caterpillar shaped cookie this fun filled butterfly themed day incorporate science reading math art it day children remember fore ver please help us bring project life

In [25]:
```
for i in range(3):
    print('%s\n'%(cluster3[i]))
```

```
my students diverse socioeconomic backgrounds ethnicities they hardworking wa
nt best i want provide stimulating environment encourages respect innovation
creativeness collaboration many students come school wanting learn want conti
nue enthusiastic learning our school reaches families family fun nights trave
ling neighborhoods tutoring sessions fun nights school build sense community
all children deserve learn based learning styles we several wiggly first grad
ers could benefit using wobble chairs i feel would engaged projects able wobb
le move around learning first graders spend much day participating literacy a
ctivities hence sit much this challenging young children i feel wobble chairs
help kids excited different seating options reading writing learning i excite
d get 21st century seating classroom thank considering donation

as educator every experience classes smart literally extra homework keep our
kindergarten classes done year year able grasp many first grade concepts lite
racy math most kindergarten kids live urban area baltimore yet despite realit
y oblivious condition they love love books i mean love books new math concept
s introduced see brains calculating working when lightbulb aha moment happens
i know i helped click learning however i not take credit introduced primed ri
ght love learning colleague pre k teacher many years if able meet chat wonder
kids would definitely 100 amazed love kids remember poem all i needed to know
i learned kindergarten sandwiched milk cookies taking nap key unlocking poten
tial arts integrated learning live balanced life learn think draw paint sing
dance play work every day the document camera chromebook combo donation allow
learners chance not visualize follow step step directions needed create origi
nal raps dances 3 d drawing painting projects new feature arts integrated exp
loration station

our school located rural community every student needs something different wa
lk classroom my job provide safe place students grow learners no matter backg
round coming about 60 students receive free reduced lunch our students come d
iver backgrounds they different life experiences these students going achieve
whatever set minds go far places life i love watching grow learners in first
grade kids like move lot having stability discs sit allow continue chairs cus
hions use would prefer students also like sit around classroom learning the r
ug would provide area students sit whole group lessons area students work ind
ependently i excited see student use new learning space helps improve learnin
g focus thank taking look first project
```

In [26]:
```
for i in range(3):
    print('%s\n'%(cluster4[i]))
```

my classroom slc consist low function students see world differently us they face challenges everyday hard time functioning others we title 1 school difficult get materials need keep minds hands busy my students live neighborhood low income rates not necessities nor materials need help better understand world my students need different materials regular my scholars learn best able exploder see world offer whether hands computer they however learn differently see world different light my students not grade levels different learning levels which challenging times i believe students necessities others students unfortunately title 1 school unable supply us materials need i would like teach students everyday living your donations greatly appreciated thank

in class multiply disabled students camden new jersey daily learning overcome obstacles break barriers many little experience world outside tiny socioeconomically depressed neighborhoods with proper technological resources sky limit camden poorest one dangerous cities america many students lack stable home environments school provides safe know my students little experience world outside tiny socioeconomically depressed neighborhoods with proper technological resources learning stations expand classroom experience outside walls classroom my students work lessons integrated across content differentiated meet specific needs interests learning styles that proper academic material important with proper supplies resources expand classroom experience outside walls classroom my students work lessons integrated across content differentiated meet specific needs interests learning styles technological literacy essential students competitive 21st century global economy as result integrating technology classroom instruction one top priorities good educator access technology docucam would allow students work cooperatively active participants learning process rather passive recipients information would ensure develop better depth knowledge concepts learning it also allows differentiation not basis academic levels interests well the round table chairs help provide setting conducive project based instruction using technology supplies collaborative learning environment researching writing mathematical application skills rather simple drills practice helps foster student creativity collaboration communication skills many employers demand 21st century workforce these items allow instruction highly customized students work independently active participants learning process rather passive recipients information

my students come school day eager learn the school title i school means 99 students poverty level they attend school six hundred students resources limited it hard get students bring two dollars class trip let alone anything else need supplies since special education class really benefit hands class work work environment allows get move around room whenever needed my students struggle sitting long period time tables these chairs bands help feeling fidgety like need move order stay focused by options available classroom better chance maintaining focus completing independent work these chairs bands also come handy small group centers the bands easy take even go student classroom services ensure stay focused well

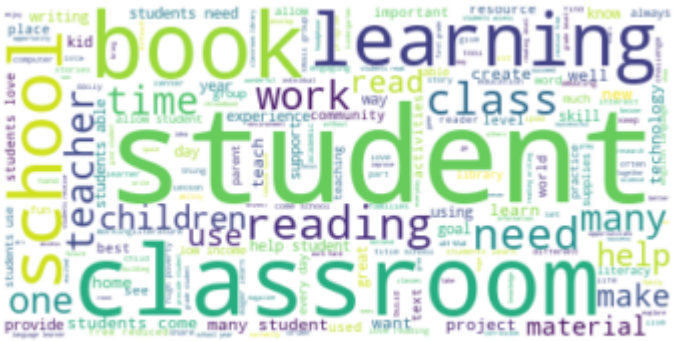
In [27]:
```
for i in range(3):
    print('%s\n'%(cluster5[i]))
```

```
our upper elementary school serves 3rd 5th graders south carolina our student
s come high poverty social economic status 98 receiving free lunch a good maj
ority students receive backpack lunches weekends half students speak another
language home my students read 1 2 reading levels 3rd grade status my student
s lack background knowledge home resources education the majority class males
attention deficits girls need constant movement the kore wobble chairs allow
students move around never leave seat the chair helps develop balance control
the kore wobble chair life changer active kids kids adhd autistic kids aggres
sive wobbling kore chair increases blood flow brain thereby quiets requiremen
t fidgety kids move kids learn better productive the kid may considered un te
achable able learn flexible seating wobble chairs allow students work learn b
est students able choose chair desk works best

i one two pe teachers title 1 school located florence sc our students come us
various backgrounds environments some two parent homes single family foster m
any raised grandparents even living hotels the positive environment provide s
chool positive thing many students see day everyday try greet kids smile posi
tive attitude when students enter gym ready move as pe teachers job encourage
set good example health wellness our students sit day play video games aftern
oon need encourage stay active the students classes tune today technology inc
luding fitbits they want monitor steps techy way better way encourage fitbit
zip technology made kids excited active want encourage way possible the kids
feel accomplished proud concrete way show fruits labor they excited show step
numbers compare compete friends steps many parents cannot afford latest techn
ology donation allow students opportunity try make lasting effect health well
we would use fitbits lead example track physical activity reward accomplishme
nts

our students come across city philadelphia entire school qualifies free lunch
transportation school based fact majority students live poverty level despite
issues facing students every day come school every day persevere mostly every
student attending college post graduation our students need support available
school district often cannot provide resources necessary students compete hig
h schools region my students discussed needs fitness room spinning machine on
e first ideas students they felt would help reach cardio goals given fact sch
ool not gym we limited space fitness room mainly weights not idea students an
spinning machine would meet every students needs allow students reach activit
y goals daily basis developing fitness room goal students since i hired schoo
l the student body given several goals wish accomplish development fitness ce
nter make accessible appealing students school
```

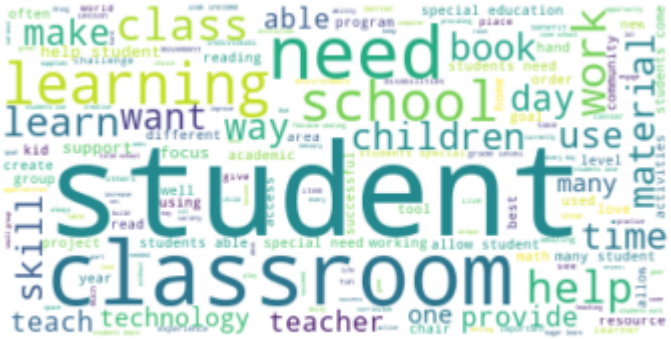
In [28]:
```
#cluster 1
words=''
for i in cluster1:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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

In [29]:
```
#cluster 2
words=''
for i in cluster2:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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

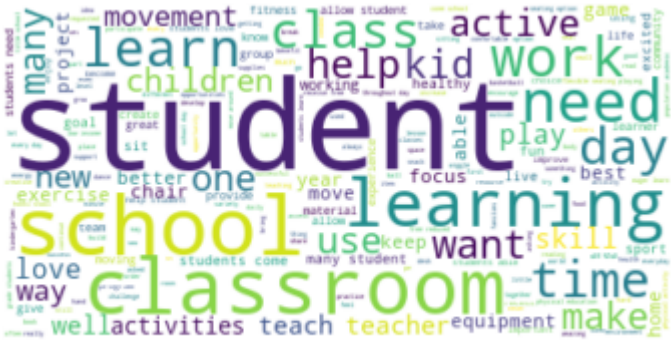
In [30]:
```
#cluster 3
words=''
for i in cluster3:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



In [31]:
```
#cluster 4
words=''
for i in cluster4:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



In [32]:
```
#cluster 5
words=''
for i in cluster5:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



In [ ]:
```
```

In [0]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the rea
der
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

2.7 Apply DBSCAN

In [1]:
```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import dill
# dill.dump_session('notebook_env.db')
dill.load_session('notebook_env.db')
```

```
C:\Users\LENOVO\Anaconda3\lib\site-packages\gensim\utils.py:1197: UserWarnin
g: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

In [2]:
```
project_data.columns
```

Out[2]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', 'project_title',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay', 'price', 'quantit
y',
       'digits_in_summary', 'clean_project_grade_category',
       'preprocessed_essays', 'preprocessed_titles'],
      dtype='object')
```

In [ ]:
```

```

In [3]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr = hstack((categories_one_hot_train,sub_categories_one_hot_train,school_st
ate_one_hot_train,teacher_prefix_one_hot_train
               ,project_grade_category_one_hot_train,price_standardized_train,
quantity_standardized_train
               ,teacher_number_of_previously_posted_projects_standardized_trai
n,text_tfidf_train,title_tfidf_train))
# X_cr = hstack((categories_one_hot_cv,sub_categories_one_hot_cv,school_state_
one_hot_cv,teacher_prefix_one_hot_cv
#                ,project_grade_category_one_hot_cv,price_standardized_cv,quan
tity_standardized_cv
#                ,teacher_number_of_previously_posted_projects_standardized_c
v,text_tfidf_cv,title_tfidf_cv)).tocsr()
X_te = hstack((categories_one_hot_test,sub_categories_one_hot_test,school_stat
e_one_hot_test,teacher_prefix_one_hot_test
               ,project_grade_category_one_hot_test,price_standardized_test,qu
antity_standardized_test
               ,teacher_number_of_previously_posted_projects_standardized_test
,text_tfidf_test,title_tfidf_test))

print("Final Data matrix on TFIDF")
print(X_tr.shape, y_train.shape)
# print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix on TFIDF
(73196, 7745) (73196,)
(36052, 7745) (36052,)
====================================================================================
======================
```

In [4]:
```
X_te.shape
```

Out[4]:
```
(36052, 7745)
```

In [5]:
```
# please write all the code with proper documentation, and proper titles for each subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debugging your code
# make sure you featurize train and test data separatly

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

2.4 Dimensionality Reduction on the selected features

In [6]:
```
#########################################################################
# from sklearn.preprocessing import MaxAbsScaler
# scaler = MaxAbsScaler()
# X_tr = scaler.fit_transform(X_tr,y_train)
# X_te = scaler.transform(X_te)
#########################################################################
from sklearn.feature_selection import SelectKBest, chi2
t = SelectKBest(chi2,k=5000).fit(X_tr, y_train)
X_tr = t.transform(X_tr)
X_te = t.transform(X_te)
#########################################################################
print("Final Data matrix on TFIDF")
print(X_tr.shape, y_train.shape)
# print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix on TFIDF
(73196, 5000) (73196,)
(36052, 5000) (36052,)
====================================================================================================
```

In [7]:
```
X_tr = X_tr[:5000]
X_train = X_train[:5000]
```

In [8]:
```
X_tr.shape
```

Out[8]:
```
(5000, 5000)
```

In [9]:
```
from sklearn.preprocessing import StandardScaler
# dat=StandardScaler().fit_transform(X_tr.toarray())
dat=X_tr.toarray()
dat
```

Out[9]:
```
array([[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.]])
```

In [10]:
```
from sklearn.metrics.pairwise import euclidean_distances

euclidean_distances(dat, dat[1].reshape(1, -1))
```

Out[10]:
```
array([[3.13236693e+00],
       [4.21468485e-08],
       [3.20124722e+00],
       ...,
       [3.10522436e+00],
       [3.27882999e+00],
       [3.19873778e+00]])
```

In [27]:
```
# sorted_distance = np.sort(np.array(distance))
# len(sorted_distance)
```

Out[27]:
```
806
```

In [27]:
```
# temp=np.sort(np.sum((dat-point),axis=1),axis=None)
# (temp).shape
```

Out[27]:
```
(5000,)
```

In [29]:
```
# np.sum((dat-point),axis=1)
```

Out[29]:
```
array([1.64712652, 2.30341677, 1.58066117, ..., 2.93591794, 1.03866231,
       0.        ])
```

In [62]:
```
# tt=(dat-point)**2
# tt[2000]
```

Out[62]:
```
array([0., 0., 0., ..., 0., 0., 0.])
```

In [59]:
```
# np.sum((dat-point)**2,axis=1)
```

Out[59]:
```
array([11.70919891, 10.23192341, 10.30200722, ..., 11.59827737,
        8.73397533,  0.        ])
```

In [30]:
```
# distance
```

Out[30]:
```
[array([104.26679466]),
 array([97.61290802]),
 array([92.24469259]),
 array([94.72413144]),
 array([91.92699148]),
 array([101.63258988]),
 array([94.72665935]),
 array([111.4489379]),
 array([92.75662884]),
 array([94.62546444]),
 array([106.28762275]),
 array([113.50326139]),
 array([104.62735836]),
 array([95.92549774]),
 array([98.26058222]),
 array([109.9701546]),
 array([91.7236505]),
 array([104.21250268]),
 array([90.0890306]),
 array([117.76242536]),
 array([94.69313352]),
 array([91.97822737]),
 array([92.95772527]),
 array([95.75440967]),
 array([91.00358232]),
 array([125.25245375]),
 array([96.55960104]),
 array([93.61834726]),
 array([99.3178604]),
 array([110.71166597]),
 array([90.82297469]),
 array([95.86445436]),
 array([84.95199019]),
 array([93.56511226]),
 array([99.80492315]),
 array([95.50944315]),
 array([99.72099326]),
 array([96.18770598]),
 array([97.21200634])]
```

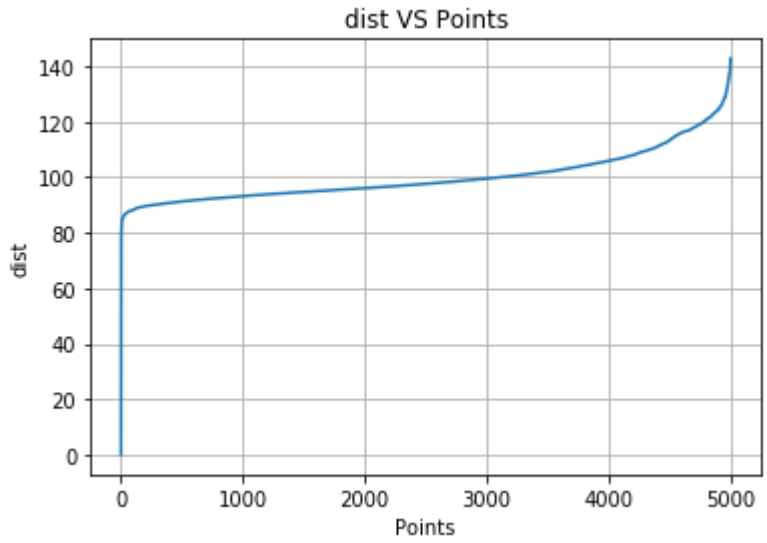
In [21]:
```
from sklearn.metrics.pairwise import euclidean_distances

euclidean_distances(dat, dat[464].reshape(1, -1))
```

Out[21]:
```
array([[3.6456085 ],
       [2.77862866],
       [3.41598141],
       ...,
       [2.60627774],
       [3.15575795],
       [3.12398718]])
```

In [60]:
```
# np.sort(sorted_dist[:50].reshape(1,-1))[0]
```

Out[60]:
```
array([ 84.95199019,  90.0890306 ,  90.82297469,  91.00358232,
        91.7236505 ,  91.92699148,  91.97822737,  92.24469259,
        92.75662884,  92.91832753,  92.95772527,  93.56511226,
        93.61834726,  94.5552292 ,  94.62546444,  94.69313352,
        94.72413144,  94.72665935,  95.48144355,  95.50944315,
        95.75440967,  95.86445436,  95.92549774,  96.18770598,
        96.33094241,  96.41486617,  96.55960104,  97.21200634,
        97.61290802,  98.26058222,  99.3178604 ,  99.66575521,
        99.72099326,  99.80492315, 101.63258988, 103.80911931,
       104.21250268, 104.26679466, 104.62735836, 106.28762275,
       108.84403901, 109.9701546 , 110.71166597, 111.4489379 ,
       112.27431254, 113.50326139, 116.28452388, 116.53726028,
       117.76242536, 125.25245375])
```

In [10]:
```
min_points = 1500
from sklearn.preprocessing import StandardScaler
from sklearn.metrics.pairwise import euclidean_distances
datt=StandardScaler().fit_transform(dat)

distance=[]
for point in tqdm1(datt):
    temp = euclidean_distances(datt, point.reshape(1, -1))
    distance.append(temp[min_points])
sorted_distance = np.sort(np.array(distance))

sorted_dist = np.sort(sorted_distance.reshape(1,-1)[0])
points = [i for i in range(len(datt))]


# Draw distances(d_i) VS points(x_i) plot
plt.plot(points, sorted_dist)
plt.xlabel('Points')
plt.ylabel('dist')
plt.title('dist VS Points')
plt.grid()
plt.show()
```



In [73]:
```
#we can see that point of inflexion is at eps=9
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=90,n_jobs=-1)
dbscan.fit(datt)
print('No of clusters: ',len(set(dbscan.labels_)))
print('Cluster are including noise i.e -1: ',set(dbscan.labels_))
```

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

In [76]:
```
#ignoring -1 as it is for noise
cluster1=[]
noisecluster1=[]
for i in range(dbscan.labels_.shape[0]):
    if dbscan.labels_[i] == 0:
        cluster1.append(essays[i])
    elif dbscan.labels_[i] == -1:
        noisecluster1.append(essays[i])
```

In [77]:
```
for i in range(3):
    print('%s\n'%(cluster1[i]))
```

```
my students diverse socioeconomic backgrounds ethnicities they hardworking wa
nt best i want provide stimulating environment encourages respect innovation
creativeness collaboration many students come school wanting learn want conti
nue enthusiastic learning our school reaches families family fun nights trave
ling neighborhoods tutoring sessions fun nights school build sense community
all children deserve learn based learning styles we several wiggly first grad
ers could benefit using wobble chairs i feel would engaged projects able wobb
le move around learning first graders spend much day participating literacy a
ctivities hence sit much this challenging young children i feel wobble chairs
help kids excited different seating options reading writing learning i excite
d get 21st century seating classroom thank considering donation

within class i great diversity learners i never school including preschool ki
ndergarten second time finding ways differentiate bit difficult i want make s
ure students challenged not discouraged every student learns pace need suppor
t class although student different background different learning style amazin
g support they love come class knowing opportunity learn help teach one anoth
er the magic boards practice cards language learning centers math folder game
s students practice word building number recognition the sand alphabet letter
s allows students practice letter recognition fun time the parachute large di
ce alphabet bean bags give class opportunity build team building skills well
learning abc 123 your generous donation allow expand knowledge sweet kinderga
rtners build teamwork skills with support yacker tracker i able teach student
s manage voices working groups i also able provide extra one one practice stu
dent help become confident learning

as educator every experience classes smart literally extra homework keep our
kindergarten classes done year year able grasp many first grade concepts lite
racy math most kindergarten kids live urban area baltimore yet despite realit
y oblivious condition they love love books i mean love books new math concept
s introduced see brains calculating working when lightbulb aha moment happens
i know i helped click learning however i not take credit introduced primed ri
ght love learning colleague pre k teacher many years if able meet chat wonder
kids would definitely 100 amazed love kids remember poem all i needed to know
i learned kindergarten sandwiched milk cookies taking nap key unlocking poten
tial arts integrated learning live balanced life learn think draw paint sing
dance play work every day the document camera chromebook combo donation allow
learners chance not visualize follow step step directions needed create origi
nal raps dances 3 d drawing painting projects new feature arts integrated exp
loration station
```

In [78]:
```
for i in range(3):
    print('%s\n'%(noisecluster1[i]))
```

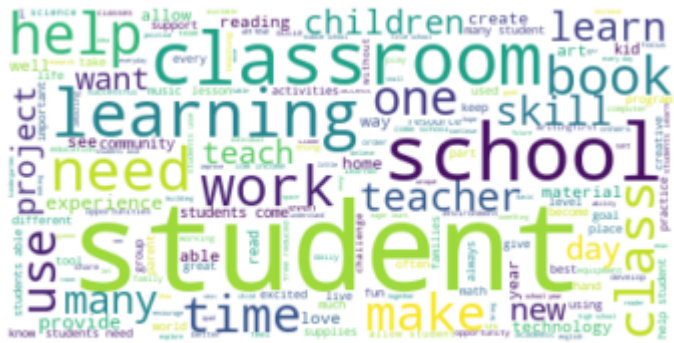
my students hungry meaning they want learn learning connects real world work improves world everyone i found literature helpful bridge meaning science classrooms these kids need read science classroom when learn science concepts story helps make unknown aspects science come life this also improves literacy skills along science math engineering technology knowledge know in order students understand importance science math engineering technology stem education need context the martian classroom edition takes world setting brings vivid detail the mathematical problems exposed book offer incredible opportunities student teams figure answers the engineering feats offer exceptional launching points design challenges students solve all students rooting hero make safely back home students read the martian classroom edition create curricular web support robust pbl project the full pbl broken following components science focusing physics space travel sustaining life outside earth engineering project duplicating one problems exposed story persuasive writing section based analysis book

my students english language learners non english speakers they live poverty line parents speak spanish limited english although many students born united states families came try give better life economically education better future our school elementary school serves students kindergarten second grade i one class amazing fantastic children eager learn english there no equality treatment merely providing students facilities textbooks teachers curriculum students not understand english effectively foreclosed meaningful education i respectfully request letter learning activity carpet students i feel would benefit greatly spanish first language going school new experience little fear apprehension enjoying school friends kindergarten family gives sense trust helps students transition easier coming home happy classroom by clean colorful fun learning safe environment full love mere joy learning english build confidence self esteem class structure sit learn if child heart head

students come school eager learn explore they excited talk friends teacher at young age children acquire new skills every day my biggest challenge getting students attention keeping engaged activity play often talked relief serious learning but children play serious learning play really work childhood fred rogers i teach 24 wonderful four five year old children love learning enjoy coming school every day they happy inquisitive curious energetic love learn play i want students look forward coming school every day fun learn i want first experience school positive watch wonder class four year olds release butterfly each child unique live adult butterfly release children learn life cycle butterfly listen stories butterflies look books completing puzzle butterfly increase spatial awareness fine motor skills the class collaboratively work coloring illustrations the very hungry caterpillar having child take part process coloring illustrations give ownership child class make book meaningful children get paint paper butterfly watch color spreads water added diffusing paper finally children get eat caterpillar shaped cookie this fun filled butterfly themed day incorporate science reading math art it day children remember forever please help us bring project life

In [79]:
```
#cluster 1
words=''
for i in cluster1:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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

In [80]:
```
#noise cluster 1
words=''
for i in noisecluster1:
    words+=str(i)
from wordcloud import WordCloud
wordcloud = WordCloud(background_color="white").generate(words)

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



In [45]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the rea
der
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

3. Cocnlusions

Please write down few lines of your observations on this assignment.

1.K-Means:

1.First i hyperparameter tuned K_values with 2,3,4,5,6,7,8 and got the inflection point at k=6

2.Then i trained K-Means on K_value=6

3.After training i clustered the essays into 6 seperate clusters

4.Then i plotted the word cloud

2.Agglomerative:

1.First i reduced the dimentions to 5000 and also took 5000 points same as in K-Means

2.Then i applied Agglomerative clustering on k=2

3.Then i clustered the essays into 2 seperate clusters

4.After that i plotted the wordcloud for each of the clusters

5.Then i applied Agglomerative clustering on k=5

6.Then i clustered the essays into 5 seperate clusters

7.After that i plotted the wordcloud for each of the clusters

3.DBScan:

1.First i converted the reduced sparce matrix to dense using toarray()

2.Then i transformed the data to standard scalar

3.Then i computed euclidean distance for every point to every other point and took the distance of min_pts

4.Obtained the best eps to be 90 from the above graph b/w dist and points

5.Then formed clusters on noise points and non-noise points

6.Printed the essays in each of the two clusters

7.Then printed the wordcloud.

Pretty Table

K-Means:

In [2]:
```
#prettytable for kmeans
from prettytable import PrettyTable
x = PrettyTable()
x.field_names = ["Vectorizer","Best k"]
x.add_row(['TFIDF','6'])
print(x)
```

```
+------------+--------+
| Vectorizer | Best k |
+------------+--------+
|   TFIDF    |   6    |
+------------+--------+
```

Agglomerative:

In [3]:
```
#prettytable for kmeans
from prettytable import PrettyTable
x = PrettyTable()
x.field_names = ["Vectorizer","Best k"]
x.add_row(['TFIDF','2'])
x.add_row(['TFIDF','5'])
print(x)
```

```
+------------+--------+
| Vectorizer | Best k |
+------------+--------+
|   TFIDF    |   2    |
|   TFIDF    |   5    |
+------------+--------+
```

DBScan:

In [5]:
```
#prettytable for kmeans
from prettytable import PrettyTable
x = PrettyTable()
x.field_names = ["Vectorizer","Best k","Eps","Number of clusters(INCLUDING NOI
SE)"]
x.add_row(['TFIDF','2',90,2])
print(x)
```

```
+------------+--------+-----+--------------------------------------+
| Vectorizer | Best k | Eps | Number of clusters(INCLUDING NOISE) |
+------------+--------+-----+--------------------------------------+
|   TFIDF    |   2    |  90 |                  2                   |
+------------+--------+-----+--------------------------------------+
```

In [ ]:
```

```