# DonorsChoose

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

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

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

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

# About the DonorsChoose Data Set

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

| Feature | Description |
|---|---|
| **`project_id`** | A unique identifier for the proposed project. **Exam** `p036502` |
| **`project_title`** | Title of the project. **Examples:**<br>• `Art Will Make You Happy!`<br>• `First Grade Fun` |
| **`project_grade_category`** | Grade level of students for which the project is targeted. One of the following enumerated values<br>• `Grades PreK-2`<br>• `Grades 3-5`<br>• `Grades 6-8`<br>• `Grades 9-12` |
| **`project_subject_categories`** | One or more (comma-separated) subject categor the project from the following enumerated list of values:<br>• `Applied Learning`<br>• `Care & Hunger`<br>• `Health & Sports`<br>• `History & Civics`<br>• `Literacy & Language`<br>• `Math & Science`<br>• `Music & The Arts`<br>• `Special Needs`<br>• `Warmth`<br>**Examples:**<br>• `Music & The Arts`<br>• `Literacy & Language, Math & Science` |
| **`school_state`** | State where school is located (Two-letter U.S. pos code (https://en.wikipedia.org /wiki/List_of_U.S._state_abbreviations#Postal_co **Example:** `WY` |

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

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

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

| Feature | Description |
|---|---|
| **id** | A `project_id` value from the `train.csv` file. **Example:** `p036502` |

| Feature | Description |
|---|---|
| **`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):

## 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 [128]:
```
%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

import chart_studio.plotly as py
import plotly.graph_objs as go
from collections import Counter
```

# 1.1 Reading Data

In [129]:
```
project_data = pd.read_csv('train_data.csv',nrows=30000)
resource_data = pd.read_csv('resources.csv')
```

In [130]:
```
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 (30000, 17)
--------------------------------------------------
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_pre
fix' 'school_state'
 'project_submitted_datetime' 'project_grade_category'
 'project_subject_categories' 'project_subject_subcategories'
 'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3
'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects' 'project_is_approved
']
```

In [131]:
```
project_data["project_is_approved"].value_counts()
```

Out[131]:
```
1    25380
0     4620
Name: project_is_approved, dtype: int64
```

In [132]:
```
# how to replace elements in list python: https://stackoverflow.com/a/2582163/4084039
cols = ['Date' if x=='project_submitted_datetime' else x for x in list(project_data.columns)]

#sort dataframe based on time pandas python: https://stackoverflow.com/a/49702492/4084039
project_data['Date'] = pd.to_datetime(project_data['project_submitted_datetime'])
project_data.drop('project_submitted_datetime', axis=1, inplace=True)
project_data.sort_values(by=['Date'], inplace=True)

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

project_data.head(2)
```

Out[132]:

| | Unnamed: 0 | id | teacher_id | teacher_prefix | schoo |
|---|---|---|---|---|---|
| **473** | 100660 | p234804 | cbc0e38f522143b86d372f8b43d4cff3 | Mrs. | GA |
| **29891** | 146723 | p099708 | c0a28c79fe8ad5810da49de47b3fb491 | Mrs. | CA |

In [133]:
```
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[133]:

| | 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 [134]:
```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stack
overflow.com/a/47301924/4084039

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

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

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 [135]:
```
sub_catogories = list(project_data['project_subject_subcategories'].va
lues)
# remove special characters from list of strings python: https://stack
overflow.com/a/47301924/4084039

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

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

project_data['clean_subcategories'] = sub_cat_list
project_data.drop(['project_subject_subcategories'], axis=1, inplace=T
rue)

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

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

# 1.3 Text preprocessing

In [136]:
```
# 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 [137]:
```
project_data.head(2)
```

Out[137]:

| | Unnamed: 0 | id | teacher_id | teacher_prefix | schoo |
|---|---|---|---|---|---|
| **473** | 100660 | p234804 | cbc0e38f522143b86d372f8b43d4cff3 | Mrs. | GA |
| **29891** | 146723 | p099708 | c0a28c79fe8ad5810da49de47b3fb491 | Mrs. | CA |

In [138]:
```
#### 1.4.2.3 Using Pretrained Models: TFIDF weighted W2V
```

In [139]:
```
# printing some random reviews
print(project_data['essay'].values[0])
print("="*50)
print(project_data['essay'].values[150])
print("="*50)
print(project_data['essay'].values[1000])
print("="*50)
#print(project_data['essay'].values[20000])
#print("="*50)
#print(project_data['essay'].values[99999])
#print("="*50)
```

I recently read an article about giving students a choice about how t
hey learn. We already set goals; why not let them choose where to si
t, and give them options of what to sit on?I teach at a low-income (T
itle 1) school. Every year, I have a class with a range of abilities,
yet they are all the same age. They learn differently, and they have
different interests. Some have ADHD, and some are fast learners. Yet
they are eager and active learners that want and need to be able to m
ove around the room, yet have a place that they can be comfortable to
complete their work.We need a classroom rug that we can use as a clas
s for reading time, and students can use during other learning times.
I have also requested four Kore Kids wobble chairs and four Back Jack
padded portable chairs so that students can still move during whole g
roup lessons without disrupting the class. Having these areas will pr
ovide these little ones with a way to wiggle while working.Benjamin F
ranklin once said, \"Tell me and I forget, teach me and I may remembe
r, involve me and I learn.\" I want these children to be involved in
their learning by having a choice on where to sit and how to learn, a
ll by giving them options for comfortable flexible seating.
==========
Do you remember working hard towards that special incentive or rewar
d? Remember how great it felt and how proud you were when you finall
y earned it? I have the opportunity to work with a large variety of
students who struggle with academic and behavioral challenges in my e
lementary school.My students are diverse in their grade levels as wel
l as backgrounds, who attend a primarily military school. It is a tr
ansitional environment, therefore many of my students have difficulti
es with making good choices due to deployments, moving, and family st
ruggles.As the School Behavior Health Specialist, I work with student
s from kindergarten to 5th grade. These students come me with a gamu
t of challenges, both academic, behavioral, emotional and social. I
work with the students and their teacher to develop behavioral plans
to maximize success in the classroom. These rewards are essential to
motivate students to make good choices.These incentive materials will
help to impact behavioral in these students which leads to positive c
hanges in their live. Many students are able to feel proud when they
reach their goals and learn that they too can be successful. Howeve
r, without gracious donations, the high reward incentives are very li
mited. Any donations are greatly appreciated! Mahalo!
==========
“Attitude is everything!” This quote best describes my classroom. My
students have learning disabilities in reading fluency and/or compreh
ension. They have significant struggles in reading. Our goal in class
is to improve reading levels, and build confidence in their reading s
kills.I teach at a Title I middle school in an urban neighborhood. 7
4% of our students qualify for free or reduced, rate lunch and many c
ome from very technology-poor homes. My students struggle with grade
level learning, as all of them have some type of learning disability.
My students learn better when they can move around or stand, they bec
ome more focused on the task at hand. I have seen great focus in my s
tudents when they are given the opportunity to work with a clipboard
while standing against the wall. My goal is to create a classroom env
ironment where students can continue to work at their desks, while gi
ven the opportunity to learn the way that best motivates them.\r\nStu
dents will be using the Stability Balls to promote an active learning
environment. Most of my students have a hard time sitting still for a
ny amount of time, and the Stability Balls will give them the opportu

```
nity to stand at their desks while getting the materials they need to
learn to be successful. Students will use the Stability Balls for ind
ividual assignments, partner work, and small group work. My students
will never feel like they are glued to their desks, but have freedom
to move without distracting others around them.My students struggle w
ith sitting still for long periods at a time (15 minutes or more), th
e Stability Balls will give them the opportunity to move freely at th
eir desks by rocking back and forth while they learn. Students will n
o longer have to ask permission to stand in the back or off to the si
de while they listen, as they can just quietly move freely around and
work without the distractions. Students will be more focused if given
the opportunity to freely move at their desks.
==================================================
```

In [140]:
```
# 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 [141]:
```
sent = decontracted(project_data['essay'].values[20000])
print(sent)
print("="*50)
```

```
I have 63 students in three different math classes in a high poverty
school. Many of these students come from single family homes and ofte
n stay with grandparents while their parent/parents work. \r\n\r\nEve
n though my students face many challenges, they are eager to learn ne
w math concepts. I want to continue to give them the opportunity to l
earn their math concepts that will help them achieve a foundation to
be successful in their future math journey. \r\n\r\nThank you for hel
ping our future mathematicians succeed!Math is challenging. Students
learn from making mistakes but with paper and pencils students at tim
es erase so hard that they tear their paper. With the use of white bo
ards and dry erase markers students can erase over and over  without
having to worry about tearing their paper. \r\n\r\nWhite boards are a
great tool to use in the classroom that will allow students to show t
heir work from their seat and allow for some individualization and gr
oup work. \r\n\r\nStudents enjoy writing with a variety of writing to
ols and with the use of white boards and dry erase markers, I feel th
at my students will have a new excitement in learning math.nannan
==================================================
```

In [142]:
```
# \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)
```

```
I have 63 students in three different math classes in a high poverty
school. Many of these students come from single family homes and ofte
n stay with grandparents while their parent/parents work.     Even th
ough my students face many challenges, they are eager to learn new ma
th concepts. I want to continue to give them the opportunity to learn
their math concepts that will help them achieve a foundation to be su
ccessful in their future math journey.     Thank you for helping our
future mathematicians succeed!Math is challenging. Students learn fro
m making mistakes but with paper and pencils students at times erase
so hard that they tear their paper. With the use of white boards and
dry erase markers students can erase over and over  without having to
worry about tearing their paper.     White boards are a great tool to
use in the classroom that will allow students to show their work from
their seat and allow for some individualization and group work.     S
tudents enjoy writing with a variety of writing tools and with the us
e of white boards and dry erase markers, I feel that my students will
have a new excitement in learning math.nannan
```

In [143]:
```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
print(sent)
```

```
I have 63 students in three different math classes in a high poverty
school Many of these students come from single family homes and often
stay with grandparents while their parent parents work Even though my
students face many challenges they are eager to learn new math concep
ts I want to continue to give them the opportunity to learn their mat
h concepts that will help them achieve a foundation to be successful
in their future math journey Thank you for helping our future mathema
ticians succeed Math is challenging Students learn from making mistak
es but with paper and pencils students at times erase so hard that th
ey tear their paper With the use of white boards and dry erase marker
s students can erase over and over without having to worry about tear
ing their paper White boards are a great tool to use in the classroom
that will allow students to show their work from their seat and allow
for some individualization and group work Students enjoy writing with
a variety of writing tools and with the use of white boards and dry e
rase markers I feel that my students will have a new excitement in le
arning math nannan
```

In [144]:
```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'no
t'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves
', 'you', "you're", "you've",\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselve
s', 'he', 'him', 'his', 'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'i
ts', 'itself', 'they', 'them', 'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 't
his', 'that', "that'll", 'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', '
have', 'has', 'had', 'having', 'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or
', 'because', 'as', 'until', 'while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between',
'into', 'through', 'during', 'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
'on', 'off', 'over', 'under', 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', '
how', 'all', 'any', 'both', 'each', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 's
o', 'than', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should',
"should've", 'now', 'd', 'll', 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't",
'didn', "didn't", 'doesn', "doesn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "is
n't", 'ma', 'mightn', "mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn
', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [145]:
```
# 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.lower() not in stopwo
rds)
    preprocessed_essays.append(sent.lower().strip())
```

```
100%|██████████| 30000/30000 [00:44<00:00, 681.45it/s]
```

In [146]:
```
# after preprocesing
#creating a new column with the preprocessed essays and replacing it w
ith the original columns
project_data['preprocessed_essays'] = preprocessed_essays
project_data.drop(['project_essay_1'], axis=1, inplace=True)
project_data.drop(['project_essay_2'], axis=1, inplace=True)
project_data.drop(['project_essay_3'], axis=1, inplace=True)
project_data.drop(['project_essay_4'], axis=1, inplace=True)
preprocessed_essays[20000]
```

Out[146]:
```
'63 students three different math classes high poverty school many st
udents come single family homes often stay grandparents parent parent
s work even though students face many challenges eager learn new math
concepts want continue give opportunity learn math concepts help achi
eve foundation successful future math journey thank helping future ma
thematicians succeed math challenging students learn making mistakes
paper pencils students times erase hard tear paper use white boards d
ry erase markers students erase without worry tearing paper white boa
rds great tool use classroom allow students show work seat allow indi
vidualization group work students enjoy writing variety writing tools
use white boards dry erase markers feel students new excitement learn
ing math nannan'
```

## 1.4 Preprocessing of `project_title`

In [147]:
```
# Combining all the above statemennts
from tqdm import tqdm
preprocessed_titles = []
# tqdm is for printing the status bar
for sentance in tqdm(project_data['project_title'].values):
    sent = decontracted(sentance)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = 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_titles.append(sent.lower().strip())
```

100%|██████████| 30000/30000 [00:02<00:00, 10326.99it/s]

In [148]:
```
#creating a new column with the preprocessed titles,useful for analysi
s
project_data['preprocessed_titles'] = preprocessed_titles
```

In [149]:
```
#
------------------------------------------------------------------------
-----------------
# Preprocessing Categorical Features: teacher_prefix
print(project_data['teacher_prefix'].value_counts())
print("="*100)

print(project_data[project_data['teacher_prefix'].isnull()]['teacher_p
refix'])

print("="*100)
project_data['teacher_prefix']=project_data['teacher_prefix'].fillna('
Mrs.')
print(project_data['teacher_prefix'].value_counts())

print("="*100)
#
------------------------------------------------------------------------
-----------------
```

```
Mrs.       15682
Ms.        10779
Mr.         2895
Teacher      643
Name: teacher_prefix, dtype: int64
====================================================================
================================
7820    NaN
Name: teacher_prefix, dtype: object
====================================================================
================================
Mrs.       15683
Ms.        10779
Mr.         2895
Teacher      643
Name: teacher_prefix, dtype: int64
====================================================================
================================
```

In [150]:
```
#
------------------------------------------------------------------------
--------------------------------
# Preprocessing Categorical Features: project_grade_category
project_data['project_grade_category'] = project_data['project_grade_c
ategory'].str.replace(' ', '_')
project_data['project_grade_category'] = project_data['project_grade_c
ategory'].str.replace('-', '_')
project_data['project_grade_category'] = project_data['project_grade_c
ategory'].str.lower()
#
------------------------------------------------------------------------
--------------------------------
```

# Splitting data into Train, cross validation and test: Stratified Sampling

In [151]:
```
from sklearn.model_selection import train_test_split
#How to split whole dataset into Train,CV and test
#https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html#sklearn.model_selection.train_test_split
project_data_train, project_data_test, y_train, y_test = train_test_split(project_data, project_data['project_is_approved'], test_size=0.33, stratify = project_data['project_is_approved'])
print(project_data_train.shape,project_data_test.shape, y_train.shape, y_test.shape)
```

```
(20100, 16) (9900, 16) (20100,) (9900,)
```

In [152]:
```
print("Split ratio")
print('-'*50)
print('Train dataset:',len(project_data_train)/len(project_data)*100,'%\n','size:',len(project_data_train))
print('Test dataset:',len(project_data_test)/len(project_data)*100,'%\n','size:',len(project_data_test))
```

```
Split ratio
--------------------------------------------------
Train dataset: 67.0 %
 size: 20100
Test dataset: 33.0 %
 size: 9900
```

# 1.5 Preparing data for models

In [153]:
```
project_data.columns
```

Out[153]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'Date', 'project_grade_category', 'project_title',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay',
       'preprocessed_essays', 'preprocessed_titles'],
      dtype='object')
```

## Vectorizing Categorical data Using Response Coding

In [154]:
```
def Responsetable(table, col) :
    cat = table[col].unique()

    freq_Pos = []
    for i in cat :
        freq_Pos.append(len(table.loc[(table[col] == i) & (table['proj
ect_is_approved'] == 1)]))


    freq_Neg = []
    for i in cat :
        freq_Neg.append(len(table.loc[(table[col] == i) & (table['proj
ect_is_approved'] == 0)]))


    encoded_Pos = []
    for i in range(len(cat)) :
        encoded_Pos.append(freq_Pos[i]/(freq_Pos[i] + freq_Neg[i]))

    encoded_Neg = []
    encoded_Neg[:] = [1 - x for x in encoded_Pos]

    encoded_Pos_val = dict(zip(cat, encoded_Pos))
    encoded_Neg_val = dict(zip(cat, encoded_Neg))

    return encoded_Pos_val, encoded_Neg_val
```

In [155]:
```
def Responsecode(table) :
    pos_cleancat, neg_cleancat = Responsetable(table,'clean_categories
')
    pos_cleansubcat, neg_cleansubcat = Responsetable(table,'clean_subc
ategories')
    pos_schoolstate, neg_schoolstate = Responsetable(table, 'school_st
ate')
    pos_teacherprefix, neg_teacherprefix = Responsetable(table, 'teach
er_prefix')
    pos_projgradecat, neg_projgradecat = Responsetable(table, 'project
_grade_category')

    df = pd.DataFrame()
    df['clean_cat_pos'] = table['clean_categories'].map(pos_cleancat)
    df['clean_cat_neg'] = table['clean_categories'].map(neg_cleancat)
    df['clean_subcat_pos'] = table['clean_subcategories'].map(pos_clea
nsubcat)
    df['clean_subcat_neg'] = table['clean_subcategories'].map(neg_clea
nsubcat)
    df['school_state_pos'] = table['school_state'].map(pos_schoolstat
e)
    df['school_state_neg'] = table['school_state'].map(neg_schoolstat
e)
    df['teacher_prefix_pos'] = table['teacher_prefix'].map(pos_teacher
prefix)
    df['teacher_prefix_neg'] = table['teacher_prefix'].map(neg_teacher
prefix)
    df['proj_grade_cat_pos'] = table['project_grade_category'].map(pos
_projgradecat)
    df['proj_grade_cat_neg'] = table['project_grade_category'].map(neg
_projgradecat)

    return df
```

In [156]:
```
newTrain = Responsecode(project_data_train)
newTest = Responsecode(project_data_test)
```

In [157]:
```
def mergeEncoding(table, p, n) :
    lstPos = table[p].values.tolist()
    lstNeg = table[n].values.tolist()
    frame = pd.DataFrame(list(zip(lstNeg, lstPos)))

    return frame
```

In [158]:
```
#Clean Categories
X_train_clean_cat_ohe = mergeEncoding(newTrain, 'clean_cat_pos', 'clea
n_cat_neg')
X_test_clean_cat_ohe = mergeEncoding(newTest, 'clean_cat_pos', 'clean_
cat_neg')
print(X_train_clean_cat_ohe.shape)
print(X_test_clean_cat_ohe.shape)
```

```
(20100, 2)
(9900, 2)
```

In [159]:
```
#Clean SUB Categories
X_train_clean_subcat_ohe = mergeEncoding(newTrain, 'clean_subcat_pos',
'clean_subcat_neg')
X_test_clean_subcat_ohe = mergeEncoding(newTest, 'clean_subcat_pos', '
clean_subcat_neg')
print(X_train_clean_subcat_ohe.shape)
print(X_test_clean_subcat_ohe.shape)
```

```
(20100, 2)
(9900, 2)
```

In [160]:
```
#Project Grade Category
X_train_grade_ohe = mergeEncoding(newTrain, 'proj_grade_cat_pos', 'pro
j_grade_cat_neg')
X_test_grade_ohe = mergeEncoding(newTest, 'proj_grade_cat_pos', 'proj_
grade_cat_neg')
print(X_train_grade_ohe.shape)
print(X_test_grade_ohe.shape)
```

```
(20100, 2)
(9900, 2)
```

In [161]:
```
#School State
X_train_state_ohe = mergeEncoding(newTrain, 'school_state_pos', 'schoo
l_state_neg')
X_test_state_ohe = mergeEncoding(newTest, 'school_state_pos', 'school_
state_neg')
print(X_train_state_ohe.shape)
print(X_test_state_ohe.shape)
```

```
(20100, 2)
(9900, 2)
```

In [162]:
```
#Teacher Prefix
X_train_teacher_ohe = mergeEncoding(newTrain, 'teacher_prefix_pos', 't
eacher_prefix_neg')
X_test_teacher_ohe = mergeEncoding(newTest, 'teacher_prefix_pos', 'tea
cher_prefix_neg')
print(X_train_teacher_ohe.shape)
print(X_test_teacher_ohe.shape)
```

```
(20100, 2)
(9900, 2)
```

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

## 1.5.2 Vectorizing Text data

### 1.5.2.1 Bag of words

In [163]:
```
def VectorizingTextData(sFeature, project_data_fitting, project_data_transform):
    from sklearn.feature_extraction.text import CountVectorizer
    vectorizer_feature = CountVectorizer(lowercase=False, binary=True, min_df = 10, ngram_range=(1, 2),max_features = 5000)
    vectorizer_feature.fit(project_data_fitting[sFeature].values) #fitting has to be on Train data
    transform_one_hot = vectorizer_feature.transform(project_data_transform[sFeature].values)
    #print(vectorizer_cat.get_feature_names())
    return(transform_one_hot)

def fnGetTextFeatures(sFeature, project_data_fitting, project_data_transform):
    from sklearn.feature_extraction.text import CountVectorizer
    vectorizer_feature = CountVectorizer(lowercase=False, binary=True, min_df = 10, ngram_range=(1, 2),max_features = 5000)
    vectorizer_feature.fit(project_data_fitting[sFeature].values) #fitting has to be on Train data
    return(vectorizer_feature.get_feature_names())
```

In [164]:
```
train_essay_bow = VectorizingTextData('preprocessed_essays', project_d
ata_train, project_data_train)
test_essay_bow = VectorizingTextData('preprocessed_essays', project_da
ta_train, project_data_test)

print("Shape of train data matrix after one hot encoding ",train_essay
_bow.shape)
print("Shape of test data matrix after one hot encoding ",test_essay_b
ow.shape)

essay_features = fnGetTextFeatures('preprocessed_essays', project_data
_train, project_data_train)
print(essay_features)
```

```
Shape of train data matrix after one hot encoding  (20100, 5000)
Shape of test data matrix after one hot encoding  (9900, 5000)
['000', '10', '100', '100 free', '100 percent', '100 students', '11',
'12', '12th', '13', '14', '15', '16', '17', '18', '19', '1st', '1st g
rade', '20', '20 students', '200', '2016', '2017', '21', '21st', '21s
t century', '22', '23', '24', '25', '25 students', '26', '27', '28',
'2nd', '2nd grade', '2nd graders', '30', '30 students', '32', '35', '
3d', '3d printer', '3rd', '3rd grade', '3rd graders', '40', '400', '4
5', '4th', '4th 5th', '4th grade', '4th graders', '50', '50 students
', '500', '500 students', '5th', '5th grade', '5th graders', '60', '6
0 minutes', '60 students', '600', '600 students', '6th', '6th grade',
'6th graders', '70', '70 students', '75', '75 students', '7th', '7th
8th', '7th grade', '80', '80 students', '85', '8th', '8th grade', '8t
h graders', '90', '90 students', '95', '98', '99', '9th', 'abilities
', 'abilities students', 'ability', 'ability focus', 'ability learn',
'ability levels', 'able', 'able access', 'able choose', 'able complet
e', 'able control', 'able create', 'able experience', 'able explore',
'able focus', 'able get', 'able give', 'able help', 'able keep', 'abl
e learn', 'able listen', 'able make', 'able move', 'able play', 'able
practice', 'able print', 'able provide', 'able read', 'able see', 'ab
le share', 'able sit', 'able take', 'able teach', 'able use', 'able u
tilize', 'able work', 'absolutely', 'absolutely love', 'absorb', 'abs
tract', 'academic', 'academic achievement', 'academic areas', 'academ
ic excellence', 'academic needs', 'academic performance', 'academic s
kills', 'academic social', 'academic success', 'academically', 'acade
mically socially', 'academics', 'academy', 'accelerated', 'accept', '
acceptance', 'accepted', 'access', 'access books', 'access computers
', 'access home', 'access internet', 'access many', 'access materials
', 'access online', 'access resources', 'access technology', 'accessi
ble', 'accessible students', 'accessories', 'accommodate', 'accomplis
h', 'accomplished', 'accomplishments', 'according', 'achieve', 'achie
ve goals', 'achieve success', 'achievement', 'achievement gap', 'achi
eving', 'acquire', 'acquiring', 'acquisition', 'across', 'act', 'acti
on', 'actions', 'active', 'active learners', 'active learning', 'acti
ve students', 'actively', 'actively engaged', 'activities', 'activiti
es help', 'activities students', 'activity', 'actual', 'actually', 'a
dapt', 'add', 'added', 'adding', 'addition', 'addition classroom', 'a
ddition students', 'addition subtraction', 'additional', 'additionall
y', 'address', 'adequate', 'adhd', 'adjust', 'administration', 'adult
', 'adults', 'advance', 'advanced', 'advantage', 'adventure', 'advent
ures', 'adversity', 'affect', 'affects', 'affluent', 'afford', 'afrai
d', 'african', 'african american', 'afternoon', 'age', 'age appropria
te', 'ages', 'ago', 'ahead', 'ahead early', 'aid', 'aids', 'aim', 'ai
r', 'alive', 'allow', 'allow children', 'allow student', 'allow stude
nts', 'allow us', 'allowed', 'allowing', 'allowing students', 'allows
', 'allows students', 'almost', 'alone', 'along', 'aloud', 'alouds',
'alphabet', 'already', 'also', 'also able', 'also allow', 'also give
', 'also help', 'also helps', 'also learn', 'also learning', 'also li
ke', 'also love', 'also need', 'also provide', 'also requesting', 'al
so students', 'also teach', 'also use', 'also used', 'also want', 'al
ternative', 'alternative seating', 'although', 'although students', '
always', 'always asking', 'always eager', 'always excited', 'always l
ooking', 'always ready', 'amaze', 'amazed', 'amazing', 'amazing group
', 'amazing students', 'amazing things', 'ambitious', 'america', 'ame
rican', 'americans', 'among', 'among students', 'amount', 'amount tim
e', 'ample', 'analysis', 'analyze', 'anchor', 'angeles', 'animal', 'a
```

```
nimals', 'another', 'answer', 'answer questions', 'answers', 'anxiety
', 'anxious', 'anyone', 'anything', 'anywhere', 'ap', 'apart', 'app',
'apple', 'application', 'applications', 'apply', 'applying', 'appreci
ate', 'appreciated', 'appreciation', 'appreciative', 'approach', 'app
ropriate', 'appropriately', 'approximately', 'apps', 'area', 'area ma
ny', 'area students', 'areas', 'areas students', 'around', 'around cl
assroom', 'around room', 'around school', 'around students', 'around
us', 'around world', 'arrangement', 'array', 'arrive', 'art', 'art cl
ass', 'art projects', 'art room', 'art students', 'art supplies', 'ar
ticles', 'artist', 'artistic', 'artists', 'arts', 'arts math', 'arts
students', 'artwork', 'asian', 'ask', 'ask questions', 'asked', 'aski
ng', 'asking help', 'aspect', 'aspects', 'assess', 'assessment', 'ass
essments', 'asset', 'assigned', 'assignment', 'assignments', 'assist
', 'assist students', 'assistance', 'athletes', 'athletic', 'atmosphe
re', 'attend', 'attend school', 'attend title', 'attendance', 'attend
ed', 'attending', 'attention', 'attention many', 'attitude', 'attitud
es', 'audio', 'auditory', 'august', 'authentic', 'author', 'authors',
'autism', 'autism spectrum', 'autistic', 'available', 'available clas
sroom', 'available students', 'average', 'avid', 'award', 'aware', 'a
wareness', 'away', 'away home', 'awesome', 'back', 'background', 'bac
kground knowledge', 'backgrounds', 'backgrounds cultures', 'backgroun
ds many', 'backgrounds school', 'backgrounds students', 'backpack', '
backpack food', 'backpacks', 'bad', 'bag', 'bag chairs', 'bags', 'bal
ance', 'balance balls', 'balanced', 'ball', 'ball chairs', 'balls', '
band', 'bands', 'barrier', 'barriers', 'base', 'based', 'based learni
ng', 'based socioeconomic', 'basic', 'basic needs', 'basic school', '
basic skills', 'basic supplies', 'basics', 'basis', 'basis students',
'basketball', 'bean', 'bean bag', 'bean bags', 'beautiful', 'became',
'become', 'become better', 'become engaged', 'become independent', 'b
ecome life', 'become lifelong', 'become successful', 'becomes', 'beco
ming', 'began', 'begin', 'beginning', 'beginning school', 'beginning
year', 'begins', 'behavior', 'behavioral', 'behaviors', 'behind', 'be
lief', 'believe', 'believe students', 'beneficial', 'beneficial stude
nts', 'benefit', 'benefit greatly', 'benefit students', 'benefits', '
besides', 'best', 'best education', 'best every', 'best learning', 'b
est part', 'best possible', 'best school', 'best students', 'best wan
t', 'best way', 'best work', 'better', 'better able', 'better place',
'better readers', 'better students', 'better understand', 'better und
erstanding', 'better way', 'beyond', 'big', 'bigger', 'biggest', 'bil
ingual', 'binders', 'bins', 'bit', 'black', 'blended', 'blessed', 'bl
ock', 'blocks', 'blood', 'blue', 'board', 'boards', 'bodies', 'body',
'book', 'book read', 'book students', 'books', 'books allow', 'books
also', 'books classroom', 'books help', 'books home', 'books nannan',
'books not', 'books read', 'books reading', 'books students', 'books
used', 'books would', 'boost', 'boring', 'born', 'borrow', 'bottom',
'bought', 'bounce', 'bouncy', 'bouncy bands', 'bound', 'box', 'boxes
', 'boys', 'boys girls', 'brain', 'brain breaks', 'brains', 'brand',
'brand new', 'break', 'breakfast', 'breakfast lunch', 'breaks', 'brid
ge', 'bright', 'brilliant', 'bring', 'bringing', 'brings', 'broaden',
'broken', 'bronx', 'brooklyn', 'brought', 'budget', 'budget cuts', 'b
udgets', 'build', 'build confidence', 'building', 'builds', 'built',
'bunch', 'burn', 'bus', 'business', 'busy', 'buy', 'buying', 'calcula
tors', 'california', 'call', 'called', 'calm', 'calming', 'calories',
'came', 'camera', 'cameras', 'campus', 'cannot', 'cannot afford', 'ca
nnot wait', 'capabilities', 'capable', 'capacity', 'capture', 'card',
'cards', 'care', 'cared', 'career', 'career ready', 'careers', 'carin
```

```
g', 'carolina', 'carpet', 'carry', 'cart', 'case', 'cases', 'catch',
'caucasian', 'cause', 'causes', 'cd', 'celebrate', 'center', 'center
students', 'center time', 'centered', 'centers', 'centers students',
'central', 'century', 'century learners', 'century learning', 'centur
y skills', 'certain', 'certainly', 'certainly control', 'chair', 'cha
irs', 'chairs allow', 'chairs help', 'chairs students', 'challenge',
'challenge students', 'challenged', 'challenges', 'challenges classro
om', 'challenges face', 'challenges students', 'challenging', 'chance
', 'chances', 'change', 'change students', 'change world', 'changed',
'changes', 'changing', 'changing world', 'chapter', 'chapter books',
'character', 'characters', 'charge', 'charge learning', 'charging', '
chart', 'charter', 'charter school', 'charts', 'check', 'checking', '
chemistry', 'chicago', 'child', 'childhood', 'children', 'children ab
le', 'children come', 'children learn', 'children love', 'children ne
ed', 'children not', 'children school', 'children students', 'choice
', 'choices', 'choose', 'choosing', 'chose', 'chosen', 'chrome', 'chr
ome books', 'chromebook', 'chromebooks', 'chromebooks allow', 'chrome
books classroom', 'chromebooks students', 'chromebooks would', 'circl
e', 'circles', 'circumstances', 'citizens', 'city', 'city school', 'c
ity students', 'class', 'class able', 'class also', 'class consists',
'class full', 'class made', 'class nannan', 'class need', 'class not
', 'class school', 'class set', 'class sizes', 'class students', 'cla
ss time', 'class use', 'class work', 'class would', 'class year', 'cl
asses', 'classes students', 'classic', 'classified', 'classmates', 'c
lassroom', 'classroom able', 'classroom activities', 'classroom allow
', 'classroom also', 'classroom always', 'classroom classroom', 'clas
sroom come', 'classroom community', 'classroom consists', 'classroom
currently', 'classroom day', 'classroom despite', 'classroom diverse
', 'classroom environment', 'classroom every', 'classroom family', 'c
lassroom feel', 'classroom filled', 'classroom first', 'classroom foc
us', 'classroom full', 'classroom give', 'classroom help', 'classroom
home', 'classroom learn', 'classroom learning', 'classroom library',
'classroom love', 'classroom made', 'classroom make', 'classroom many
', 'classroom materials', 'classroom nannan', 'classroom need', 'clas
sroom needs', 'classroom not', 'classroom one', 'classroom place', 'c
lassroom project', 'classroom provide', 'classroom safe', 'classroom
school', 'classroom set', 'classroom setting', 'classroom student', '
classroom students', 'classroom supplies', 'classroom technology', 'c
lassroom use', 'classroom want', 'classroom well', 'classroom work',
'classroom would', 'classroom year', 'classrooms', 'classwork', 'clay
', 'clean', 'cleaning', 'clear', 'clearly', 'clever', 'climate', 'cli
pboards', 'close', 'close achievement', 'close knit', 'closer', 'clot
hes', 'clothing', 'club', 'clubs', 'co', 'code', 'coding', 'cognitive
', 'cold', 'collaborate', 'collaborating', 'collaboration', 'collabor
ative', 'collaborative learning', 'collaboratively', 'collect', 'coll
ection', 'college', 'college career', 'college students', 'color', 'c
olor printer', 'colored', 'colored pencils', 'colorful', 'colors', 'c
om', 'combination', 'combine', 'combined', 'come', 'come alive', 'com
e class', 'come classroom', 'come different', 'come diverse', 'come e
very', 'come families', 'come high', 'come homes', 'come life', 'come
low', 'come many', 'come nannan', 'come school', 'come single', 'come
together', 'come true', 'come variety', 'come various', 'come wide',
'comes', 'comfort', 'comfortable', 'comfortable learning', 'comfortab
le place', 'comfortably', 'comfy', 'coming', 'coming school', 'commit
ment', 'committed', 'common', 'common core', 'communicate', 'communic
ating', 'communication', 'communication skills', 'communities', 'comm
```

```
unity', 'community learners', 'community many', 'community school', '
community students', 'compare', 'compassion', 'compassionate', 'compe
te', 'competition', 'competitions', 'competitive', 'complete', 'compl
ete assignments', 'complete work', 'completed', 'completely', 'comple
ting', 'complex', 'component', 'components', 'composed', 'composition
', 'comprehend', 'comprehension', 'comprehension skills', 'comprised
', 'computer', 'computer lab', 'computer programming', 'computer scie
nce', 'computers', 'computers classroom', 'computers home', 'concentr
ate', 'concentration', 'concept', 'concepts', 'concepts students', 'c
oncrete', 'conditions', 'conducive', 'conduct', 'conduct research', '
confidence', 'confident', 'connect', 'connected', 'connecting', 'conn
ection', 'connections', 'consider', 'consider helping', 'consideratio
n', 'considered', 'considering', 'consist', 'consistent', 'consistent
ly', 'consists', 'constant', 'constantly', 'construct', 'construction
', 'contagious', 'contain', 'contained', 'content', 'content areas',
'continually', 'continue', 'continue grow', 'continue path', 'continu
ed', 'continues', 'continuing', 'continuously', 'contribute', 'contri
buting', 'contribution', 'control', 'control experience', 'control ho
me', 'control learning', 'conversation', 'conversations', 'cooking',
'cool', 'cooperation', 'cooperative', 'cooperative learning', 'cooper
atively', 'coordination', 'copies', 'copy', 'core', 'core muscles', '
core standards', 'core strength', 'corner', 'correct', 'correctly', '
cost', 'could', 'could not', 'could use', 'count', 'counting', 'count
less', 'countries', 'country', 'county', 'couple', 'course', 'courses
', 'cover', 'covered', 'covers', 'cozy', 'craft', 'crave', 'crayons',
'create', 'create art', 'create classroom', 'create environment', 'cr
eate learning', 'create new', 'create positive', 'create projects', '
created', 'creates', 'creating', 'creation', 'creations', 'creative',
'creative clever', 'creative meaningful', 'creative positive', 'creat
ive thinking', 'creative ways', 'creatively', 'creativity', 'crime',
'critical', 'critical thinkers', 'critical thinking', 'critically', '
cross', 'crucial', 'cultural', 'cultural backgrounds', 'culturally',
'culturally diverse', 'culture', 'cultures', 'curiosity', 'curious',
'curious world', 'current', 'current events', 'currently', 'currently
not', 'currently students', 'curricular', 'curriculum', 'curriculum s
tudents', 'cushions', 'cut', 'cuts', 'cutting', 'cycle', 'daily', 'da
ily basis', 'daily classroom', 'daily lives', 'daily students', 'danc
e', 'dancing', 'dash', 'data', 'date', 'day', 'day class', 'day class
room', 'day come', 'day creative', 'day day', 'day eager', 'day excit
ed', 'day full', 'day learn', 'day learning', 'day long', 'day love',
'day many', 'day nannan', 'day not', 'day one', 'day ready', 'day sch
ool', 'day students', 'day want', 'day work', 'days', 'deal', 'dealin
g', 'decide', 'decided', 'decisions', 'decrease', 'dedicated', 'dedic
ation', 'deep', 'deepen', 'deeper', 'deeper understanding', 'deeply',
'deficit', 'define', 'definitely', 'delays', 'deliver', 'demand', 'de
mands', 'demonstrate', 'department', 'depend', 'depth', 'describe', '
deserve', 'deserve best', 'deserves', 'deserving', 'design', 'designa
ted', 'designed', 'designing', 'designs', 'desire', 'desire learn', '
desk', 'desks', 'desks chairs', 'desktop', 'desperate', 'desperate ne
ed', 'desperately', 'desperately need', 'despite', 'despite challenge
s', 'despite hardships', 'despite many', 'details', 'determination',
'determine', 'determined', 'develop', 'develop love', 'develop skills
', 'developed', 'developing', 'development', 'developmental', 'develo
pmentally', 'device', 'devices', 'diagnosed', 'difference', 'differen
ce classroom', 'difference lives', 'difference students', 'difference
s', 'different', 'different backgrounds', 'different countries', 'dif
```

```
ferent cultures', 'different economic', 'different languages', 'diffe
rent learning', 'different levels', 'different needs', 'different sea
ting', 'different types', 'different way', 'different ways', 'differe
ntiate', 'differentiate instruction', 'differentiated', 'differentiat
ed instruction', 'differentiation', 'differently', 'difficult', 'diff
icult students', 'difficult time', 'difficulties', 'difficulty', 'dig
', 'digital', 'dinner', 'direct', 'directed', 'direction', 'direction
s', 'directly', 'dirty', 'disabilities', 'disabilities students', 'di
sability', 'disabled', 'disadvantage', 'disadvantaged', 'discipline',
'discover', 'discover new', 'discovered', 'discoveries', 'discovering
', 'discovery', 'discuss', 'discussing', 'discussion', 'discussions',
'disorder', 'disorders', 'display', 'displayed', 'distracted', 'distr
acting', 'distraction', 'distractions', 'district', 'district student
s', 'districts', 'dive', 'diverse', 'diverse backgrounds', 'diverse c
ommunity', 'diverse group', 'diverse learners', 'diverse learning', '
diverse population', 'diverse school', 'diverse student', 'diverse st
udents', 'diversity', 'docs', 'doctors', 'document', 'document camera
', 'documents', 'donate', 'donated', 'donating', 'donating project',
'donation', 'donation help', 'donation project', 'donations', 'donati
ons help', 'donations project', 'done', 'donors', 'donors choose', 'd
onorschoose', 'door', 'door classroom', 'doors', 'dot', 'dr', 'dr seu
ss', 'drama', 'dramatic', 'dramatic play', 'draw', 'drawing', 'drawin
gs', 'dream', 'dreamers', 'dreams', 'drive', 'driven', 'drop', 'dry',
'dry erase', 'dual', 'dual language', 'due', 'due lack', 'durable', '
duty', 'dynamic', 'eager', 'eager excited', 'eager learn', 'eager lea
rners', 'eagerly', 'eagerness', 'eagerness learn', 'earliest', 'earli
est learners', 'early', 'early age', 'early childhood', 'early life',
'earn', 'earth', 'ease', 'easel', 'easier', 'easily', 'east', 'easy',
'easy access', 'eat', 'eating', 'economic', 'economic backgrounds', '
economic status', 'economically', 'economically disadvantaged', 'ed',
'edge', 'edit', 'editing', 'educate', 'educated', 'education', 'educa
tion class', 'education classroom', 'education nannan', 'education no
t', 'education possible', 'education services', 'education students',
'education teacher', 'educational', 'educational apps', 'educational
experience', 'educational experiences', 'educational games', 'educato
r', 'educators', 'effect', 'effective', 'effectively', 'effects', 'ef
ficient', 'efficiently', 'effort', 'efforts', 'eight', 'eighth', 'eig
hth grade', 'either', 'ela', 'electronic', 'elementary', 'elementary
school', 'elementary schools', 'elementary students', 'elements', 'el
igible', 'eligible free', 'eliminate', 'ell', 'ell students', 'else',
'embrace', 'emotional', 'emotional needs', 'emotionally', 'emotions',
'empathy', 'emphasis', 'empower', 'empower students', 'empowered', 'e
mpowering', 'enable', 'enable students', 'enables', 'encounter', 'enc
ourage', 'encourage students', 'encouraged', 'encouragement', 'encour
ages', 'encouraging', 'end', 'end day', 'end school', 'end year', 'en
deavors', 'endless', 'ends', 'ends meet', 'energetic', 'energy', 'eng
age', 'engage learning', 'engage students', 'engaged', 'engaged learn
ing', 'engagement', 'engages', 'engaging', 'engaging activities', 'en
gaging learning', 'engaging students', 'engaging way', 'engineer', 'e
ngineering', 'engineering math', 'engineers', 'english', 'english lan
guage', 'english learners', 'english not', 'english second', 'english
spanish', 'english speakers', 'english students', 'enhance', 'enhance
learning', 'enhance students', 'enhanced', 'enhances', 'enhancing', '
enjoy', 'enjoy coming', 'enjoy learning', 'enjoy reading', 'enjoy usi
ng', 'enjoy working', 'enjoyable', 'enjoyed', 'enjoying', 'enjoyment
', 'enough', 'enrich', 'enriching', 'enrichment', 'enrolled', 'enroll
```

```
ment', 'ensure', 'ensure students', 'enter', 'enter classroom', 'ente
ring', 'enthusiasm', 'enthusiasm learning', 'enthusiastic', 'enthusia
stic learners', 'enthusiastic learning', 'entire', 'entire class', 'e
ntire school', 'environment', 'environment classroom', 'environment n
annan', 'environment students', 'environmental', 'environments', 'env
ision', 'equal', 'equipment', 'equipment students', 'equipped', 'eras
e', 'erase boards', 'erase markers', 'erasers', 'escape', 'esl', 'esl
students', 'esol', 'especially', 'essays', 'essential', 'essentials',
'esteem', 'etc', 'ethnic', 'ethnic backgrounds', 'ethnically', 'ethni
cally diverse', 'ethnicities', 'ethnicity', 'even', 'even earliest',
'even not', 'even though', 'event', 'events', 'eventually', 'ever', '
ever changing', 'every', 'every child', 'every day', 'every morning',
'every one', 'every opportunity', 'every single', 'every student', 'e
very time', 'every week', 'every year', 'everyday', 'everyday student
s', 'everyone', 'everything', 'everything need', 'everywhere', 'evide
nce', 'exactly', 'example', 'examples', 'exceed', 'excel', 'excellenc
e', 'excellent', 'exceptional', 'excess', 'excess energy', 'excite',
'excited', 'excited come', 'excited learn', 'excited learning', 'exci
ted reading', 'excited ready', 'excited school', 'excited see', 'exci
ted store', 'excitement', 'excitement learning', 'exciting', 'exercis
e', 'exercise balls', 'exercises', 'exercising', 'expand', 'expanding
', 'expect', 'expectations', 'expected', 'expensive', 'experience', '
experience school', 'experience students', 'experienced', 'experience
s', 'experiences many', 'experiences school', 'experiences students',
'experiencing', 'experiment', 'experiments', 'explain', 'exploration
', 'explore', 'explore learn', 'explore new', 'explore world', 'explo
ring', 'expose', 'expose students', 'exposed', 'exposing', 'exposure
', 'express', 'expressed', 'expression', 'extend', 'extended', 'extra
', 'extra energy', 'extra help', 'extra support', 'extreme', 'extreme
ly', 'extremely hard', 'eye', 'eyes', 'fabulous', 'face', 'face chall
enges', 'face daily', 'face looking', 'face many', 'face students', '
faced', 'faced challenges', 'faced many', 'faced several', 'faces', '
facilitate', 'facing', 'fact', 'factors', 'facts', 'faculty', 'fail',
'failure', 'fair', 'fall', 'fall love', 'falling', 'familiar', 'famil
ies', 'families many', 'families not', 'families school', 'families s
truggle', 'families students', 'family', 'family members', 'family st
udents', 'fantastic', 'far', 'farm', 'farming', 'fast', 'faster', 'fa
vorite', 'fear', 'features', 'feed', 'feedback', 'feel', 'feel comfor
table', 'feel confident', 'feel like', 'feel safe', 'feel successful
', 'feeling', 'feelings', 'feels', 'feet', 'felt', 'fiction', 'fidget
', 'fidgeting', 'field', 'field trips', 'fields', 'fifth', 'fifth gra
de', 'fifth graders', 'fight', 'figure', 'fill', 'filled', 'film', 'f
inal', 'finally', 'financial', 'financially', 'find', 'find ways', 'f
inding', 'findings', 'fine', 'fine motor', 'fingertips', 'finish', 'f
inished', 'fire', 'first', 'first day', 'first experience', 'first ge
neration', 'first grade', 'first graders', 'first hand', 'first langu
age', 'first second', 'first time', 'first year', 'fit', 'fitness', '
fits', 'five', 'five six', 'five year', 'flexibility', 'flexible', 'f
lexible seating', 'floor', 'florida', 'flourish', 'flow', 'fluency',
'fluency comprehension', 'fluent', 'focus', 'focus learning', 'focus
potential', 'focus students', 'focus work', 'focused', 'focused learn
ing', 'focuses', 'focusing', 'folder', 'folders', 'follow', 'follow a
long', 'following', 'food', 'food weekend', 'foods', 'foot', 'footbal
l', 'force', 'forced', 'forever', 'forget', 'form', 'formal', 'format
', 'forms', 'forth', 'fortunate', 'fortunate enough', 'forward', 'for
ward coming', 'foster', 'foster love', 'fostering', 'fosters', 'found
```

', 'foundation', 'foundational', 'four', 'fourth', 'fourth grade', 'f
ourth graders', 'free', 'free breakfast', 'free lunch', 'free reduced
', 'freedom', 'freely', 'frequent', 'frequently', 'fresh', 'friday',
'friend', 'friendly', 'friends', 'friendships', 'front', 'frustrated
', 'frustration', 'fuel', 'fulfill', 'full', 'full energy', 'full lif
e', 'full potential', 'fullest', 'fullest potential', 'fully', 'fun',
'fun engaging', 'fun exciting', 'fun interactive', 'fun learning', 'f
un loving', 'fun nannan', 'fun students', 'fun way', 'function', 'fun
ctional', 'functioning', 'functions', 'fund', 'fundamental', 'funded
', 'funding', 'funding project', 'funds', 'funny', 'furniture', 'furt
hermore', 'future', 'future leaders', 'future nannan', 'future studen
ts', 'future want', 'futures', 'gain', 'gain confidence', 'gaining',
'gains', 'game', 'games', 'games activities', 'games help', 'games st
udents', 'gap', 'gaps', 'garden', 'gather', 'general', 'general educa
tion', 'generally', 'generation', 'generations', 'generosity', 'gener
ous', 'generous donation', 'genre', 'genres', 'geography', 'geometry
', 'get', 'get chance', 'get excited', 'get experience', 'get hands',
'get move', 'get moving', 'get new', 'get opportunity', 'get school',
'get students', 'get use', 'get wiggles', 'get work', 'gets', 'gettin
g', 'getting ahead', 'gift', 'gifted', 'gifted students', 'gifted tal
ented', 'girls', 'give', 'give best', 'give chance', 'give every', 'g
ive opportunity', 'give student', 'give students', 'give us', 'given
', 'given opportunity', 'gives', 'gives students', 'giving', 'giving
students', 'global', 'glue', 'glue sticks', 'go', 'go beyond', 'go co
llege', 'go home', 'go school', 'goal', 'goal create', 'goal help', '
goal make', 'goal provide', 'goal students', 'goals', 'goals students
', 'goes', 'going', 'gone', 'good', 'good book', 'google', 'google cl
assroom', 'google docs', 'got', 'government', 'grab', 'grade', 'grade
class', 'grade classroom', 'grade level', 'grade levels', 'grade math
', 'grade reading', 'grade school', 'grade students', 'grade teacher
', 'grade year', 'graders', 'graders come', 'graders eager', 'graders
love', 'graders students', 'grades', 'graduate', 'graduation', 'gramm
ar', 'grandparents', 'grant', 'granted', 'graphic', 'graphic novels',
'grasp', 'grateful', 'great', 'great addition', 'great deal', 'great
group', 'great kids', 'great start', 'great students', 'great things
', 'great way', 'greater', 'greatest', 'greatly', 'greatly appreciate
d', 'greatly benefit', 'greatness', 'green', 'greet', 'gross', 'gross
motor', 'ground', 'group', 'group activities', 'group children', 'gro
up instruction', 'group kids', 'group learners', 'group lessons', 'gr
oup students', 'group time', 'group work', 'groups', 'groups students
', 'grow', 'grow learn', 'grow students', 'growing', 'grown', 'growth
', 'growth may', 'growth mindset', 'growth students', 'guidance', 'gu
ide', 'guided', 'guided reading', 'gym', 'habits', 'half', 'half stud
ents', 'hand', 'handle', 'hands', 'hands activities', 'hands experien
ce', 'hands experiences', 'hands learning', 'hands materials', 'hands
projects', 'hands students', 'happen', 'happening', 'happens', 'happy
', 'hard', 'hard every', 'hard get', 'hard make', 'hard students', 'h
ard time', 'hard work', 'hard workers', 'hard working', 'harder', 'ha
rdest', 'hardships', 'hardships students', 'hardworking', 'head', 'he
adphones', 'headphones students', 'health', 'healthier', 'healthy', '
healthy lifestyle', 'healthy snacks', 'hear', 'heard', 'hearing', 'he
art', 'hearts', 'heavy', 'held', 'hello', 'help', 'help achieve', 'he
lp become', 'help bring', 'help build', 'help children', 'help classr
oom', 'help create', 'help develop', 'help ensure', 'help focus', 'he
lp foster', 'help get', 'help give', 'help grow', 'help help', 'help
improve', 'help increase', 'help keep', 'help kids', 'help learn', 'h

```
elp learning', 'help make', 'help meet', 'help nannan', 'help prepare
', 'help provide', 'help reach', 'help stay', 'help student', 'help s
tudents', 'help succeed', 'help successful', 'help support', 'help te
ach', 'help understand', 'help us', 'helped', 'helpful', 'helping', '
helping students', 'helping us', 'helps', 'helps students', 'high', '
high achieving', 'high energy', 'high expectations', 'high interest',
'high level', 'high needs', 'high percentage', 'high poverty', 'high
quality', 'high school', 'higher', 'higher level', 'highest', 'highes
t potential', 'highlight', 'highly', 'highly motivated', 'hispanic',
'historical', 'history', 'hit', 'hokki', 'hokki stools', 'hold', 'hol
ding', 'home', 'home life', 'home lives', 'home many', 'home nannan',
'home not', 'home school', 'home students', 'home want', 'homeless',
'homes', 'homework', 'honor', 'honored', 'hope', 'hope students', 'ho
peful', 'hopeful inspire', 'hopefully', 'hopes', 'hoping', 'hot', 'ho
ur', 'hours', 'hours day', 'house', 'household', 'households', 'house
holds receive', 'houses', 'housing', 'however', 'however certainly',
'however many', 'however not', 'however students', 'huge', 'huge diff
erence', 'human', 'hundred', 'hundreds', 'hunger', 'hungry', 'idea',
'ideal', 'ideas', 'identified', 'identify', 'iep', 'ignite', 'illustr
ations', 'images', 'imagination', 'imaginations', 'imaginative', 'ima
gine', 'immediate', 'immediately', 'immersion', 'immigrant', 'immigra
nts', 'impact', 'impact students', 'impacts', 'impairments', 'imperat
ive', 'implement', 'implemented', 'implementing', 'importance', 'impo
rtant', 'important part', 'important skills', 'important students', '
importantly', 'impossible', 'impoverished', 'improve', 'improve class
room', 'improve learning', 'improve reading', 'improve students', 'im
proved', 'improvement', 'improves', 'improving', 'incentive', 'includ
e', 'included', 'includes', 'including', 'inclusion', 'inclusive', 'i
ncome', 'income area', 'income community', 'income families', 'income
high', 'income homes', 'income households', 'income neighborhood', 'i
ncome school', 'income students', 'incorporate', 'incorporate technol
ogy', 'incorporated', 'incorporating', 'increase', 'increase academic
', 'increase reading', 'increase student', 'increase students', 'incr
eased', 'increases', 'increasing', 'increasingly', 'incredible', 'inc
redibly', 'independence', 'independent', 'independent learners', 'ind
ependent reading', 'independent work', 'independently', 'individual',
'individual learning', 'individual needs', 'individual student', 'ind
ividualized', 'individually', 'individuals', 'indoor', 'industry', 'i
nformation', 'informational', 'initiative', 'ink', 'inner', 'inner ci
ty', 'innovation', 'innovative', 'innovators', 'input', 'inquiry', 'i
nquiry based', 'inquisitive', 'inside', 'inside classroom', 'inside o
utside', 'inspiration', 'inspire', 'inspire even', 'inspire students
', 'inspired', 'inspired project', 'inspires', 'inspiring', 'instant
', 'instead', 'instill', 'instill love', 'instruction', 'instruction
students', 'instructional', 'instructions', 'instrument', 'instrument
s', 'integral', 'integrate', 'integrated', 'integrating', 'integratio
n', 'intellectual', 'intellectual disabilities', 'intelligent', 'inte
ract', 'interacting', 'interaction', 'interactions', 'interactive', '
interactive learning', 'interactive notebooks', 'interest', 'interest
reading', 'interest students', 'interested', 'interesting', 'interest
s', 'international', 'internet', 'internet access', 'intervention', '
interventions', 'introduce', 'introduce students', 'introduced', 'int
roducing', 'invaluable', 'invested', 'investigate', 'investigations',
'investment', 'inviting', 'involve', 'involved', 'involvement', 'invo
lves', 'ipad', 'ipad mini', 'ipad minis', 'ipads', 'ipads classroom',
'ipads students', 'ipads would', 'issue', 'issues', 'item', 'items',
```

'items help', 'items students', 'job', 'jobs', 'join', 'journal', 'jo
urnals', 'journey', 'joy', 'jump', 'junior', 'keep', 'keep engaged',
'keep students', 'keep things', 'keeping', 'keeps', 'key', 'keyboard
', 'kid', 'kid inspired', 'kiddos', 'kids', 'kids come', 'kids learn
', 'kids love', 'kids need', 'kids not', 'kind', 'kindergarten', 'kin
dergarten class', 'kindergarten classroom', 'kindergarten fifth', 'ki
ndergarten first', 'kindergarten students', 'kindergarten teacher', '
kindergarteners', 'kindergartners', 'kindle', 'kindles', 'kindness',
'kinds', 'kinesthetic', 'kit', 'kits', 'knew', 'knit', 'know', 'know
learn', 'know students', 'knowing', 'knowledge', 'knowledge students
', 'known', 'knows', 'lab', 'labeled', 'labs', 'lack', 'lack resource
s', 'lacking', 'lacks', 'language', 'language arts', 'language englis
h', 'language learners', 'language skills', 'language students', 'lan
guages', 'languages spoken', 'lap', 'laptop', 'laptops', 'large', 'la
rge population', 'large urban', 'larger', 'largest', 'last', 'last ye
ar', 'last years', 'lasting', 'lastly', 'late', 'later', 'latest', 'l
atino', 'laugh', 'laughter', 'lay', 'lead', 'leader', 'leaders', 'lea
dership', 'leading', 'leads', 'learn', 'learn best', 'learn better',
'learn classroom', 'learn come', 'learn create', 'learn day', 'learn
different', 'learn differently', 'learn english', 'learn every', 'lea
rn excited', 'learn explore', 'learn fun', 'learn grow', 'learn hands
', 'learn important', 'learn love', 'learn make', 'learn many', 'lear
n math', 'learn much', 'learn nannan', 'learn need', 'learn new', 'le
arn not', 'learn one', 'learn play', 'learn read', 'learn school', 'l
earn science', 'learn skills', 'learn something', 'learn students', '
learn successful', 'learn technology', 'learn together', 'learn use',
'learn using', 'learn want', 'learn way', 'learn work', 'learn world
', 'learned', 'learner', 'learners', 'learners classroom', 'learners
come', 'learners continue', 'learners love', 'learners many', 'learne
rs nannan', 'learners need', 'learners school', 'learners students',
'learning', 'learning activities', 'learning also', 'learning center
', 'learning centers', 'learning class', 'learning classroom', 'learn
ing come', 'learning community', 'learning different', 'learning disa
bilities', 'learning english', 'learning environment', 'learning expe
rience', 'learning experiences', 'learning fun', 'learning games', 'l
earning goals', 'learning hands', 'learning help', 'learning love', '
learning many', 'learning materials', 'learning math', 'learning much
', 'learning nannan', 'learning need', 'learning needs', 'learning ne
w', 'learning not', 'learning opportunities', 'learning play', 'learn
ing process', 'learning project', 'learning read', 'learning reading
', 'learning school', 'learning science', 'learning skills', 'learnin
g space', 'learning students', 'learning style', 'learning styles', '
learning take', 'learning technology', 'learning time', 'learning too
ls', 'learning use', 'learning using', 'learning want', 'learning wel
l', 'learning work', 'learning would', 'learns', 'least', 'leave', 'l
eaves', 'leaving', 'led', 'left', 'lego', 'legos', 'legs', 'less', 'l
esson', 'lessons', 'lessons students', 'let', 'let alone', 'let stude
nts', 'lets', 'letter', 'letters', 'letters sounds', 'letting', 'leve
l', 'level many', 'level reading', 'level students', 'leveled', 'leve
led books', 'levels', 'levels students', 'libraries', 'library', 'lib
rary books', 'library students', 'life', 'life experiences', 'life lo
ng', 'life may', 'life nannan', 'life ready', 'life skills', 'life st
udents', 'lifelong', 'lifelong learners', 'lifestyle', 'lifetime', 'l
ight', 'lights', 'like', 'like move', 'like provide', 'like students
', 'like use', 'likely', 'limit', 'limitations', 'limited', 'limited
access', 'limited resources', 'limits', 'line', 'lines', 'list', 'lis

ten', 'listen stories', 'listening', 'listening center', 'literacy',
'literacy centers', 'literacy math', 'literacy skills', 'literally',
'literary', 'literate', 'literature', 'little', 'little bit', 'little
learners', 'little no', 'little ones', 'live', 'live high', 'live low
', 'live poverty', 'live rural', 'live small', 'lively', 'lives', 'li
ves better', 'lives however', 'lives nannan', 'lives students', 'livi
ng', 'living poverty', 'local', 'located', 'located high', 'located l
ow', 'located rural', 'location', 'long', 'long learners', 'long peri
ods', 'long readers', 'long term', 'long way', 'longer', 'look', 'loo
k forward', 'look like', 'looking', 'looking forward', 'looking keep
', 'looking new', 'looking ways', 'looks', 'los', 'los angeles', 'los
e', 'lost', 'lot', 'lot students', 'lot time', 'lots', 'lots positive
', 'loud', 'love', 'love able', 'love books', 'love come', 'love comi
ng', 'love explore', 'love get', 'love hands', 'love learn', 'love le
arning', 'love lots', 'love move', 'love music', 'love opportunity',
'love play', 'love read', 'love reading', 'love school', 'love scienc
e', 'love see', 'love share', 'love students', 'love teaching', 'love
technology', 'love use', 'love using', 'love work', 'love working', '
loved', 'loves', 'loving', 'low', 'low economic', 'low income', 'low
socio', 'low socioeconomic', 'lower', 'lower income', 'lowest', 'luck
y', 'lucky enough', 'lunch', 'lunch based', 'lunch breakfast', 'lunch
despite', 'lunch many', 'lunch program', 'lunch school', 'lunch stude
nts', 'lunches', 'machine', 'machines', 'made', 'magazine', 'magazine
s', 'magic', 'magical', 'magnet', 'magnet school', 'magnetic', 'magne
ts', 'main', 'mainly', 'maintain', 'maintaining', 'major', 'majority
', 'majority students', 'make', 'make better', 'make choices', 'make
classroom', 'make connections', 'make difference', 'make ends', 'make
feel', 'make happen', 'make huge', 'make learning', 'make possible',
'make reading', 'make school', 'make students', 'make sure', 'make wo
rld', 'maker', 'makerspace', 'makes', 'makes difficult', 'makes learn
ing', 'makes students', 'making', 'making sure', 'manage', 'managemen
t', 'manipulate', 'manipulative', 'manipulatives', 'manner', 'many',
'many books', 'many challenges', 'many children', 'many come', 'many
different', 'many english', 'many families', 'many first', 'many hard
ships', 'many kids', 'many learning', 'many not', 'many obstacles', '
many opportunities', 'many parents', 'many raised', 'many resources',
'many students', 'many things', 'many times', 'many us', 'many ways',
'many years', 'maps', 'markers', 'master', 'mastered', 'mastering', '
mastery', 'match', 'material', 'materials', 'materials allow', 'mater
ials classroom', 'materials give', 'materials help', 'materials make
', 'materials need', 'materials needed', 'materials not', 'materials
project', 'materials provide', 'materials requested', 'materials requ
esting', 'materials students', 'materials supplies', 'materials used
', 'materials would', 'math', 'math centers', 'math class', 'math con
cepts', 'math facts', 'math games', 'math literacy', 'math manipulati
ves', 'math problems', 'math reading', 'math science', 'math skills',
'math students', 'mathematical', 'mathematicians', 'mathematics', 'ma
ts', 'matter', 'maximize', 'maximum', 'may', 'may face', 'may not', '
may prevent', 'maybe', 'meal', 'meals', 'mean', 'meaning', 'meaningfu
l', 'meaningful learning', 'means', 'means students', 'meant', 'measu
re', 'media', 'media center', 'medium', 'meet', 'meet individual', 'm
eet needs', 'meet students', 'meeting', 'meetings', 'meets', 'member
', 'members', 'members society', 'memorable', 'memories', 'memory', '
men', 'mental', 'message', 'met', 'method', 'methods', 'mexico', 'mid
dle', 'middle class', 'middle school', 'might', 'might not', 'miles',
'military', 'military families', 'mind', 'minds', 'mindset', 'mine',

```
'mini', 'minimal', 'minis', 'minority', 'minute', 'minute walk', 'min
utes', 'minutes day', 'miss', 'missing', 'mission', 'mistakes', 'mix
', 'mixed', 'mixture', 'mobile', 'model', 'modeling', 'models', 'mode
rate', 'modern', 'moment', 'moments', 'money', 'monitor', 'month', 'm
onths', 'morning', 'mostly', 'motion', 'motivate', 'motivate students
', 'motivated', 'motivated learn', 'motivates', 'motivating', 'motiva
tion', 'motor', 'motor skills', 'motto', 'mouse', 'move', 'move aroun
d', 'move learn', 'move learning', 'move love', 'moved', 'movement',
'movements', 'movies', 'moving', 'moving around', 'mrs', 'ms', 'much
', 'much easier', 'much fun', 'much needed', 'much possible', 'much s
tudents', 'much time', 'multi', 'multicultural', 'multiple', 'multipl
ication', 'multitude', 'muscles', 'music', 'music class', 'music stud
ents', 'musical', 'musicians', 'must', 'name', 'names', 'nannan', 'na
tion', 'national', 'native', 'native american', 'natural', 'naturally
', 'nature', 'navigate', 'near', 'nearly', 'necessary', 'necessary ma
terials', 'necessary supplies', 'necessary tools', 'necessities', 'ne
cessity', 'need', 'need able', 'need access', 'need additional', 'nee
d basic', 'need books', 'need classroom', 'need extra', 'need get', '
need hands', 'need help', 'need know', 'need learn', 'need little', '
need make', 'need many', 'need materials', 'need move', 'need movemen
t', 'need nannan', 'need new', 'need opportunities', 'need opportunit
y', 'need order', 'need resources', 'need students', 'need succeed',
'need successful', 'need supplies', 'need support', 'need technology
', 'need tools', 'need variety', 'need work', 'needed', 'needing', 'n
eeds', 'needs classroom', 'needs many', 'needs met', 'needs nannan',
'needs school', 'needs students', 'negative', 'neighborhood', 'neighb
orhood school', 'neighborhood students', 'neighborhoods', 'never', 'n
ew', 'new books', 'new challenges', 'new classroom', 'new concepts',
'new country', 'new exciting', 'new experiences', 'new ideas', 'new i
nformation', 'new language', 'new learning', 'new materials', 'new sc
hool', 'new skills', 'new students', 'new technology', 'new things',
'new vocabulary', 'new way', 'new ways', 'new world', 'new york', 'ne
wly', 'news', 'next', 'next generation', 'next level', 'next year', '
nice', 'night', 'nine', 'ninety', 'no', 'no longer', 'no matter', 'no
one', 'noise', 'non', 'non fiction', 'none', 'nonfiction', 'normal',
'normally', 'north', 'north carolina', 'northern', 'not', 'not able',
'not access', 'not afford', 'not allow', 'not always', 'not available
', 'not books', 'not come', 'not easy', 'not enough', 'not even', 'no
t exposed', 'not get', 'not give', 'not help', 'not know', 'not learn
', 'not let', 'not like', 'not lot', 'not make', 'not many', 'not muc
h', 'not necessary', 'not need', 'not one', 'not opportunity', 'not p
rovide', 'not read', 'not receive', 'not resources', 'not school', 'n
ot sit', 'not speak', 'not stop', 'not students', 'not teach', 'not t
hink', 'not understand', 'not use', 'not wait', 'not want', 'not work
', 'not worry', 'not yet', 'note', 'notebook', 'notebooks', 'notes',
'nothing', 'notice', 'noticed', 'novel', 'novels', 'number', 'number
sense', 'number students', 'numbers', 'numerous', 'nurture', 'nurturi
ng', 'nutrition', 'objectives', 'objects', 'observe', 'obstacle', 'ob
stacles', 'obtain', 'occur', 'odds', 'offer', 'offer students', 'offe
red', 'offering', 'offers', 'office', 'often', 'often come', 'often n
ot', 'often students', 'often times', 'old', 'old children', 'old stu
dents', 'older', 'olds', 'one', 'one another', 'one best', 'one day',
'one favorite', 'one hundred', 'one important', 'one one', 'one place
', 'one school', 'one student', 'one students', 'one thing', 'one tim
e', 'one way', 'ones', 'online', 'onto', 'open', 'opened', 'opening',
'opens', 'opinions', 'opportunities', 'opportunities explore', 'oppor
```

```
tunities learn', 'opportunities students', 'opportunity', 'opportunit
y create', 'opportunity experience', 'opportunity explore', 'opportun
ity learn', 'opportunity move', 'opportunity practice', 'opportunity
read', 'opportunity students', 'opportunity use', 'opportunity work',
'option', 'options', 'options allow', 'options classroom', 'options s
tudents', 'oral', 'order', 'order help', 'order learn', 'order make',
'order students', 'order successful', 'org', 'organization', 'organiz
ational', 'organize', 'organized', 'organizing', 'original', 'osmo',
'others', 'others students', 'otherwise', 'outdated', 'outdoor', 'out
doors', 'outlet', 'outside', 'outside box', 'outside classroom', 'out
side school', 'outstanding', 'overall', 'overcome', 'overwhelming', '
ownership', 'ownership learning', 'pace', 'paced', 'pads', 'page', 'p
ages', 'paint', 'painting', 'pair', 'pairs', 'paper', 'paper pencil',
'papers', 'parent', 'parent homes', 'parent households', 'parental',
'parents', 'parents not', 'parents students', 'parents work', 'part',
'part classroom', 'part day', 'part learning', 'part school', 'partic
ipants', 'participate', 'participating', 'participation', 'particular
', 'particularly', 'partner', 'partners', 'parts', 'pass', 'passion',
'passion learning', 'passionate', 'past', 'past year', 'past years',
'path', 'path academic', 'patterns', 'pay', 'pay attention', 'pe', 'p
eer', 'peers', 'peers students', 'pen', 'pencil', 'pencils', 'pens',
'people', 'per', 'percent', 'percent students', 'percentage', 'percen
tage students', 'perfect', 'perform', 'performance', 'performances',
'performing', 'perhaps', 'period', 'periods', 'periods time', 'persev
erance', 'persevere', 'person', 'personal', 'personalities', 'persona
lity', 'personalized', 'personally', 'perspective', 'perspectives', '
philadelphia', 'phone', 'phonemic', 'phones', 'phonics', 'photography
', 'photos', 'physical', 'physical activity', 'physical education', '
physically', 'physically active', 'pick', 'picture', 'pictures', 'pie
ce', 'pieces', 'pillows', 'place', 'place learn', 'place sit', 'place
students', 'placed', 'places', 'plan', 'plan use', 'planning', 'plans
', 'plant', 'plants', 'plastic', 'play', 'play games', 'play students
', 'played', 'player', 'players', 'playground', 'playing', 'playing f
ield', 'playing games', 'plays', 'please', 'please consider', 'please
help', 'pleasure', 'plenty', 'plus', 'pocket', 'pockets', 'point', 'p
oints', 'poor', 'poorest', 'popular', 'population', 'population stude
nts', 'portable', 'portion', 'position', 'positive', 'positive attent
ion', 'positive attitude', 'positive impact', 'positive learning', 'p
ositive way', 'positively', 'possibilities', 'possibilities endless',
'possibility', 'possible', 'possible students', 'possibly', 'post', '
posters', 'posture', 'potential', 'potential growth', 'potential stud
ents', 'poverty', 'poverty area', 'poverty level', 'poverty line', 'p
overty rate', 'poverty school', 'poverty stricken', 'poverty students
', 'power', 'powerful', 'practice', 'practice math', 'practice readin
g', 'practice skills', 'practices', 'practicing', 'pre', 'pre kinderg
arten', 'precious', 'prefer', 'prek', 'preparation', 'prepare', 'prep
are future', 'prepare students', 'prepared', 'preparing', 'preschool
', 'preschoolers', 'present', 'presentation', 'presentations', 'prese
nted', 'pressure', 'pretty', 'prevent', 'prevent getting', 'previous
', 'previously', 'price', 'price lunch', 'priced', 'priceless', 'prid
e', 'primarily', 'primary', 'print', 'printed', 'printer', 'printing
', 'prior', 'priority', 'privilege', 'privileged', 'pro', 'problem',
'problem solve', 'problem solvers', 'problem solving', 'problems', 'p
rocess', 'process students', 'processing', 'produce', 'product', 'pro
duction', 'productive', 'productivity', 'products', 'professional', '
proficiency', 'proficient', 'program', 'program school', 'program stu
```

```
dents', 'programming', 'programs', 'progress', 'project', 'project al
low', 'project based', 'project funded', 'project give', 'project hel
p', 'project helping', 'project improve', 'project make', 'project na
nnan', 'project not', 'project provide', 'project students', 'project
would', 'projector', 'projects', 'projects students', 'promote', 'pro
motes', 'promoting', 'proper', 'properly', 'protect', 'proud', 'prove
', 'proven', 'provide', 'provide best', 'provide life', 'provide many
', 'provide materials', 'provide opportunities', 'provide opportunity
', 'provide safe', 'provide students', 'provided', 'provides', 'provi
des students', 'providing', 'providing students', 'public', 'public s
chool', 'publish', 'pull', 'purchase', 'purchased', 'purchasing', 'pu
rpose', 'purposes', 'pursue', 'push', 'pushing', 'put', 'puts', 'putt
ing', 'puzzles', 'qualifies', 'qualify', 'qualify free', 'qualifying
', 'qualifying free', 'quality', 'quality education', 'question', 'qu
estions', 'quick', 'quickly', 'quiet', 'quietly', 'quite', 'quizzes',
'quote', 'race', 'races', 'raise', 'raised', 'raised single', 'range
', 'range abilities', 'ranging', 'rarely', 'rate', 'rates', 'rather',
'reach', 'reach full', 'reach goal', 'reach goals', 'reach students',
'reaching', 'read', 'read aloud', 'read alouds', 'read book', 'read b
ooks', 'read independently', 'read learn', 'read love', 'read nannan
', 'read students', 'read write', 'reader', 'readers', 'readers nanna
n', 'readers students', 'readers writers', 'readily', 'readily availa
ble', 'readiness', 'reading', 'reading also', 'reading book', 'readin
g books', 'reading comprehension', 'reading fluency', 'reading grade
', 'reading groups', 'reading learning', 'reading level', 'reading le
vels', 'reading material', 'reading materials', 'reading math', 'read
ing nannan', 'reading not', 'reading program', 'reading reading', 're
ading skills', 'reading students', 'reading time', 'reading writing',
'reads', 'ready', 'ready learn', 'ready take', 'real', 'real life', '
real world', 'reality', 'realize', 'realized', 'really', 'really enjo
y', 'really need', 'really want', 'reason', 'reasoning', 'reasons', '
receive', 'receive free', 'received', 'receives', 'receives free', 'r
eceiving', 'receiving free', 'recent', 'recently', 'recess', 'recess
time', 'recognition', 'recognize', 'record', 'recording', 'reduce', '
reduced', 'reduced breakfast', 'reduced lunch', 'reduced lunches', 'r
educed price', 'reduced priced', 'reference', 'reflect', 'regardless
', 'regular', 'regular basis', 'regular education', 'regularly', 'rei
nforce', 'reinforcement', 'relate', 'related', 'relationship', 'relat
ionships', 'relax', 'relaxed', 'release', 'relevant', 'reluctant', 'r
eluctant readers', 'rely', 'remain', 'remaining', 'remember', 'remind
', 'replace', 'reports', 'represent', 'represented', 'request', 'requ
ested', 'requesting', 'require', 'required', 'requirements', 'require
s', 'research', 'research projects', 'research shown', 'research show
s', 'researched', 'researching', 'resilient', 'resource', 'resources
', 'resources available', 'resources help', 'resources home', 'resour
ces need', 'resources needed', 'resources students', 'respect', 'resp
ectful', 'respond', 'response', 'responsibility', 'responsible', 'res
t', 'rest lives', 'result', 'results', 'retain', 'return', 'review',
'reward', 'rewarding', 'rewards', 'rich', 'ride', 'right', 'right too
ls', 'rigor', 'rigorous', 'rigorous academics', 'rise', 'risk', 'risk
s', 'road', 'robot', 'robotics', 'robots', 'rock', 'role', 'role mode
ls', 'roll', 'rolling', 'room', 'room students', 'ropes', 'rotate', '
rotation', 'rotations', 'rough', 'round', 'rounded', 'routine', 'rout
ines', 'rug', 'rules', 'run', 'running', 'rural', 'rural area', 'rura
l community', 'rural school', 'sadly', 'safe', 'safe comfortable', 's
afe environment', 'safe learning', 'safe place', 'safely', 'safety',
```

```
'said', 'san', 'sand', 'save', 'savvy', 'saw', 'say', 'saying', 'says
', 'scale', 'schedule', 'scholars', 'scholastic', 'scholastic news',
'school', 'school 100', 'school also', 'school building', 'school chi
ldren', 'school classroom', 'school college', 'school come', 'school
community', 'school considered', 'school currently', 'school day', 's
chool district', 'school diverse', 'school eager', 'school environmen
t', 'school every', 'school everyday', 'school excited', 'school expe
rience', 'school family', 'school feel', 'school first', 'school full
', 'school great', 'school help', 'school high', 'school home', 'scho
ol kids', 'school large', 'school learn', 'school learning', 'school
library', 'school life', 'school limited', 'school located', 'school
love', 'school low', 'school made', 'school majority', 'school many',
'school means', 'school nannan', 'school need', 'school new', 'school
not', 'school offers', 'school often', 'school one', 'school place',
'school population', 'school program', 'school provides', 'school rea
dy', 'school receive', 'school rural', 'school safe', 'school school
', 'school serves', 'school setting', 'school small', 'school student
', 'school students', 'school supplies', 'school system', 'school tea
ch', 'school title', 'school urban', 'school want', 'school well', 's
chool wide', 'school without', 'school work', 'school would', 'school
year', 'schooling', 'schools', 'science', 'science class', 'science c
lassroom', 'science concepts', 'science math', 'science social', 'sci
ence standards', 'science students', 'science technology', 'scientifi
c', 'scientist', 'scientists', 'scissors', 'scores', 'screen', 'searc
h', 'searching', 'season', 'seat', 'seated', 'seating', 'seating allo
w', 'seating allows', 'seating choices', 'seating classroom', 'seatin
g option', 'seating options', 'seating students', 'seats', 'second',
'second grade', 'second graders', 'second language', 'second year', '
secondary', 'section', 'secure', 'see', 'see students', 'see typical
', 'see world', 'seeing', 'seek', 'seeking', 'seem', 'seems', 'seen',
'select', 'selected', 'selection', 'self', 'self confidence', 'self c
ontained', 'self esteem', 'semester', 'send', 'seniors', 'sense', 'se
nse community', 'senses', 'sensory', 'sensory needs', 'sent', 'senten
ce', 'sentences', 'separate', 'series', 'serious', 'serve', 'serve st
udents', 'served', 'serves', 'serves students', 'service', 'services
', 'services students', 'serving', 'sessions', 'set', 'set goals', 's
et students', 'sets', 'setting', 'setting students', 'settings', 'seu
ss', 'seven', 'seventh', 'several', 'several challenges', 'several di
fferent', 'several students', 'several years', 'severe', 'shape', 'sh
apes', 'share', 'share ideas', 'share work', 'shared', 'sharing', 'sh
arpener', 'sheets', 'shelf', 'shelves', 'shine', 'shoes', 'short', 's
how', 'show students', 'showcase', 'showing', 'shown', 'shows', 'show
s students', 'siblings', 'side', 'sight', 'sight word', 'sight words
', 'sign', 'significant', 'significantly', 'silly', 'similar', 'simpl
e', 'simple provide', 'simply', 'simply not', 'simultaneously', 'sinc
e', 'since students', 'sing', 'singing', 'single', 'single day', 'sin
gle parent', 'sit', 'sit floor', 'sit still', 'site', 'sites', 'sitti
ng', 'sitting desk', 'sitting still', 'situation', 'situations', 'six
', 'six year', 'sixth', 'sixth grade', 'size', 'sizes', 'skill', 'ski
lls', 'skills also', 'skills help', 'skills learn', 'skills learning
', 'skills nannan', 'skills necessary', 'skills need', 'skills needed
', 'skills reading', 'skills students', 'skills taught', 'skills use
', 'skills using', 'skills well', 'slides', 'slow', 'slowly', 'small
', 'small community', 'small group', 'small groups', 'small rural', '
small school', 'small town', 'smaller', 'smart', 'smile', 'smile face
', 'smiles', 'smiles faces', 'smiling', 'snack', 'snacks', 'soak', 's
```

```
oar', 'soccer', 'social', 'social economic', 'social emotional', 'soc
ial skills', 'social studies', 'socially', 'society', 'socio', 'socio
economic', 'socioeconomic', 'socioeconomic background', 'socioeconomi
c backgrounds', 'socioeconomic status', 'socioeconomically', 'soft',
'software', 'solid', 'solution', 'solutions', 'solve', 'solve problem
s', 'solvers', 'solving', 'solving skills', 'someone', 'something', '
something new', 'sometimes', 'song', 'songs', 'soon', 'sort', 'sound
', 'sounds', 'source', 'sources', 'south', 'southern', 'space', 'spac
e students', 'spaces', 'spanish', 'spanish speaking', 'spark', 'speak
', 'speak english', 'speakers', 'speaking', 'special', 'special educa
tion', 'special needs', 'specialized', 'specific', 'specific learning
', 'specifically', 'spectrum', 'spectrum disorder', 'speech', 'speech
language', 'speed', 'spelling', 'spend', 'spend lot', 'spend time', '
spending', 'spent', 'spirit', 'spite', 'spoken', 'sponges', 'spontane
ous', 'sport', 'sports', 'spot', 'spots', 'spread', 'spring', 'stabil
ity', 'stability balls', 'stable', 'staff', 'stage', 'stamina', 'stan
d', 'standard', 'standardized', 'standards', 'standards students', 's
tanding', 'standing desks', 'stands', 'star', 'stars', 'start', 'star
t day', 'start school', 'start year', 'started', 'starting', 'starts
', 'state', 'state standards', 'states', 'station', 'stations', 'stat
us', 'status things', 'stay', 'stay active', 'stay engaged', 'stay fo
cused', 'stay organized', 'stay task', 'staying', 'steam', 'stem', 's
tem activities', 'stem science', 'step', 'steps', 'sticks', 'still',
'still learning', 'still need', 'stimulate', 'stimulation', 'stock',
'stool', 'stools', 'stools allow', 'stools help', 'stop', 'storage',
'store', 'store day', 'stories', 'stories read', 'story', 'strategies
', 'strategy', 'strength', 'strengthen', 'strengths', 'stress', 'stre
tch', 'stricken', 'strive', 'strive best', 'strive make', 'strive pro
vide', 'strives', 'striving', 'strong', 'strong foundation', 'stronge
r', 'structure', 'structured', 'structures', 'struggle', 'struggle re
ading', 'struggles', 'struggling', 'struggling readers', 'struggling
students', 'student', 'student able', 'student body', 'student center
ed', 'student choice', 'student class', 'student classroom', 'student
engagement', 'student learning', 'student needs', 'student population
', 'student school', 'student success', 'student use', 'student work
', 'students', 'students ability', 'students able', 'students absolut
ely', 'students academic', 'students access', 'students achieve', 'st
udents active', 'students actively', 'students ages', 'students allow
ed', 'students already', 'students also', 'students always', 'student
s amazing', 'students art', 'students ask', 'students asked', 'studen
ts asking', 'students attend', 'students attention', 'students autism
', 'students awesome', 'students become', 'students begin', 'students
beginning', 'students believe', 'students benefit', 'students best',
'students better', 'students books', 'students bright', 'students bri
ng', 'students build', 'students building', 'students cannot', 'stude
nts chance', 'students children', 'students choice', 'students choose
', 'students class', 'students classroom', 'students collaborate', 's
tudents come', 'students comfortable', 'students coming', 'students c
ommunity', 'students complete', 'students constantly', 'students cont
inue', 'students could', 'students create', 'students creative', 'stu
dents curious', 'students currently', 'students daily', 'students day
', 'students deserve', 'students desire', 'students develop', 'studen
ts different', 'students difficulty', 'students disabilities', 'stude
nts diverse', 'students eager', 'students easily', 'students economic
ally', 'students encouraged', 'students energetic', 'students engage
', 'students engaged', 'students engaging', 'students english', 'stud
```

```
ents enjoy', 'students enter', 'students enthusiastic', 'students esp
ecially', 'students even', 'students ever', 'students every', 'studen
ts excel', 'students excited', 'students expected', 'students experie
nce', 'students explore', 'students exposed', 'students extremely', '
students face', 'students faced', 'students fall', 'students families
', 'students feel', 'students find', 'students first', 'students focu
s', 'students focused', 'students free', 'students full', 'students f
un', 'students future', 'students gain', 'students get', 'students gi
ve', 'students given', 'students go', 'students going', 'students gra
de', 'students grades', 'students great', 'students group', 'students
grow', 'students hands', 'students hard', 'students hear', 'students
help', 'students high', 'students highly', 'students however', 'stude
nts improve', 'students increase', 'students individual', 'students i
nquisitive', 'students interact', 'students interested', 'students in
volved', 'students keep', 'students kindergarten', 'students know', '
students lack', 'students learn', 'students learning', 'students life
', 'students like', 'students limited', 'students listen', 'students
little', 'students live', 'students lives', 'students living', 'stude
nts look', 'students lot', 'students love', 'students low', 'students
make', 'students many', 'students materials', 'students math', 'stude
nts may', 'students meet', 'students mostly', 'students motivated', '
students move', 'students moving', 'students much', 'students multipl
e', 'students must', 'students nannan', 'students need', 'students ne
eds', 'students never', 'students new', 'students no', 'students not
', 'students often', 'students one', 'students opportunities', 'stude
nts opportunity', 'students options', 'students parents', 'students p
art', 'students participate', 'students place', 'students play', 'stu
dents practice', 'students pre', 'students provide', 'students qualif
y', 'students range', 'students reach', 'students read', 'students re
ading', 'students ready', 'students really', 'students receive', 'stu
dents receiving', 'students require', 'students required', 'students
research', 'students resources', 'students room', 'students safe', 's
tudents school', 'students second', 'students see', 'students self',
'students serve', 'students share', 'students show', 'students sit',
'students small', 'students speak', 'students special', 'students spe
nd', 'students staff', 'students start', 'students stay', 'students s
till', 'students strive', 'students struggle', 'students struggling',
'students students', 'students succeed', 'students success', 'student
s successful', 'students take', 'students taught', 'students teach',
'students teachers', 'students technology', 'students think', 'studen
ts thrive', 'students throughout', 'students time', 'students title',
'students tools', 'students truly', 'students try', 'students unable
', 'students understand', 'students understanding', 'students unique
', 'students use', 'students used', 'students using', 'students utili
ze', 'students variety', 'students various', 'students varying', 'stu
dents visual', 'students walk', 'students want', 'students way', 'stu
dents well', 'students wide', 'students wonderful', 'students work',
'students working', 'students world', 'students would', 'students wri
te', 'students year', 'students years', 'students young', 'studies',
'studies science', 'studies shown', 'study', 'studying', 'sturdy', 's
tyle', 'styles', 'subject', 'subject areas', 'subjects', 'subscriptio
n', 'subtraction', 'suburban', 'succeed', 'succeed school', 'succeed
students', 'success', 'success nannan', 'success school', 'success st
udents', 'successes', 'successful', 'successful classroom', 'successf
ul learning', 'successful nannan', 'successful one', 'successful scho
ol', 'successful students', 'successfully', 'summer', 'super', 'suppl
```

```
ement', 'supplies', 'supplies allow', 'supplies help', 'supplies need
', 'supplies needed', 'supplies school', 'supplies students', 'supply
', 'support', 'support home', 'support learning', 'support nannan', '
support need', 'support students', 'supported', 'supporting', 'suppor
tive', 'supports', 'sure', 'sure students', 'surface', 'surprise', 's
urrounded', 'surrounding', 'surroundings', 'sweet', 'system', 'system
s', 'table', 'tables', 'tablet', 'tablets', 'tackle', 'tactile', 'tak
e', 'take care', 'take granted', 'take home', 'take ownership', 'take
place', 'take pride', 'take risks', 'take turns', 'taken', 'takes', '
taking', 'taking time', 'talented', 'talents', 'talk', 'talking', 'ta
ngible', 'tape', 'target', 'task', 'task hand', 'tasks', 'taught', 't
each', 'teach high', 'teach kindergarten', 'teach low', 'teach school
', 'teach small', 'teach students', 'teach title', 'teacher', 'teache
r low', 'teacher students', 'teacher title', 'teacher want', 'teacher
s', 'teachers school', 'teachers students', 'teaches', 'teaching', 't
eaching learning', 'teaching students', 'team', 'team building', 'tea
m work', 'teams', 'teamwork', 'tech', 'techniques', 'technological',
'technologically', 'technologies', 'technology', 'technology also', '
technology available', 'technology classroom', 'technology daily', 't
echnology engineering', 'technology hands', 'technology help', 'techn
ology home', 'technology learning', 'technology many', 'technology na
nnan', 'technology not', 'technology resources', 'technology school',
'technology skills', 'technology students', 'technology use', 'techno
logy would', 'tell', 'telling', 'ten', 'tend', 'term', 'terms', 'test
', 'test scores', 'testing', 'tests', 'texas', 'text', 'textbook', 't
extbooks', 'texts', 'thank', 'thank advance', 'thank considering', 't
hank helping', 'thank much', 'thank nannan', 'thankful', 'thanks', 't
heme', 'themes', 'therapy', 'therefore', 'thing', 'thing common', 'th
ings', 'things like', 'things may', 'things simple', 'things students
', 'think', 'think critically', 'think outside', 'thinkers', 'thinkin
g', 'thinking problem', 'thinking skills', 'third', 'third grade', 't
hird graders', 'thirst', 'thirst knowledge', 'thirty', 'though', 'tho
ugh students', 'thought', 'thoughtful', 'thoughts', 'three', 'thrille
d', 'thrive', 'throughout', 'throughout day', 'throughout school', 't
hroughout year', 'thus', 'tight', 'tight knit', 'tiles', 'time', 'tim
e classroom', 'time day', 'time learning', 'time nannan', 'time not',
'time read', 'time school', 'time sitting', 'time spent', 'time stude
nts', 'times', 'times day', 'times students', 'tiny', 'tired', 'title
', 'title one', 'title school', 'titles', 'today', 'today students',
'today world', 'together', 'together create', 'together students', 't
old', 'tomorrow', 'took', 'tool', 'tool students', 'tools', 'tools he
lp', 'tools need', 'tools students', 'top', 'topic', 'topics', 'total
', 'touch', 'tough', 'toward', 'towards', 'town', 'toys', 'track', 't
raditional', 'traditional classroom', 'training', 'traits', 'transfer
', 'transform', 'transient', 'transition', 'transitional', 'transport
ation', 'trauma', 'travel', 'treat', 'tremendous', 'tremendously', 't
ried', 'trip', 'trips', 'trouble', 'true', 'truly', 'truly believe',
'try', 'try best', 'try make', 'try new', 'try provide', 'trying', 't
urn', 'turning', 'turns', 'tv', 'twenty', 'twice', 'two', 'two studen
ts', 'two years', 'type', 'types', 'typical', 'typical day', 'typical
minute', 'typically', 'typing', 'ultimate', 'ultimate goal', 'ultimat
ely', 'unable', 'uncomfortable', 'understand', 'understanding', 'unfo
rtunately', 'unfortunately not', 'uniforms', 'unique', 'unique learni
ng', 'unit', 'united', 'united states', 'units', 'university', 'unlim
ited', 'upcoming', 'updated', 'upon', 'upper', 'urban', 'urban area',
'urban school', 'us', 'us get', 'us see', 'use', 'use books', 'use ch
```

romebooks', 'use classroom', 'use computer', 'use computers', 'use da
ily', 'use google', 'use hands', 'use help', 'use ipad', 'use ipads',
'use many', 'use materials', 'use new', 'use resources', 'use student
s', 'use tablets', 'use technology', 'used', 'used classroom', 'used
daily', 'used help', 'used students', 'useful', 'uses', 'using', 'usi
ng hands', 'using ipads', 'using materials', 'using technology', 'usu
ally', 'utilize', 'utilized', 'utilizing', 'valuable', 'value', 'valu
ed', 'values', 'varied', 'variety', 'variety backgrounds', 'variety b
ooks', 'variety different', 'variety learning', 'variety ways', 'vari
ous', 'various backgrounds', 'vary', 'varying', 'vast', 'vegetables',
'verbal', 'via', 'vibrant', 'video', 'videos', 'view', 'violence', 'v
irtual', 'vision', 'visit', 'visual', 'visual learners', 'visualize',
'visually', 'visuals', 'vital', 'vocabulary', 'voice', 'voices', 'vol
ume', 'wait', 'wait see', 'waiting', 'walk', 'walk classroom', 'walk
door', 'walking', 'walks', 'walks life', 'wall', 'walls', 'want', 'wa
nt able', 'want best', 'want classroom', 'want continue', 'want creat
e', 'want give', 'want help', 'want know', 'want learn', 'want make',
'want provide', 'want read', 'want school', 'want see', 'want student
s', 'want succeed', 'want use', 'wanted', 'wanting', 'wants', 'warm',
'washington', 'waste', 'watch', 'watching', 'water', 'way', 'way get
', 'way help', 'way hopeful', 'way learn', 'way learning', 'way nanna
n', 'way students', 'way teach', 'ways', 'ways help', 'ways learn', '
ways students', 'wear', 'weather', 'web', 'website', 'websites', 'wee
k', 'week students', 'weekend', 'weekends', 'weekly', 'weeks', 'weigh
t', 'welcome', 'welcoming', 'well', 'well nannan', 'well rounded', 'w
ell school', 'well students', 'wellness', 'went', 'west', 'whatever',
'whenever', 'whether', 'white', 'white board', 'white boards', 'white
board', 'whiteboards', 'whole', 'whole child', 'whole class', 'whole
group', 'whole new', 'whose', 'wide', 'wide range', 'wide variety', '
wiggle', 'wiggles', 'wiggling', 'wiggly', 'willing', 'willing learn',
'willingness', 'win', 'winning', 'winter', 'wireless', 'wish', 'withi
n', 'within classroom', 'within school', 'without', 'wobble', 'wobble
chairs', 'wobble stools', 'women', 'wonder', 'wonderful', 'wonderful
group', 'wonderful students', 'wonderfully', 'word', 'word work', 'wo
rds', 'work', 'work best', 'work classroom', 'work collaboratively',
'work groups', 'work hard', 'work independently', 'work learn', 'work
many', 'work nannan', 'work not', 'work school', 'work small', 'work
students', 'work team', 'work time', 'work title', 'work together', '
work well', 'worked', 'worked hard', 'workers', 'workforce', 'working
', 'working class', 'working hard', 'working independently', 'working
small', 'working students', 'working together', 'works', 'works best
', 'worksheets', 'workshop', 'world', 'world around', 'world better',
'world live', 'world nannan', 'world problems', 'world students', 'wo
rld technology', 'worlds', 'worn', 'worry', 'worth', 'would', 'would
able', 'would allow', 'would also', 'would benefit', 'would give', 'w
ould great', 'would greatly', 'would help', 'would like', 'would love
', 'would make', 'would never', 'would not', 'would provide', 'would
really', 'would use', 'would used', 'write', 'writer', 'writers', 'wr
iting', 'writing math', 'writing skills', 'writing students', 'writte
n', 'wrong', 'year', 'year first', 'year learning', 'year long', 'yea

In [165]:
```
train_title_bow = VectorizingTextData('preprocessed_titles', project_d
ata_train, project_data_train)
test_title_bow = VectorizingTextData('preprocessed_titles', project_da
ta_train, project_data_test)

print("Shape of train data matrix after one hot encoding ",train_title
_bow.shape)
print("Shape of test data matrix after one hot encoding ",test_title_b
ow.shape)

title_features = fnGetTextFeatures('preprocessed_titles', project_data
_train, project_data_train)
print(title_features)
```

```
Shape of train data matrix after one hot encoding  (20100, 1668)
Shape of test data matrix after one hot encoding  (9900, 1668)
['05', '16', '1st', '1st grade', '1st graders', '2016', '2017', '21st
', '21st century', '2nd', '2nd grade', '2nd graders', '3d', '3d print
er', '3d printing', '3doodler', '3rd', '3rd grade', '3rd graders', '4
th', '4th grade', '4th graders', '5th', '5th grade', '5th graders', '
6th', '6th grade', '8th', 'about', 'about it', 'about our', 'academic
', 'access', 'accessible', 'accessing', 'accessories', 'achieve', 'ac
hievement', 'action', 'active', 'active bodies', 'active learners', '
active learning', 'active minds', 'active seating', 'active students
', 'activities', 'activity', 'add', 'adding', 'adventure', 'adventure
s', 'after', 'again', 'age', 'ahead', 'air', 'algebra', 'alive', 'all
', 'all about', 'all day', 'all learners', 'all students', 'along', '
aloud', 'alouds', 'alphabet', 'alternative', 'alternative seating', '
amazing', 'america', 'american', 'an', 'an apple', 'an ipad', 'ancien
t', 'and', 'another', 'ants', 'anything', 'ap', 'app', 'apple', 'appl
e day', 'apples', 'approach', 'apps', 'are', 'area', 'around', 'aroun
d us', 'around world', 'art', 'art room', 'art science', 'art supplie
s', 'artist', 'artistic', 'artists', 'arts', 'as', 'aspiring', 'at',
'at time', 'atpe', 'attention', 'audio', 'authors', 'autism', 'avid',
'awareness', 'away', 'awesome', 'baby', 'back', 'back basics', 'back
school', 'backpacks', 'bag', 'bags', 'balance', 'balancing', 'ball',
'ball chairs', 'balls', 'band', 'bands', 'based', 'based learning', '
basic', 'basic supplies', 'basics', 'basketball', 'be', 'be fun', 'be
an', 'beat', 'beautiful', 'because', 'become', 'becoming', 'begin', '
beginning', 'behavior', 'being', 'best', 'better', 'better readers',
'beyond', 'big', 'big books', 'bilingual', 'binders', 'bins', 'biolog
y', 'blast', 'blended', 'blended learning', 'blocks', 'board', 'board
s', 'bodies', 'bodies active', 'body', 'boogie', 'book', 'book bins',
'book club', 'book clubs', 'books', 'books are', 'books books', 'book
s classroom', 'books for', 'books more', 'books our', 'books we', 'bo
ost', 'bots', 'bounce', 'bouncing', 'bouncy', 'bouncy bands', 'box',
'boxes', 'boys', 'brain', 'brain power', 'brains', 'break', 'breaking
', 'breakout', 'bridge', 'bridging', 'bright', 'brighter', 'brighter
future', 'bring', 'bringing', 'budding', 'build', 'build our', 'build
ing', 'building blocks', 'building community', 'busy', 'but', 'by', '
ca', 'ca not', 'calculating', 'calculators', 'calling', 'calm', 'calm
ing', 'camera', 'cameras', 'can', 'can be', 'can do', 'can hear', 'ca
n learn', 'can read', 'can see', 'can we', 'can you', 'capture', 'cap
turing', 'care', 'career', 'carpet', 'carpet ride', 'cart', 'case', '
cases', 'celebrate', 'celebration', 'center', 'centered', 'centered c
lassroom', 'centers', 'century', 'century classroom', 'century learne
rs', 'century learning', 'century skills', 'century technology', 'cha
ir', 'chairs', 'challenge', 'change', 'changing', 'chapter', 'chapter
books', 'character', 'characters', 'charge', 'charged', 'charging', '
charts', 'check', 'chemistry', 'chess', 'child', 'children', 'choice
', 'choices', 'choose', 'chrome', 'chrome books', 'chromebook', 'chro
mebooks', 'chromebooks classroom', 'chromebooks needed', 'circle', 'c
ircles', 'circuits', 'citizens', 'city', 'class', 'class needs', 'cla
ssroom', 'classroom carpet', 'classroom community', 'classroom librar
y', 'classroom rug', 'classroom seating', 'classroom supplies', 'clas
sroom technology', 'classrooms', 'clay', 'clean', 'clearly', 'click',
'club', 'clubs', 'code', 'coding', 'coffee', 'collaborate', 'collabor
ation', 'collaborative', 'college', 'color', 'color our', 'color prin
ter', 'colorful', 'colors', 'come', 'come alive', 'come life', 'comes
', 'comfort', 'comfortable', 'comfy', 'comfy cozy', 'comfy reading',
```

```
'comic', 'comic books', 'coming', 'communicate', 'communication', 'co
mmunity', 'comprehension', 'computer', 'computer programming', 'compu
ter science', 'computers', 'concentration', 'confidence', 'connect',
'connecting', 'connections', 'cooking', 'cool', 'core', 'corner', 'co
uld', 'count', 'counts', 'cozy', 'cozy reading', 'crazy', 'create', '
creating', 'creation', 'creative', 'creative minds', 'creativity', 'c
ritical', 'critical thinking', 'cultural', 'culture', 'curiosity', 'c
urious', 'current', 'current events', 'curriculum', 'cut', 'cycle', '
daily', 'dance', 'dash', 'dash dot', 'data', 'day', 'day keeps', 'day
s', 'deserve', 'deserving', 'design', 'desk', 'desks', 'develop', 'de
veloping', 'development', 'did', 'difference', 'different', 'differen
tiated', 'digital', 'discover', 'discovering', 'discovery', 'diverse
', 'diversity', 'do', 'do it', 'do not', 'do you', 'document', 'docum
ent camera', 'does', 'does not', 'doing', 'dot', 'dots', 'down', 'dra
matic', 'dramatic play', 'drawing', 'dream', 'dreams', 'drone', 'drum
ming', 'drums', 'dry', 'dry erase', 'during', 'eager', 'ear', 'early
', 'ears', 'earth', 'easel', 'easy', 'eat', 'eating', 'economics', 'e
d', 'edu', 'education', 'education class', 'education students', 'edu
cational', 'effective', 'ela', 'electronic', 'elementary', 'emotional
', 'empowering', 'empowering students', 'encouraging', 'energetic', '
energy', 'engage', 'engaged', 'engagement', 'engaging', 'engaging stu
dents', 'engineering', 'engineers', 'english', 'english language', 'e
nhance', 'enhance learning', 'enhancing', 'enrichment', 'enthusiastic
', 'environment', 'environmental', 'equals', 'equipment', 'erase', 'e
rase boards', 'escape', 'esl', 'essential', 'essentials', 'even', 'ev
ents', 'every', 'every student', 'everyday', 'everyone', 'everything
', 'everywhere', 'excel', 'excellent', 'exceptional', 'excited', 'exc
itement', 'exciting', 'exercise', 'expand', 'expanding', 'experience
', 'experiences', 'exploration', 'explore', 'explorers', 'exploring',
'expression', 'extra', 'extra extra', 'extra read', 'eye', 'eyes', 'f
abulous', 'fair', 'fairy', 'fall', 'families', 'family', 'fantastic',
'favorite', 'fear', 'feel', 'fiction', 'fidget', 'fidgeting', 'fidget
s', 'field', 'fifth', 'fifth grade', 'fill', 'financial', 'financial
literacy', 'find', 'find out', 'finding', 'fine', 'fine motor', 'fing
ers', 'fingertips', 'fire', 'fire up', 'fired', 'fired up', 'fires',
'first', 'first grade', 'first graders', 'firsties', 'fit', 'fitness
', 'five', 'flex', 'flexible', 'flexible classroom', 'flexible learne
rs', 'flexible learning', 'flexible minds', 'flexible seating', 'floo
d', 'floor', 'fluency', 'fluent', 'focus', 'focused', 'focusing', 'fo
lders', 'food', 'football', 'for', 'for all', 'for everyone', 'for ki
ds', 'for learning', 'for love', 'for our', 'for reading', 'for stude
nts', 'for success', 'for the', 'forward', 'fostering', 'foundation',
'fourth', 'fourth grade', 'fourth graders', 'free', 'freedom', 'frenc
h', 'fresh', 'friendly', 'friends', 'from', 'fuel', 'full', 'full ste
am', 'fun', 'fun learning', 'fun with', 'furniture', 'future', 'futur
e engineers', 'future leaders', 'future scientists', 'futures', 'gain
ing', 'galore', 'game', 'games', 'gap', 'garden', 'gardening', 'gathe
r', 'generation', 'genius', 'geometry', 'get', 'get fit', 'get moving
', 'get organized', 'get your', 'getting', 'getting comfy', 'getting
our', 'girls', 'give', 'giving', 'global', 'glue', 'go', 'goal', 'goa
ls', 'goes', 'going', 'gone', 'good', 'good book', 'google', 'google
classroom', 'got', 'grade', 'grade classroom', 'graders', 'graders ne
ed', 'graphic', 'graphic novels', 'graphing', 'great', 'great books',
'greatness', 'green', 'grooving', 'group', 'groups', 'grow', 'growing
', 'growing minds', 'growth', 'growth mindset', 'guided', 'guided rea
ding', 'habits', 'hand', 'hands', 'hands learning', 'hands math', 'ha
```

```
nds on', 'happy', 'hard', 'has', 'have', 'have seat', 'having', 'havi
ng fun', 'headphones', 'health', 'healthier', 'healthy', 'healthy bod
ies', 'healthy minds', 'healthy snacks', 'hear', 'hear me', 'hearing
', 'heart', 'hearts', 'help', 'help keep', 'help make', 'help me', 'h
elp my', 'help our', 'help students', 'help us', 'help we', 'helping
', 'helping students', 'helps', 'helps us', 'here', 'here we', 'heroe
s', 'high', 'high interest', 'high school', 'higher', 'history', 'hok
ki', 'hokki stools', 'holocaust', 'home', 'hope', 'hot', 'house', 'ho
w', 'human', 'hungry', 'hygiene', 'ican', 'ideas', 'if', 'if you', 'i
gnite', 'igniting', 'ii', 'ilearn', 'ilearn ipads', 'imagination', 'i
maginations', 'imagine', 'important', 'improve', 'improving', 'in', '
in need', 'in our', 'in the', 'income', 'increase', 'increasing', 'in
dependence', 'independent', 'independent reading', 'individual', 'ind
ividualized', 'individualized learning', 'indoor', 'indoor recess', '
information', 'ing', 'ink', 'inner', 'innovation', 'innovative', 'inn
ovators', 'inquiry', 'inside', 'inspiration', 'inspire', 'inspired',
'inspiring', 'instruction', 'instrument', 'instruments', 'integrating
', 'integration', 'interactive', 'interactive learning', 'interactive
notebooks', 'interest', 'interest books', 'intervention', 'into', 'in
to learning', 'inventors', 'ipad', 'ipad mini', 'ipad minis', 'ipads
', 'is', 'is fun', 'it', 'it all', 'it move', 'it out', 'it time', 'i
t up', 'items', 'its', 'journalism', 'journals', 'journey', 'jump', '
just', 'just right', 'keep', 'keep calm', 'keep our', 'keep us', 'kee
ping', 'keeps', 'key', 'key success', 'keyboards', 'kid', 'kid inspir
ed', 'kiddos', 'kids', 'kids need', 'kind', 'kinder', 'kindergarten',
'kindergarten classroom', 'kindergarten students', 'kindergarteners',
'kindergartners', 'kinders', 'kindle', 'kindle fire', 'kindle fires',
'kindles', 'kinesthetic', 'kit', 'kitchen', 'kits', 'know', 'knowledg
e', 'lab', 'labs', 'language', 'language arts', 'language learners',
'laptop', 'laptops', 'lead', 'leader', 'leaders', 'leads', 'learn', '
learn about', 'learn read', 'learn with', 'learner', 'learners', 'lea
rners need', 'learning', 'learning about', 'learning all', 'learning
environment', 'learning fun', 'learning is', 'learning read', 'learni
ng technology', 'learning through', 'learning using', 'learning with
', 'lego', 'legos', 'lesson', 'lessons', 'let', 'let get', 'let go',
'let learn', 'let make', 'let me', 'let play', 'let read', 'let us',
'lets', 'level', 'leveled', 'leveled library', 'library', 'library ne
eds', 'life', 'life skills', 'lifelong', 'lifelong readers', 'lifetim
e', 'light', 'lighting', 'lights', 'lights camera', 'like', 'like mov
e', 'listen', 'listen learn', 'listen up', 'listening', 'listening ce
nter', 'listening learning', 'literacy', 'literacy centers', 'literar
y', 'literature', 'literature circles', 'little', 'little learners',
'live', 'lives', 'living', 'long', 'look', 'looking', 'lost', 'love',
'love learn', 'love learning', 'love literacy', 'love read', 'love re
ading', 'loving', 'low', 'low income', 'macbook', 'made', 'madness',
'magazine', 'magazines', 'magic', 'magic carpet', 'magical', 'magneti
c', 'magnificent', 'make', 'make it', 'make learning', 'make our', 'm
ake reading', 'make us', 'makeover', 'maker', 'makers', 'makerspace',
'makes', 'makes learning', 'making', 'making learning', 'making math
', 'making reading', 'mania', 'manipulatives', 'many', 'markers', 'ma
rvelous', 'masters', 'materials', 'math', 'math centers', 'math class
', 'math classroom', 'math fun', 'math games', 'math manipulatives',
'math materials', 'math science', 'math skills', 'mathematicians', 'm
athematics', 'mats', 'matter', 'matters', 'may', 'me', 'me now', 'mea
ningful', 'means', 'media', 'media literacy', 'meet', 'meeting', 'mee
ts', 'memories', 'mentor', 'middle', 'middle school', 'mind', 'mindfu
```

```
lness', 'minds', 'mindset', 'mini', 'minis', 'mobile', 'modeling', 'm
odels', 'modern', 'money', 'more', 'more books', 'more than', 'mornin
g', 'motion', 'motivated', 'motivating', 'motivation', 'motor', 'moto
r skills', 'move', 'move it', 'move learn', 'move move', 'movement',
'movin', 'moving', 'moving grooving', 'moving learning', 'mr', 'mrs',
'ms', 'much', 'multi', 'muscles', 'music', 'musical', 'must', 'my', '
my classroom', 'my kids', 'my students', 'necessities', 'need', 'need
books', 'need chromebooks', 'need ipads', 'need more', 'need new', 'n
eed supplies', 'need technology', 'needed', 'needs', 'needs students
', 'never', 'new', 'new books', 'new classroom', 'new school', 'new t
eacher', 'new year', 'news', 'next', 'next generation', 'next level',
'nice', 'no', 'no more', 'noise', 'non', 'non fiction', 'nonfiction',
'nook', 'not', 'not just', 'notebooks', 'novel', 'novels', 'now', 'nu
mber', 'numbers', 'nutrition', 'of', 'off', 'oh', 'oh my', 'oh places
', 'oh the', 'old', 'on', 'on learning', 'on math', 'on target', 'on
the', 'one', 'one book', 'online', 'only', 'open', 'opening', 'operat
ion', 'opportunities', 'opportunity', 'optimal', 'optimal learning',
'options', 'organization', 'organization key', 'organize', 'organize
our', 'organized', 'organizing', 'osmo', 'our', 'our bodies', 'our bo
oks', 'our brains', 'our class', 'our classroom', 'our future', 'our
hands', 'our learning', 'our library', 'our minds', 'our own', 'our r
eading', 'our students', 'our technology', 'our way', 'our wiggles',
'our world', 'ourselves', 'out', 'outdoor', 'outside', 'over', 'owl',
'own', 'ozobots', 'pad', 'pads', 'page', 'paint', 'painting', 'pants
', 'paper', 'parents', 'part', 'part ii', 'pass', 'passion', 'past',
'path', 'pe', 'peace', 'pencil', 'pencils', 'pens', 'people', 'perfec
t', 'personal', 'personalized', 'personalized learning', 'phonics', '
physical', 'physical education', 'physical fitness', 'physics', 'pick
', 'picture', 'pictures', 'place', 'place sit', 'places', 'plants', '
play', 'playground', 'playing', 'please', 'please help', 'poetry', 'p
op', 'portfolios', 'positive', 'possibilities', 'possible', 'power',
'practice', 'pre', 'prek', 'prepare', 'prepared', 'preparing', 'presc
hool', 'preschoolers', 'pretty', 'print', 'printer', 'printing', 'pro
', 'problem', 'problem solving', 'productive', 'program', 'programmin
g', 'project', 'project based', 'projecting', 'projector', 'projects
', 'promote', 'promoting', 'protect', 'provide', 'providing', 'purpos
e', 'put', 'putting', 'quality', 'quest', 'quiet', 'reach', 'reaching
', 'read', 'read all', 'read aloud', 'read alouds', 'read more', 'rea
d read', 'read succeed', 'read write', 'reader', 'reader tomorrow', '
readers', 'readers need', 'readers writers', 'reading', 'reading cent
er', 'reading corner', 'reading fun', 'reading is', 'reading math', '
reading nook', 'reading success', 'reading writing', 'reads', 'ready
', 'ready learn', 'ready read', 'ready set', 'real', 'real world', 'r
eality', 'really', 'recess', 'reluctant', 'reluctant readers', 'remem
ber', 'replace', 'rescue', 'research', 'resource', 'resources', 'rewa
rds', 'rhythm', 'rich', 'ride', 'right', 'road', 'robot', 'robotics',
'robots', 'rock', 'rocks', 'roll', 'rolling', 'room', 'round', 'rug',
'run', 'running', 'rural', 'sacks', 'safe', 'safety', 'save', 'savvy
', 'say', 'scholars', 'scholastic', 'scholastic magazines', 'scholast
ic news', 'school', 'school students', 'school supplies', 'school yea
r', 'science', 'science classroom', 'science math', 'science technolo
gy', 'scientific', 'scientist', 'scientists', 'screen', 'seat', 'seat
sacks', 'seating', 'seating 21st', 'seating active', 'seating classro
om', 'seating flexible', 'seating for', 'seating options', 'seats', '
second', 'second grade', 'second graders', 'see', 'see it', 'seeing',
'seek', 'seeking', 'self', 'sensational', 'sense', 'senses', 'sensory
```

', 'sensory needs', 'series', 'set', 'sets', 'setting', 'shakespeare
', 'share', 'sharing', 'sharp', 'shine', 'should', 'show', 'sight', '
signing', 'simple', 'sims', 'sing', 'sit', 'sit still', 'sitting', 's
kills', 'small', 'small group', 'small groups', 'smart', 'smile', 'sn
ack', 'snacks', 'snap', 'so', 'soar', 'soaring', 'soccer', 'social',
'social emotional', 'social skills', 'social studies', 'solve', 'solv
ing', 'some', 'something', 'sound', 'sounds', 'space', 'spaces', 'spa
nish', 'spark', 'speak', 'special', 'special education', 'special nee
ds', 'spectacular', 'speech', 'speech therapy', 'sports', 'spot', 'sp
ring', 'stability', 'stability balls', 'stand', 'stand up', 'standing
', 'star', 'stars', 'start', 'starting', 'starts', 'station', 'statio
ns', 'stay', 'staying', 'staying active', 'steam', 'steam ahead', 'st
eaming', 'stem', 'stem activities', 'stem learning', 'stem projects',
'step', 'stepping', 'still', 'stock', 'stools', 'stop', 'storage', 's
tore', 'stories', 'story', 'stretch', 'striving', 'strong', 'struggli
ng', 'struggling readers', 'student', 'student centered', 'student en
gagement', 'student learning', 'student success', 'students', 'studen
ts autism', 'students learn', 'students need', 'students read', 'stud
ents through', 'students want', 'students with', 'studies', 'studio',
'study', 'stuff', 'style', 'succeed', 'success', 'successful', 'summe
r', 'super', 'super students', 'supplies', 'supplies for', 'supplies
needed', 'supplies success', 'supply', 'support', 'supporting', 'supp
orts', 'sweet', 'table', 'tables', 'tablet', 'tablets', 'take', 'take
seat', 'takes', 'taking', 'tales', 'talk', 'target', 'targeting', 'te
ach', 'teacher', 'teachers', 'teaching', 'teaching technology', 'team
', 'teamwork', 'tech', 'tech savvy', 'techies', 'techno', 'technology
', 'technology 21st', 'technology classroom', 'technology for', 'tech
nology future', 'technology help', 'technology needed', 'technology o
ur', 'techy', 'teens', 'terrific', 'testing', 'text', 'texts', 'than
', 'that', 'the', 'the art', 'the classroom', 'the future', 'the key
', 'the more', 'the new', 'the next', 'the places', 'the power', 'the
world', 'their', 'them', 'therapy', 'there', 'these', 'they', 'things
', 'think', 'thinkers', 'thinking', 'thinking through', 'third', 'thi
rd grade', 'third graders', 'this', 'those', 'thousand', 'thousand wo
rds', 'through', 'through art', 'through flexible', 'through literatu
re', 'through play', 'through reading', 'through technology', 'time',
'time for', 'time kids', 'times', 'tiny', 'title', 'to', 'to be', 'to
learn', 'to move', 'to read', 'to the', 'today', 'today reader', 'tog
ether', 'tomorrow', 'tomorrow leader', 'too', 'tool', 'tools', 'toon
', 'top', 'touch', 'toward', 'towards', 'toys', 'track', 'training',
'transforming', 'true', 'tubs', 'tune', 'turn', 'turning', 'tv', 'two
', 'ukulele', 'ukuleles', 'understanding', 'unit', 'up', 'up learning
', 'up our', 'up with', 'upon', 'urban', 'us', 'us become', 'us get',
'us grow', 'us learn', 'us read', 'us stay', 'us with', 'use', 'using
', 'using technology', 'variety', 'via', 'video', 'videos', 'virtual
', 'virtual reality', 'visual', 'visualize', 'vocabulary', 'voice', '
voices', 'volleyball', 'walk', 'wall', 'walls', 'want', 'want to', 'w
anted', 'warm', 'watch', 'water', 'way', 'way success', 'way through
', 'way to', 'ways', 'we', 'we all', 'we are', 'we can', 'we come', '
we got', 'we have', 'we learn', 'we like', 'we love', 'we need', 'we
read', 'we want', 'we will', 'we work', 'welcome', 'well', 'wellness
', 'what', 'what we', 'when', 'where', 'while', 'while learning', 'wh
ile we', 'while you', 'white', 'white boards', 'whiteboard', 'who', '
whole', 'why', 'wiggle', 'wiggle learn', 'wiggle while', 'wiggle wigg
le', 'wiggle wobble', 'wiggle work', 'wiggles', 'wiggles out', 'wiggl
ing', 'wiggly', 'wild', 'will', 'will help', 'win', 'winning', 'winte

```
r', 'wireless', 'with', 'with chromebooks', 'with flexible', 'with go
od', 'with ipads', 'with new', 'with our', 'with reading', 'with tech
nology', 'with the', 'without', 'wizards', 'wobble', 'wobble chairs',
'wobble learn', 'wobble while', 'wobbling', 'wobbly', 'wonder', 'wond
erful', 'word', 'word work', 'words', 'work', 'work part', 'working',
'workshop', 'world', 'world around', 'world through', 'world with', '
worms', 'worth', 'would', 'write', 'writers', 'writing', 'year', 'yea
rbook', 'yes', 'yoga', 'you', 'you hear', 'you know', 'you read', 'yo
```

#### 1.5.2.2 tfidf

In [166]:
```
def tfidf_Vectorizer(sFeature, project_data_fitting,project_data_trans
form):
    from sklearn.feature_extraction.text import TfidfVectorizer
    vectorizer_tfidf_feature = TfidfVectorizer(min_df=10, ngram_range=
(1, 2),max_features = 5000)
    vectorizer_tfidf_feature.fit(project_data_train[sFeature])      #F
itting has to be on Train data

    tfidf_vect = vectorizer_tfidf_feature.transform(project_data_trans
form[sFeature].values)
    return(tfidf_vect)
```

In [167]:
```
train_essay_tfidf = tfidf_Vectorizer('preprocessed_essays',  project_d
ata_train, project_data_train)
test_essay_tfidf = tfidf_Vectorizer('preprocessed_essays',  project_da
ta_train, project_data_test)

print("Shape of train data matrix after one hot encoding ",train_essay
_tfidf.shape)
print("Shape of test data matrix after one hot encoding ",test_essay_t
fidf.shape)
```

```
Shape of train data matrix after one hot encoding  (20100, 5000)
Shape of test data matrix after one hot encoding  (9900, 5000)
```

In [168]:
```
train_title_tfidf = tfidf_Vectorizer('preprocessed_titles',  project_d
ata_train, project_data_train)
test_title_tfidf = tfidf_Vectorizer('preprocessed_titles',  project_da
ta_train, project_data_test)

print("Shape of train data matrix after one hot encoding ",train_title
_tfidf.shape)
print("Shape of test data matrix after one hot encoding ",test_title_t
fidf.shape)
```

```
Shape of train data matrix after one hot encoding  (20100, 1668)
Shape of test data matrix after one hot encoding  (9900, 1668)
```

#### 1.5.2.3 Using Pretrained Models: W2V

In [169]:
```
# stronging variables into pickle files python: http://www.jessicayun
g.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 [170]:
```
# average Word2Vec
# compute average word2vec for each review.
train_No_ofWords_essays=[];
train_avg_w2v_essays = []; # the avg-w2v for each sentence/review is s
tored in this list
for sentence in tqdm(project_data_train['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/r
eview
    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
    train_avg_w2v_essays.append(vector)
    train_No_ofWords_essays.append(len(sentence.split()))

print(len(train_avg_w2v_essays))
print(len(train_avg_w2v_essays[0]))
```

```
100%|██████████| 20100/20100 [00:17<00:00, 1139.64it/s]

20100
300
```

In [171]:
```
len(train_No_ofWords_essays)
```

Out[171]: 20100

In [172]:
```
# compute average word2vec for each review.
test_No_ofWords_essays=[];
test_avg_w2v_essays = []; # the avg-w2v for each sentence/review is st
ored in this list
for sentence in tqdm(project_data_test['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/r
eview
    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
    test_avg_w2v_essays.append(vector)
    test_No_ofWords_essays.append(len(sentence.split()))

print(len(test_avg_w2v_essays))
print(len(test_avg_w2v_essays[0]))
print(len(test_No_ofWords_essays))
```

```
100%|██████████| 9900/9900 [00:07<00:00, 1303.01it/s]

9900
300
9900
```

In [173]:
```
# average Word2Vec
# compute average word2vec for each review.
train_No_ofWords_titles=[];
train_avg_w2v_titles = []; # the avg-w2v for each sentence/review is s
tored in this list
for sentence in tqdm(project_data_train['preprocessed_titles']): # for
each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/r
eview
    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
    train_avg_w2v_titles.append(vector)
    train_No_ofWords_titles.append(len(sentence.split()))

print(len(train_avg_w2v_titles))
print(len(train_avg_w2v_titles[0]))
print(len(train_No_ofWords_titles))
```

```
100%|██████████| 20100/20100 [00:01<00:00, 17328.41it/s]

20100
300
20100
```

In [174]:
```
# average Word2Vec
# compute average word2vec for each review.
test_No_ofWords_titles=[];
test_avg_w2v_titles = []; # the avg-w2v for each sentence/review is st
ored in this list
for sentence in tqdm(project_data_test['preprocessed_titles']): # for
each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/r
eview
    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
    test_avg_w2v_titles.append(vector)
    test_No_ofWords_titles.append(len(sentence.split()))

print(len(test_avg_w2v_titles))
print(len(test_avg_w2v_titles[0]))
print(len(test_No_ofWords_titles))
```

```
100%|██████████| 9900/9900 [00:00<00:00, 17704.31it/s]

9900
300
9900
```

#### 1.5.2.3 Using Pretrained Models: TFIDF weighted W2V

In [175]:
```
# Similarly you can vectorize for title also
tfidf_model = TfidfVectorizer()
tfidf_model.fit(project_data_train['preprocessed_titles'])
# 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_mode
l.idf_)))
tfidf_words = set(tfidf_model.get_feature_names())
```

In [176]:
```
# average Word2Vec
# compute average word2vec for each review.
train_tfidf_w2v_titles = []; # the avg-w2v for each sentence/review is
stored in this list
for sentence in tqdm(project_data_train['preprocessed_titles']): # for
each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the senten
ce/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and
the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(senten
ce.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
    train_tfidf_w2v_titles.append(vector)

print(len(train_tfidf_w2v_titles))
print(len(train_tfidf_w2v_titles[0]))
```

```
100%|██████████| 20100/20100 [00:02<00:00, 7178.96it/s]

20100
300
```

In [177]:
```
# compute average word2vec for each review.
test_tfidf_w2v_titles = []; # the avg-w2v for each sentence/review is
stored in this list
for sentence in tqdm(project_data_test['preprocessed_titles']): # for
each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the senten
ce/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and
the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(senten
ce.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
    test_tfidf_w2v_titles.append(vector)

print(len(test_tfidf_w2v_titles))
print(len(test_tfidf_w2v_titles[0]))
```

```
100%|██████████| 9900/9900 [00:01<00:00, 8143.33it/s]

9900
300
```

In [178]:
```
# Similarly you can vectorize for title also
tfidf_model = TfidfVectorizer()
tfidf_model.fit(project_data_train['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_mode
l.idf_)))
tfidf_words = set(tfidf_model.get_feature_names())
```

In [179]:
```
# average Word2Vec
# compute average word2vec for each review.
train_tfidf_w2v_essays = []; # the avg-w2v for each sentence/review is
stored in this list
for sentence in tqdm(project_data_train['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 senten
ce/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and
the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(senten
ce.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
    train_tfidf_w2v_essays.append(vector)

print(len(train_tfidf_w2v_essays))
print(len(train_tfidf_w2v_essays[0]))
```

```
100%|██████████| 20100/20100 [01:57<00:00, 170.69it/s]

20100
300
```

In [180]:
```
# compute average word2vec for each review.
test_tfidf_w2v_essays = []; # the avg-w2v for each sentence/review is
stored in this list
for sentence in tqdm(project_data_test['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 senten
ce/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and
the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(senten
ce.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
    test_tfidf_w2v_essays.append(vector)

print(len(test_tfidf_w2v_essays))
print(len(test_tfidf_w2v_essays[0]))
```

```
100%|██████████| 9900/9900 [00:56<00:00, 174.12it/s]

9900
300
```

### 1.5.3 Vectorizing Numerical features

In [181]:
```
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity
':'sum'}).reset_index()

project_data_train = pd.merge(project_data_train, price_data, on='id',
how='left')
project_data_test = pd.merge(project_data_test, price_data, on='id', h
ow='left')
```

In [182]:
```
from sklearn.preprocessing import Normalizer
# normalizer.fit(X_train['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.

normalizer = Normalizer()
normalizer.fit(project_data_train['price'].values.reshape(-1,1))


price_normalized_train = normalizer.transform(project_data_train['pric
e'].values.reshape(-1, 1))
price_normalized_test = normalizer.transform(project_data_test['price
'].values.reshape(-1, 1))


print('After normalization')
print(price_normalized_train.shape)
print(price_normalized_test.shape)
```

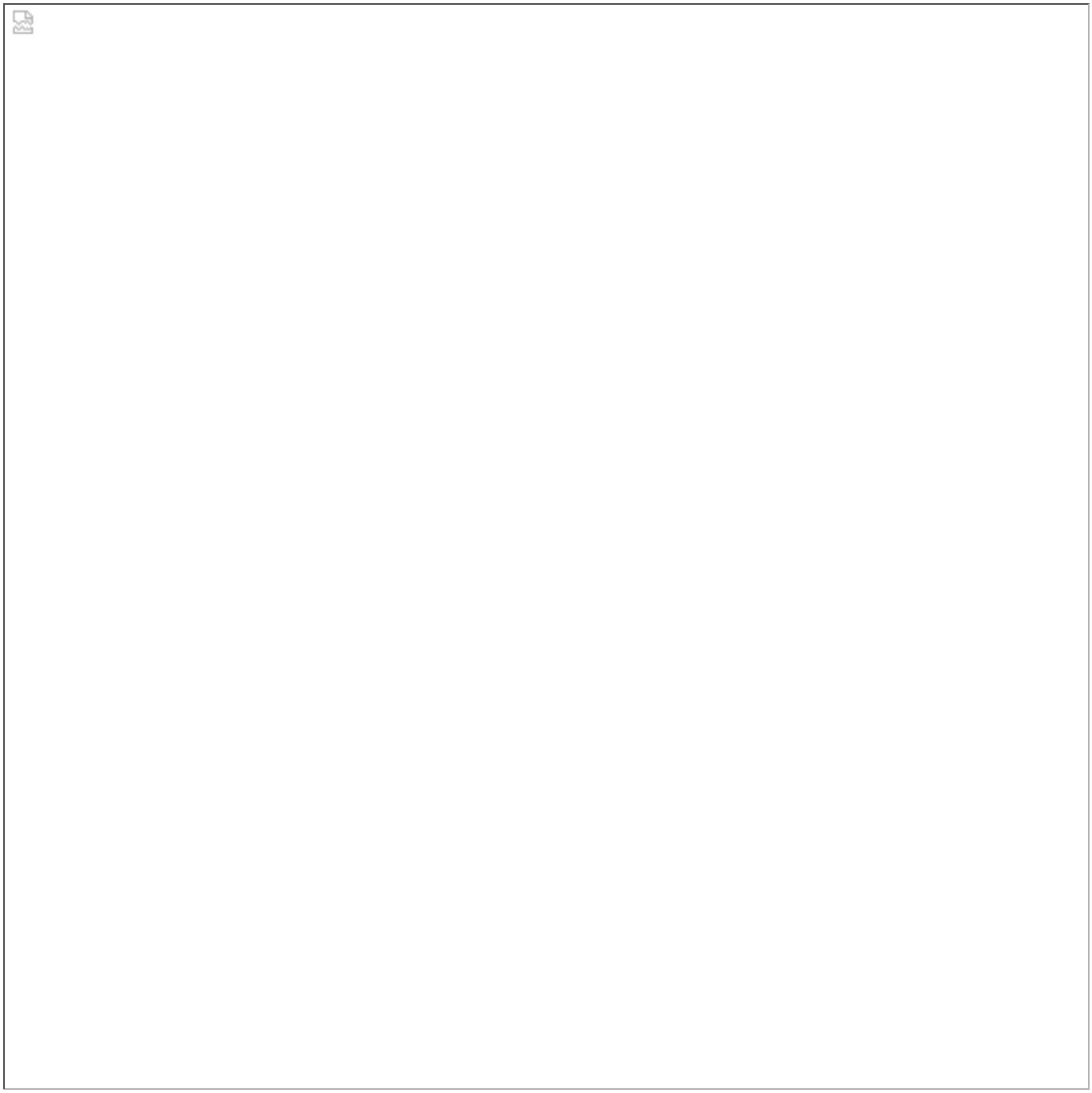
```
After normalization
(20100, 1)
(9900, 1)
```

In [183]:
```
normalizer = Normalizer()
normalizer.fit(project_data_train['teacher_number_of_previously_posted
_projects'].values.reshape(-1,1))


# Now standardize the data with above maen and variance.
previously_posted_projects_normalized_train = normalizer.transform(pro
ject_data_train['teacher_number_of_previously_posted_projects'].value
s.reshape(-1, 1))
previously_posted_projects_normalized_test = normalizer.transform(proj
ect_data_test['teacher_number_of_previously_posted_projects'].values.r
eshape(-1, 1))


print('After normalization')
print(previously_posted_projects_normalized_train.shape)
print(previously_posted_projects_normalized_test.shape)
```

```
After normalization
(20100, 1)
(9900, 1)
```

# Assignment 9: RF and GBDT

**Response Coding: Example**



The response tabel is built only on train dataset. For a category which is not there in train data and present in test data, we will encode them with default values Ex: in our test data if have State: D then we encode it as [0.5, 0.5]

1. **Apply both Random Forrest and GBDT on these feature sets**

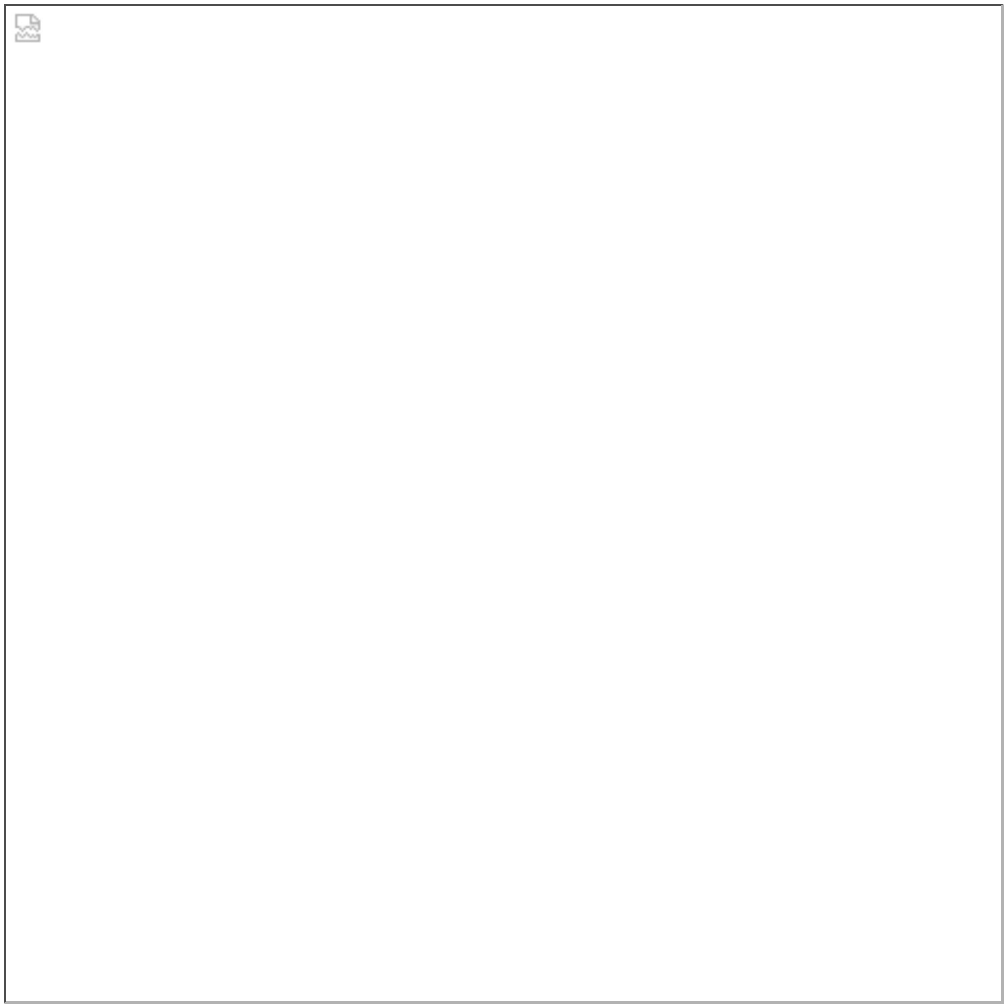
    - Set 1: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(BOW) + preprocessed_eassay (BOW)
    - Set 2: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF)+ preprocessed_eassay (TFIDF)
    - Set 3: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(AVG W2V)+ preprocessed_eassay (AVG W2V). Here for this set take **20K** datapoints only.
    - Set 4: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF W2V)+ preprocessed_eassay (TFIDF W2V). Here for this set take **20K** datapoints only.

2. **The hyper paramter tuning (Consider any two hyper parameters preferably n_estimators, max_depth)**

    - Consider the following range for hyperparameters **n_estimators** = [10, 50, 100, 150, 200, 300, 500, 1000], **max_depth** = [2, 3, 4, 5, 6, 7, 8, 9, 10]
    - Find the best hyper parameter which will give the maximum AUC (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/receiver-operating-characteristic-curve-roc-curve-and-auc-1/) value
    - Find the best hyper paramter using simple cross validation data
    - You can write your own for loops to do this task

3. **Representation of results**

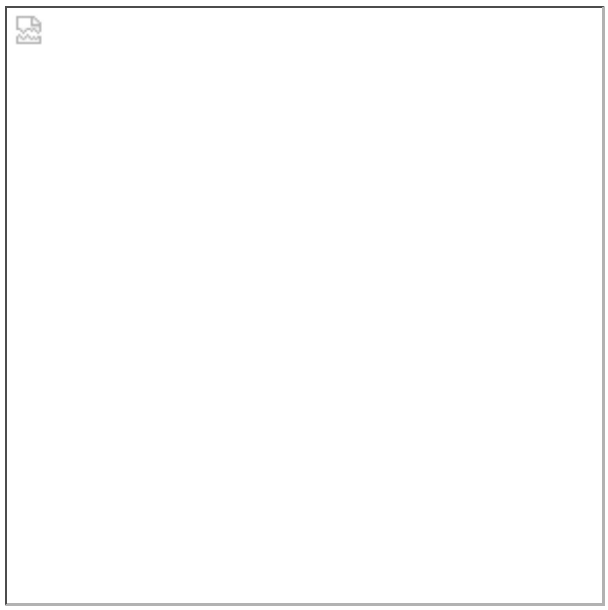
    - You need to plot the performance of model both on train data and cross validation data for each hyper parameter, like shown in the figure



with X-axis as **n_estimators**, Y-axis as **max_depth**, and Z-axis as **AUC Score** , we have given the notebook which explains how to plot this 3d plot, you can find it in the same drive *3d_scatter_plot.ipynb*

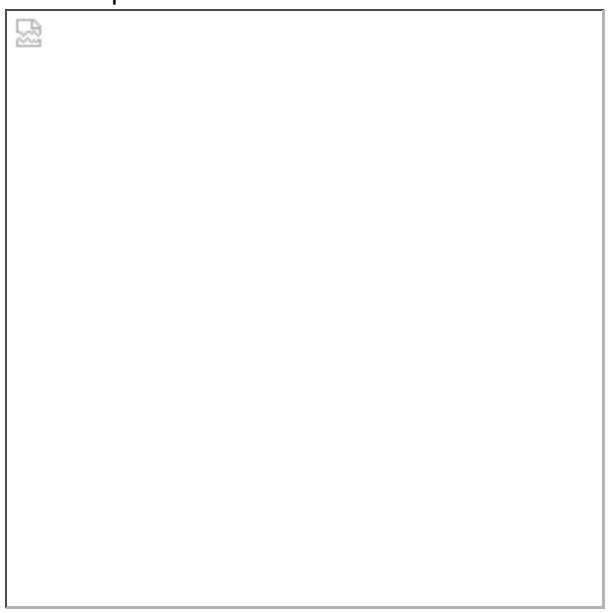
# or

- You need to plot the performance of model both on train data and cross validation data for each hyper parameter, like shown in the figure



seaborn heat maps (https://seaborn.pydata.org/generated/seaborn.heatmap.html) with rows as **n_estimators**, columns as **max_depth**, and values inside the cell representing **AUC Score**

- You can choose either of the plotting techniques: 3d plot or heat map
- Once after you found the best hyper parameter, you need to train your model with it, and find the AUC on test data and plot the ROC curve on both train and test.



- Along with plotting ROC curve, you need to print the confusion matrix (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/confusion-matrix-tpr-fpr-fnr-tnr-1/) with predicted and original labels of test data points

**Note: Data Leakage**

1. There will be an issue of data-leakage if you vectorize the entire data and then split it into train/cv /test.
2. To avoid the issue of data-leakage, make sure to split your data first and then vectorize it.
3. While vectorizing your data, apply the method fit_transform() on you train data, and apply the method transform() on cv/test data.
4. For more details please go through this link. (https://soundcloud.com/applied-ai-course/leakage-bow-and-tfidf)

# Set 1: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(BOW) + preprocessed_eassay (BOW)

In [57]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_essay_bow, train_title_bow, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe, test_essay_bow, test_title_bow,   previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(20100, 6696) (20100,)
(9900, 6696) (9900,)
```

## Using GridSearchCV

In [58]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
import seaborn as sea
```

```
C:\Users\venka\Anaconda3\lib\site-packages\sklearn\ensemble\weight_bo
osting.py:29: DeprecationWarning:

numpy.core.umath_tests is an internal NumPy module and should not be
imported. It will be removed in a future NumPy release.
```

In [59]:
```
RF = RandomForestClassifier(class_weight = 'balanced')

tree_para = {'max_depth':[1, 5, 10, 50],'min_samples_split': [5, 10, 10
0, 500]}

clf =  GridSearchCV(RF, tree_para, cv=3)

clf.fit(X_train, y_train)
```

Out[59]:
```
GridSearchCV(cv=3, error_score='raise',
       estimator=RandomForestClassifier(bootstrap=True, class_weight=
'balanced',
            criterion='gini', max_depth=None, max_features='auto',
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=10, n_jobs=1, oob_score=False, random_state=
None,
            verbose=0, warm_start=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'max_depth': [1, 5, 10, 50], 'min_samples_split':
[5, 10, 100, 500]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn
',
       scoring=None, verbose=0)
```

In [60]:
```
clf.get_params().keys()
```

Out[60]:
```
dict_keys(['cv', 'error_score', 'estimator__bootstrap', 'estimator__c
lass_weight', 'estimator__criterion', 'estimator__max_depth', 'estima
tor__max_features', 'estimator__max_leaf_nodes', 'estimator__min_impu
rity_decrease', 'estimator__min_impurity_split', 'estimator__min_samp
les_leaf', 'estimator__min_samples_split', 'estimator__min_weight_fra
ction_leaf', 'estimator__n_estimators', 'estimator__n_jobs', 'estimat
or__oob_score', 'estimator__random_state', 'estimator__verbose', 'est
imator__warm_start', 'estimator', 'fit_params', 'iid', 'n_jobs', 'par
am_grid', 'pre_dispatch', 'refit', 'return_train_score', 'scoring', '
verbose'])
```

In [61]:
```
clf.best_params_
```

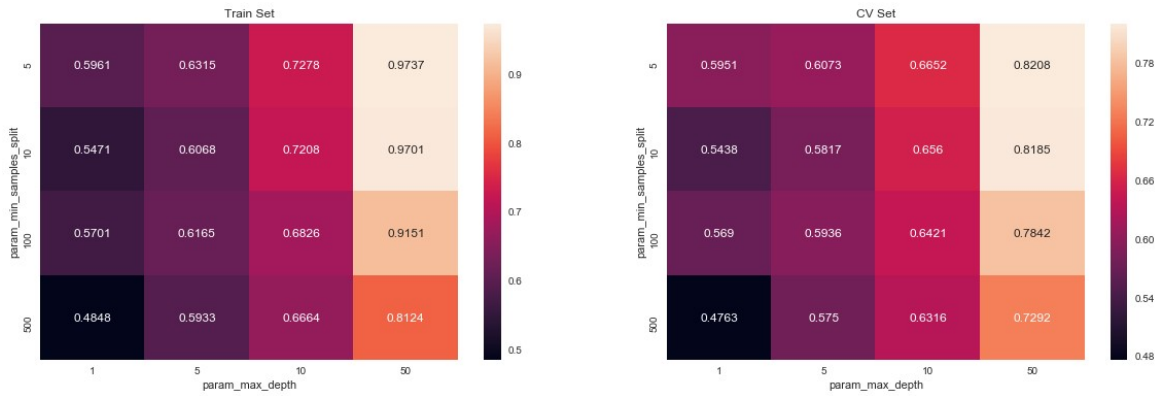
Out[61]:
```
{'max_depth': 50, 'min_samples_split': 5}
```

### Find best parameter using 'GridSearchCV'

In [62]:
```
max_d = clf.best_params_['max_depth']
min_samp_splt = clf.best_params_['min_samples_split']
```

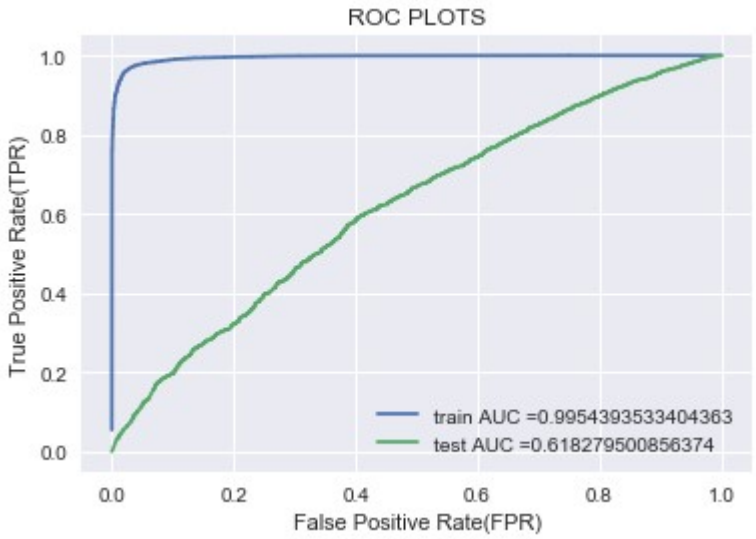
### Heat map

In [63]:
```
import seaborn as sns; sns.set()
max_scores1 = pd.DataFrame(clf.cv_results_).groupby(['param_min_samples
_split', 'param_max_depth']).max().unstack()[['mean_test_score','mean_t
rain_score']]

fig, ax = plt.subplots(1,2, figsize=(20,6))
sns.heatmap(max_scores1.mean_train_score, annot = True, fmt='.4g', ax=a
x[0])
sns.heatmap(max_scores1.mean_test_score, annot = True, fmt='.4g', ax=ax
[1])
ax[0].set_title('Train Set')
ax[1].set_title('CV Set')
plt.show()
```




In [64]:
```
def batch_predict(clf, data):
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probab
ility estimates of the positive class
    # not the predicted outputs

    y_data_pred = []
    tr_loop = data.shape[0] - data.shape[0]%1000
    # consider you X_tr shape is 49041, then your tr_loop will be 49041
- 49041%1000 = 49000
    # in this for loop we will iterate unti the last 1000 multiplier
    for i in range(0, tr_loop, 1000):
        y_data_pred.extend(clf.predict_proba(data[i:i+1000])[:,1])
    # we will be predicting for the last data points
    if data.shape[0]%1000 !=0:
        y_data_pred.extend(clf.predict_proba(data[tr_loop:])[:,1])

    return y_data_pred
```

In [65]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier

RF = RandomForestClassifier(max_depth = max_d, min_samples_split = min_samp_splt, class_weight='balanced')
RF.fit(X_train ,y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probabilit y estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(RF, X_train)#Return probability estimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(RF, X_test)#Return probability estimates f or the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```



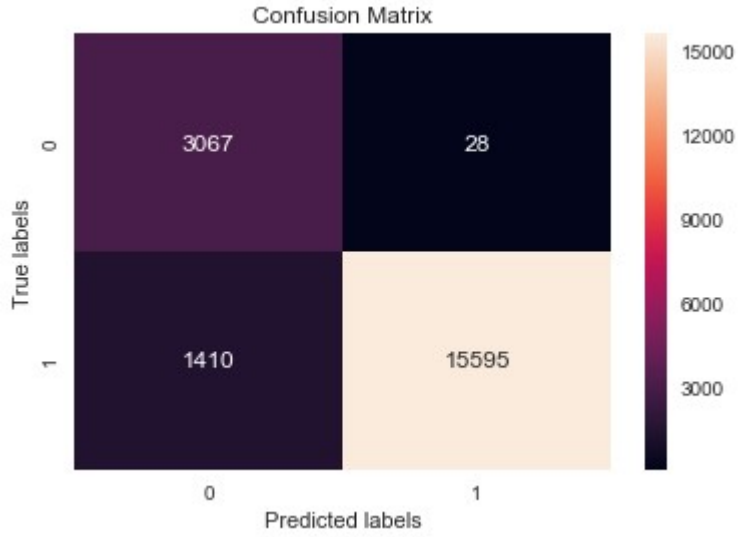

## Confusion Matrix of Train and Test Data

In [66]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is
very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for th
reshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

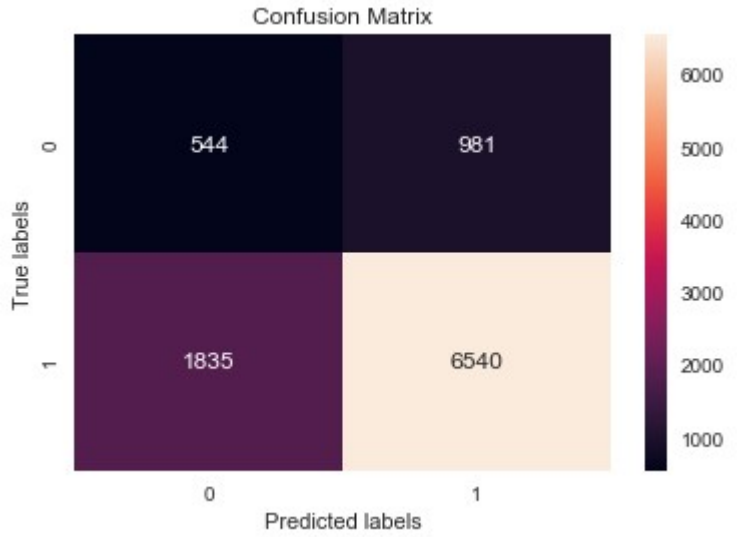
In [67]:
```
from sklearn.metrics import confusion_matrix
best_t = find_best_threshold(thresholds, train_fpr, train_tpr)
print("Train confusion matrix")
print(confusion_matrix(y_train, predict_with_best_t(y_train_pred, best_
t)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict_with_best_t(y_test_pred, best_
t)))
```

```
the maximum value of tpr*(1-fpr) 0.9401200350177346 for threshold 0.6
46
Train confusion matrix
[[ 3067    28]
 [ 1410 15595]]
Test confusion matrix
[[ 544  981]
 [1835 6540]]
```

In [68]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```




In [69]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred, b
est_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



## Set 2: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF)+ preprocessed_eassay (TFIDF)

In [70]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_essay_tfidf, train_title_tfidf, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe, test_essay_tfidf, test_title_tfidf,   previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(20100, 6696) (20100,)
(9900, 6696) (9900,)
```

### Using GridSearchCV

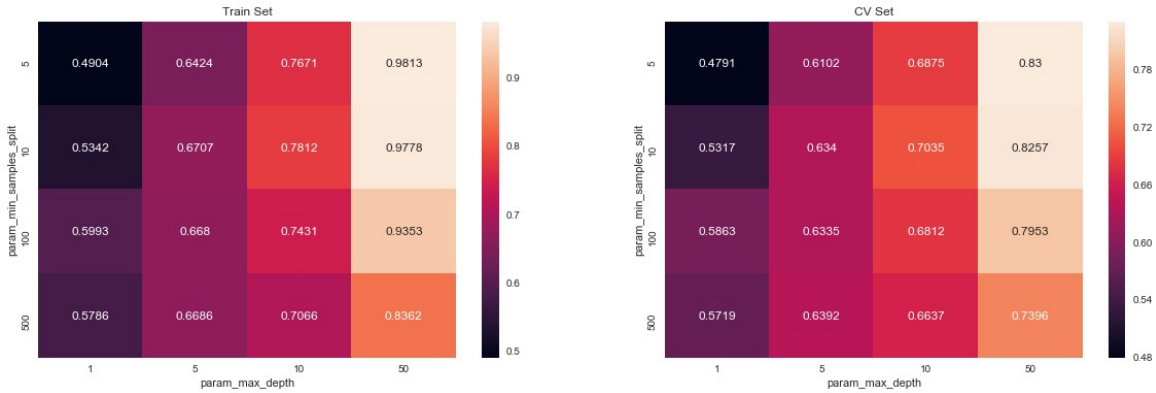
In [71]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier

RF = RandomForestClassifier(class_weight = 'balanced')

tree_para = {'max_depth':[1, 5, 10, 50],'min_samples_split': [5, 10, 10
0, 500]}

clf =  GridSearchCV(RF, tree_para, cv=3)

clf.fit(X_train, y_train)
```

Out[71]:
```
GridSearchCV(cv=3, error_score='raise',
       estimator=RandomForestClassifier(bootstrap=True, class_weight=
'balanced',
            criterion='gini', max_depth=None, max_features='auto',
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=10, n_jobs=1, oob_score=False, random_state=
None,
            verbose=0, warm_start=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'max_depth': [1, 5, 10, 50], 'min_samples_split':
[5, 10, 100, 500]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn
',
       scoring=None, verbose=0)
```

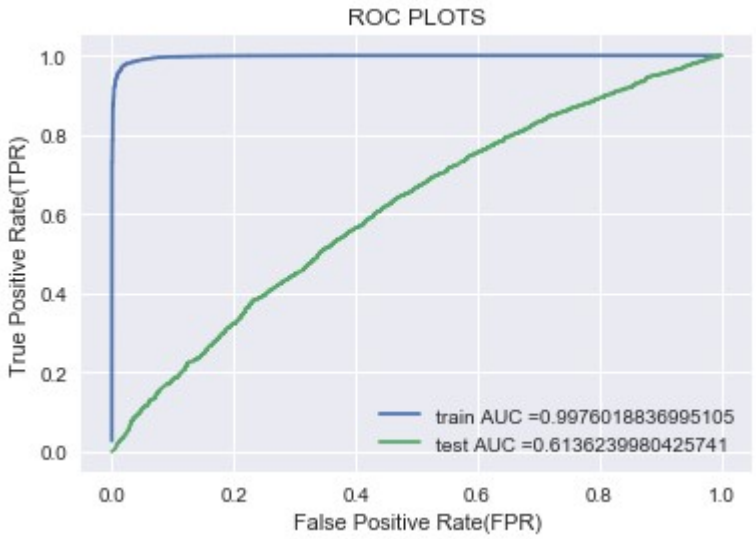
In [72]:
```
max_d = clf.best_params_['max_depth']
min_samp_splt = clf.best_params_['min_samples_split']
```

## Heat map

In [73]:
```
import seaborn as sns; sns.set()
max_scores1 = pd.DataFrame(clf.cv_results_).groupby(['param_min_samples
_split', 'param_max_depth']).max().unstack()[['mean_test_score','mean_t
rain_score']]

fig, ax = plt.subplots(1,2, figsize=(20,6))
sns.heatmap(max_scores1.mean_train_score, annot = True, fmt='.4g', ax=a
x[0])
sns.heatmap(max_scores1.mean_test_score, annot = True, fmt='.4g', ax=ax
[1])
ax[0].set_title('Train Set')
ax[1].set_title('CV Set')
plt.show()
```



In [74]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

RF = RandomForestClassifier(max_depth = max_d, min_samples_split = min_samp_splt, class_weight='balanced')
RF.fit(X_train ,y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probabilities estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(RF, X_train)#Return probability estimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(RF, X_test)#Return probability estimates for the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```



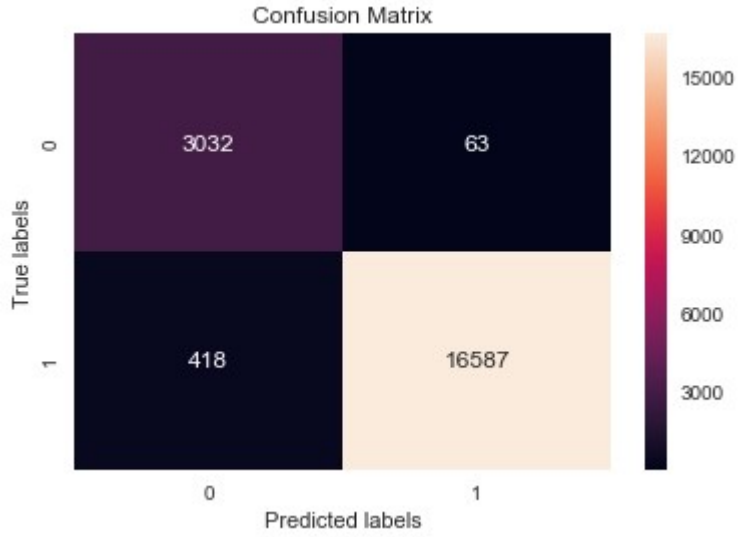

## Confusion Matrix of Train and Test Data

In [75]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is
very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for th
reshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

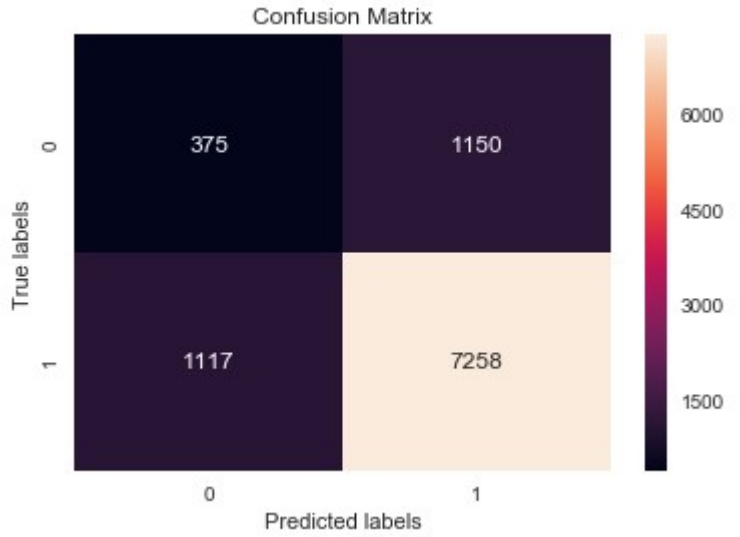
In [76]:
```
from sklearn.metrics import confusion_matrix
best_t = find_best_threshold(thresholds, train_fpr, train_tpr)
print("Train confusion matrix")
print(confusion_matrix(y_train, predict_with_best_t(y_train_pred, best_
t)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict_with_best_t(y_test_pred, best_
t)))
```

```
the maximum value of tpr*(1-fpr) 0.9566585139123293 for threshold 0.6
18
Train confusion matrix
[[ 3032    63]
 [  418 16587]]
Test confusion matrix
[[ 375 1150]
 [1117 7258]]
```

In [77]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```




In [78]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred, b
est_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



## Set 3: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(AVG W2V)+ preprocessed_eassay (AVG W2V). Here for this set take 20K datapoints only.

In [79]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_avg_w2v_titles,train_avg_w2v_essays, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe,test_avg_w2v_titles, test_avg_w2v_essays,   previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
type(X_train)
```

```
(20100, 612) (20100,)
(9900, 612) (9900,)
```

Out[79]:
```
scipy.sparse.csr.csr_matrix
```

In [80]:
```
from sklearn.model_selection import GridSearchCV

RF = RandomForestClassifier(class_weight = 'balanced')

tree_para = {'max_depth':[1, 5, 10, 50],'min_samples_split': [5, 10, 10
0, 500]}

clf =  GridSearchCV(RF, tree_para, cv=3)

clf.fit(X_train, y_train)
```

Out[80]:
```
GridSearchCV(cv=3, error_score='raise',
       estimator=RandomForestClassifier(bootstrap=True, class_weight=
'balanced',
            criterion='gini', max_depth=None, max_features='auto',
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=10, n_jobs=1, oob_score=False, random_state=
None,
            verbose=0, warm_start=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'max_depth': [1, 5, 10, 50], 'min_samples_split':
[5, 10, 100, 500]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn
',
       scoring=None, verbose=0)
```

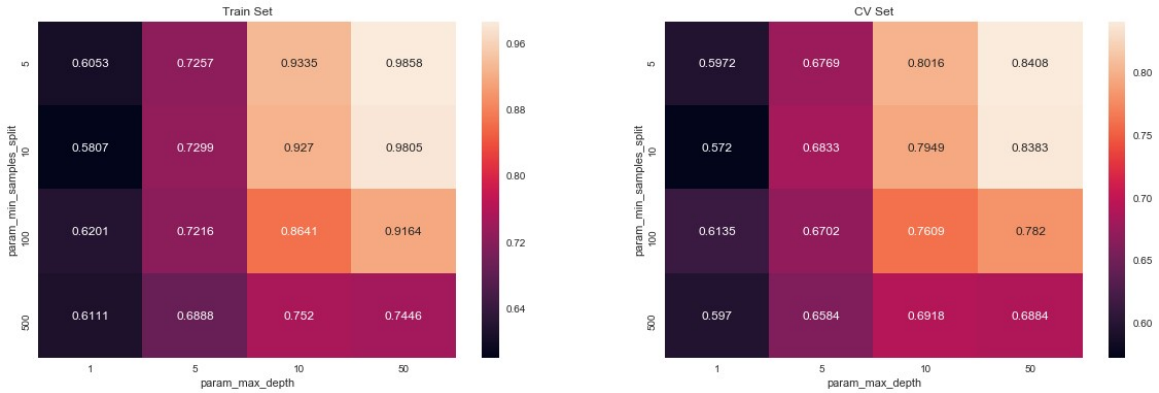
In [81]:
```
clf.best_params_
```

Out[81]:
```
{'max_depth': 50, 'min_samples_split': 5}
```

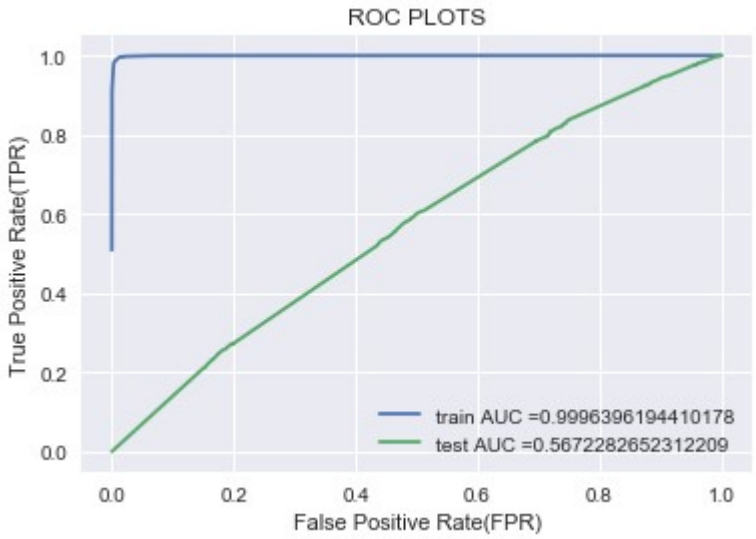
In [82]:
```
max_d = clf.best_params_['max_depth']
min_samp_splt = clf.best_params_['min_samples_split']
```

## Heat map

In [83]:
```
import seaborn as sns; sns.set()
max_scores1 = pd.DataFrame(clf.cv_results_).groupby(['param_min_samples
_split', 'param_max_depth']).max().unstack()[['mean_test_score','mean_t
rain_score']]

fig, ax = plt.subplots(1,2, figsize=(20,6))
sns.heatmap(max_scores1.mean_train_score, annot = True, fmt='.4g', ax=a
x[0])
sns.heatmap(max_scores1.mean_test_score, annot = True, fmt='.4g', ax=ax
[1])
ax[0].set_title('Train Set')
ax[1].set_title('CV Set')
plt.show()
```



In [84]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

RF = RandomForestClassifier(max_depth = max_d, min_samples_split = min_samp_splt, class_weight='balanced')
RF.fit(X_train ,y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probabilities estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(RF, X_train)#Return probability estimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(RF, X_test)#Return probability estimates for the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```

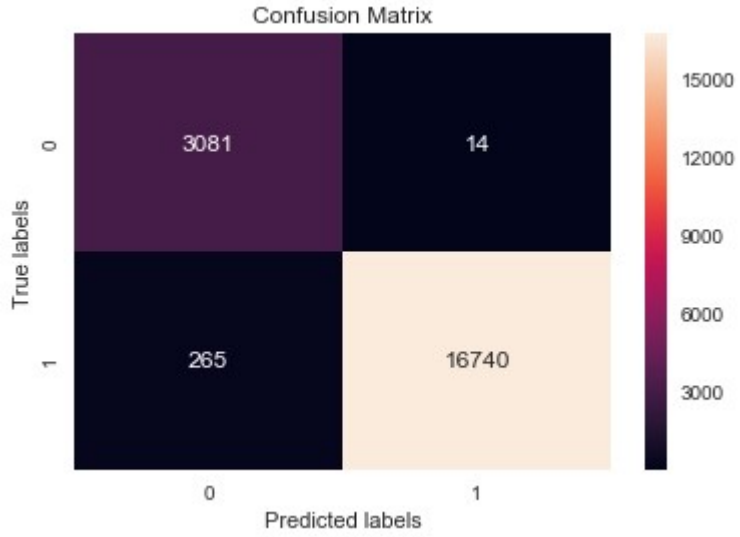


In [85]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is
very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for th
reshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

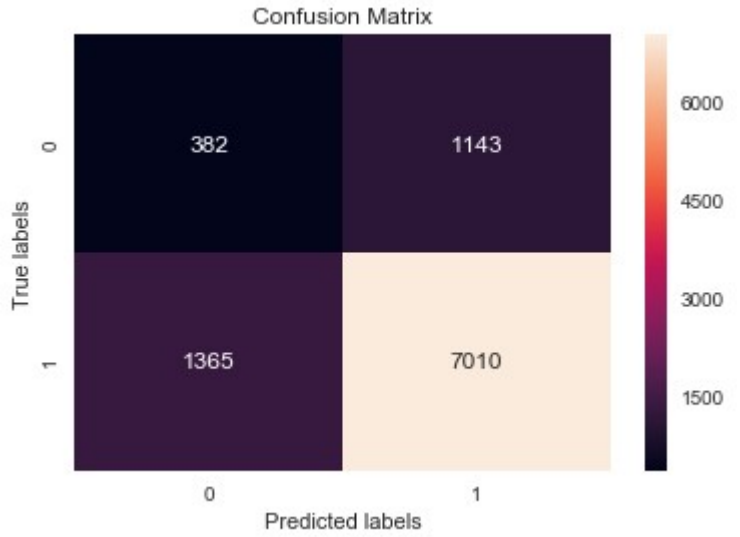
In [86]:
```
from sklearn.metrics import confusion_matrix
best_t = find_best_threshold(thresholds, train_fpr, train_tpr)
print("Train confusion matrix")
print(confusion_matrix(y_train, predict_with_best_t(y_train_pred, best_
t)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict_with_best_t(y_test_pred, best_
t)))
```

```
the maximum value of tpr*(1-fpr) 0.9840401402419415 for threshold 0.7
04
Train confusion matrix
[[ 3081    14]
 [  265 16740]]
Test confusion matrix
[[ 382 1143]
 [1365 7010]]
```

In [87]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```




In [88]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred, b
est_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



## Set 4: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF W2V)+ preprocessed_eassay (TFIDF W2V). Here for this set take 20K datapoints only.

In [89]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_tfidf_w2v_titles, train_tfidf_w2v_titles, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe,test_tfidf_w2v_essays, test_tfidf_w2v_essays,   previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
type(X_train)
```

```
(20100, 612) (20100,)
(9900, 612) (9900,)
```

Out[89]:
```
scipy.sparse.csr.csr_matrix
```

In [90]:
```
from sklearn.model_selection import GridSearchCV

RF = RandomForestClassifier(class_weight = 'balanced')

tree_para = {'max_depth':[1, 5, 10, 50],'min_samples_split': [5, 10, 10
0, 500]}

clf =  GridSearchCV(RF, tree_para, cv=3)

clf.fit(X_train, y_train)
```

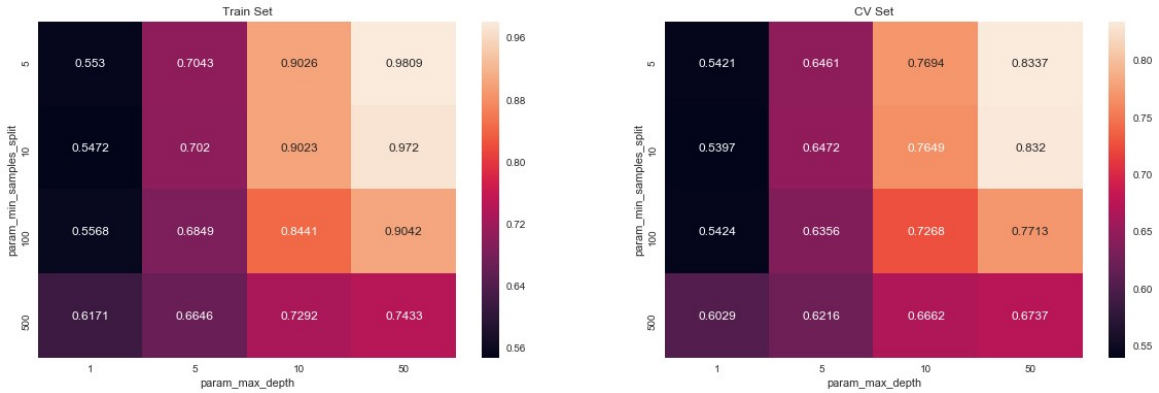
Out[90]:
```
GridSearchCV(cv=3, error_score='raise',
       estimator=RandomForestClassifier(bootstrap=True, class_weight=
'balanced',
            criterion='gini', max_depth=None, max_features='auto',
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=10, n_jobs=1, oob_score=False, random_state=
None,
            verbose=0, warm_start=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'max_depth': [1, 5, 10, 50], 'min_samples_split':
[5, 10, 100, 500]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn
',
       scoring=None, verbose=0)
```

In [91]:
```
clf.best_params_
```

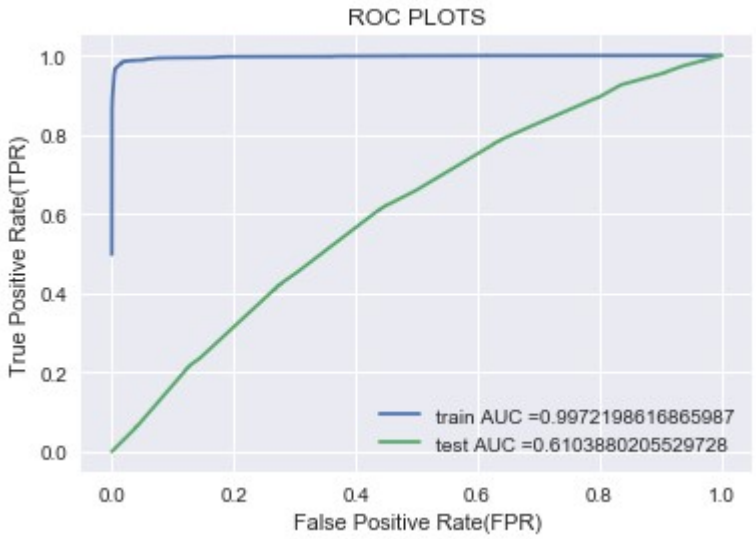
Out[91]:
```
{'max_depth': 50, 'min_samples_split': 5}
```

In [92]:
```
max_d = clf.best_params_['max_depth']
min_samp_splt = clf.best_params_['min_samples_split']
```

In [93]:
```
import seaborn as sns; sns.set()
max_scores1 = pd.DataFrame(clf.cv_results_).groupby(['param_min_samples
_split', 'param_max_depth']).max().unstack()[['mean_test_score','mean_t
rain_score']]

fig, ax = plt.subplots(1,2, figsize=(20,6))
sns.heatmap(max_scores1.mean_train_score, annot = True, fmt='.4g', ax=a
x[0])
sns.heatmap(max_scores1.mean_test_score, annot = True, fmt='.4g', ax=ax
[1])
ax[0].set_title('Train Set')
ax[1].set_title('CV Set')
plt.show()
```



In [94]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

RF = RandomForestClassifier(max_depth = max_d, min_samples_split = min_samp_splt, class_weight='balanced')
RF.fit(X_train ,y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probabilities estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(RF, X_train)#Return probability estimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(RF, X_test)#Return probability estimates for the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```

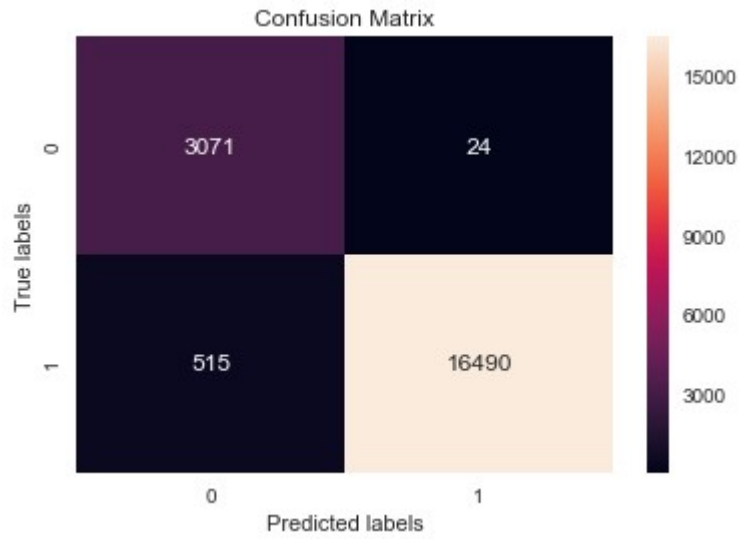


In [95]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is
very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for th
reshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

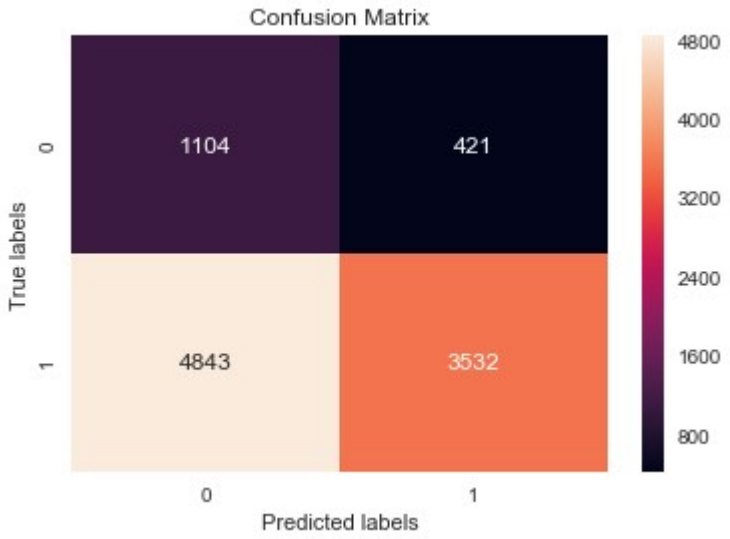
In [96]:
```
print(thresholds.shape, train_fpr.shape,train_tpr.shape)
```

```
(183,) (750,) (750,)
```

In [98]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



In [99]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred, b
est_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```




## Apply Gradient Boosted Decision Trees (GBDT)

## Set 1: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(BOW) + preprocessed_eassay (BOW)

In [100]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_essay_bow, train_title_bow, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe, test_essay_bow, test_title_bow,   previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(20100, 6696) (20100,)
(9900, 6696) (9900,)
```

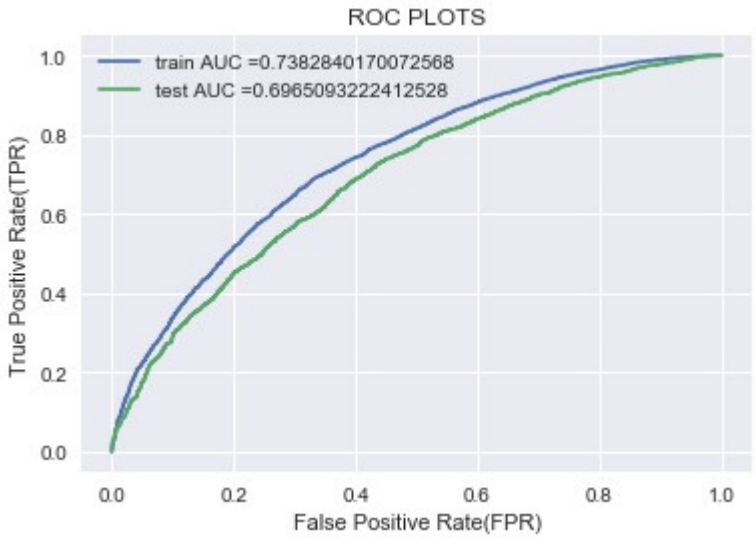
In [101]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier

GBDT = GradientBoostingClassifier()

parameters = {'learning_rate' :  [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3], 'n_estimators' : [5, 10, 50, 75, 100]}

clf =  GridSearchCV(GBDT, parameters, cv=3)

clf.fit(X_train, y_train)
```

Out[101]:
```
GridSearchCV(cv=3, error_score='raise',
       estimator=GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.1, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_decrease=0.0, min_impurity_split=None,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=100,
              presort='auto', random_state=None, subsample=1.0, verbose=0,
              warm_start=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'learning_rate': [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3], 'n_estimators': [5, 10, 50, 75, 100]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
```

In [102]:
```
print(clf.best_params_)
```

```
{'learning_rate': 0.2, 'n_estimators': 10}
```

In [103]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

classifier = GradientBoostingClassifier(learning_rate = 0.1 , n_estimators = 50)
classifier.fit(X_train, y_train)

# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(classifier, X_train)#Return probability estimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(classifier, X_test)#Return probability estimates for the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```



In [104]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is
very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for t
hreshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

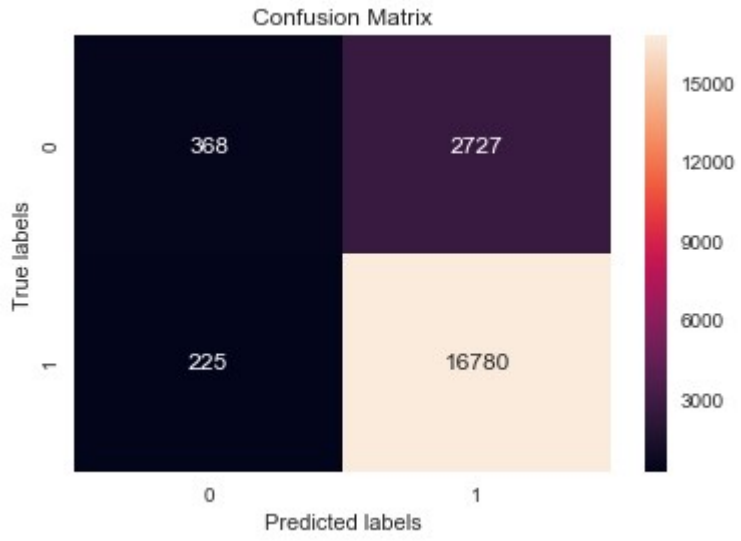
In [105]:
```
print(thresholds.shape, train_fpr.shape,train_tpr.shape)
```

```
(2646,) (5402,) (5402,)
```

In [106]:
```
from sklearn.metrics import confusion_matrix
best_t = find_best_threshold(thresholds, train_fpr, train_tpr)
print("Train confusion matrix")
print(confusion_matrix(y_train, predict_with_best_t(y_train_pred, best
_t)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict_with_best_t(y_test_pred, best_
t)))
```

```
the maximum value of tpr*(1-fpr) 0.46079817824178865 for threshold 0.
717
Train confusion matrix
[[  368  2727]
 [  225 16780]]
Test confusion matrix
[[ 144 1381]
 [ 173 8202]]
```

In [107]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pre
d, best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```




In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```

In [ ]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred, b
est_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```

## Set 2: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF)+ preprocessed_eassay (TFIDF)

In [108]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_essay_tfidf, train_title_tfidf, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe, test_essay_tfidf, test_title_tfidf,  previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

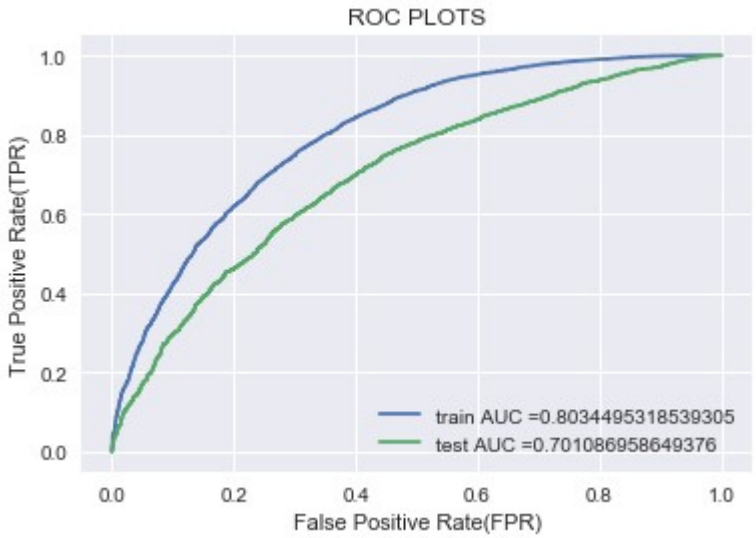
```
(20100, 6696) (20100,)
(9900, 6696) (9900,)
```

In [109]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier

GBDT = GradientBoostingClassifier()

parameters = {'learning_rate' :  [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3], 'n_estimators' : [5, 10, 50, 75, 100]}

clf =  GridSearchCV(GBDT, parameters, cv=3)

clf.fit(X_train, y_train)
```

Out[109]:
```
GridSearchCV(cv=3, error_score='raise',
       estimator=GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.1, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_decrease=0.0, min_impurity_split=None,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=100,
              presort='auto', random_state=None, subsample=1.0, verbose=0,
              warm_start=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'learning_rate': [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3], 'n_estimators': [5, 10, 50, 75, 100]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
```

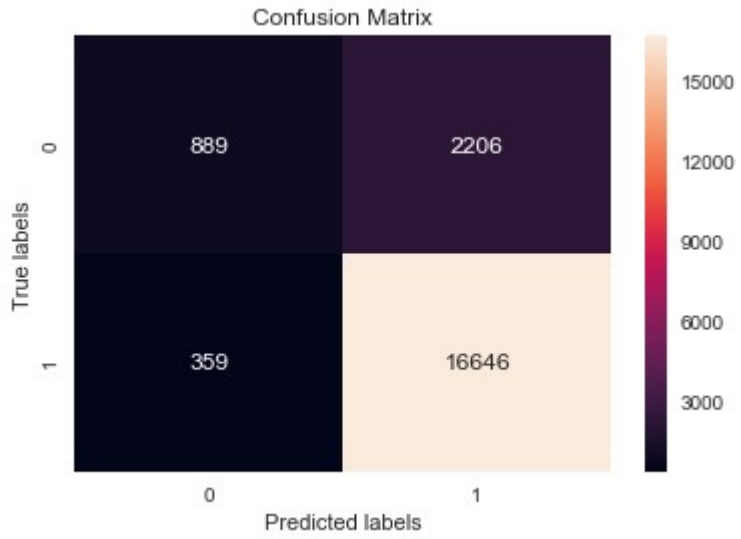
In [112]:
```
iLearning_rate = clf.best_params_['learning_rate']
iN_estimators = clf.best_params_['n_estimators']
```

In [114]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

classifier = GradientBoostingClassifier(learning_rate = iLearning_rate, n_estimators = iN_estimators)
classifier.fit(X_train, y_train)

# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(classifier, X_train)#Return probability estimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(classifier, X_test)#Return probability estimates for the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```

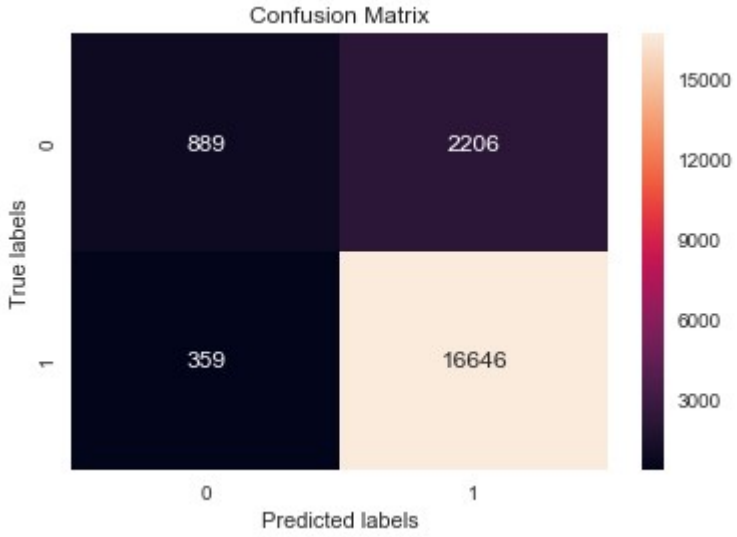


In [115]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is
very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for t
hreshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

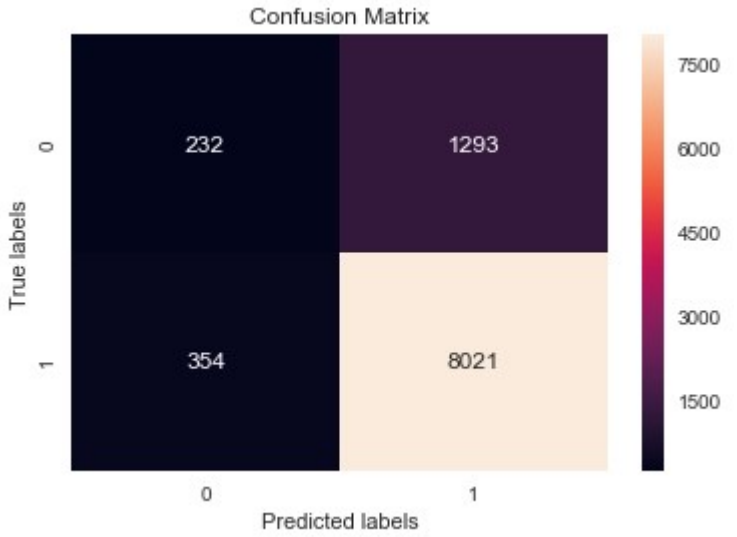
In [116]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pre
d, best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



In [117]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pre
d, best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```




In [118]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



## Set 3: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(AVG W2V)+ preprocessed_eassay (AVG W2V). Here for this set take 20K datapoints only.

In [119]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_avg_w2v_titles,train_avg_w2v_essays, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe,test_avg_w2v_titles, test_avg_w2v_essays,   previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

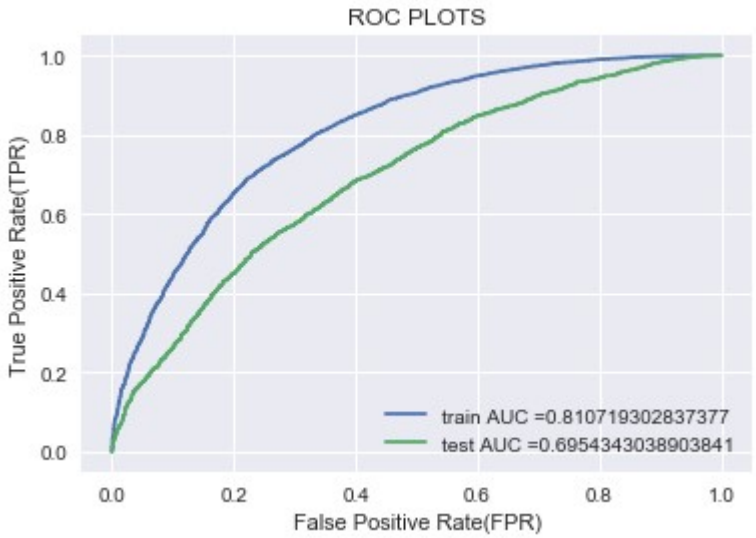
```
(20100, 612) (20100,)
(9900, 612) (9900,)
```

In [120]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier

GBDT = GradientBoostingClassifier()

parameters = {'learning_rate' :  [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]
, 'n_estimators' : [5, 10, 50, 75, 100]}

clf =  GridSearchCV(GBDT, parameters, cv=3)

clf.fit(X_train, y_train)
```

Out[120]:
```
GridSearchCV(cv=3, error_score='raise',
       estimator=GradientBoostingClassifier(criterion='friedman_mse',
init=None,
              learning_rate=0.1, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_decrease=0.0, min_impurity_split=None,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=100,
              presort='auto', random_state=None, subsample=1.0, verbo
se=0,
              warm_start=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'learning_rate': [0.0001, 0.001, 0.01, 0.1, 0.2,
0.3], 'n_estimators': [5, 10, 50, 75, 100]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn
',
       scoring=None, verbose=0)
```

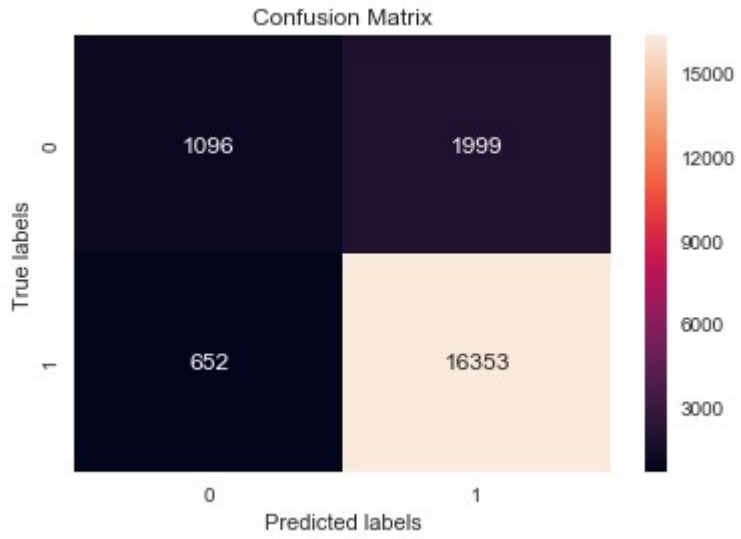
In [121]:
```
iLearning_rate = clf.best_params_['learning_rate']
iN_estimators = clf.best_params_['n_estimators']
```

In [122]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

classifier = GradientBoostingClassifier(learning_rate = iLearning_rate, n_estimators = iN_estimators)
classifier.fit(X_train, y_train)

# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(classifier, X_train)#Return probability estimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(classifier, X_test)#Return probability estimates for the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```

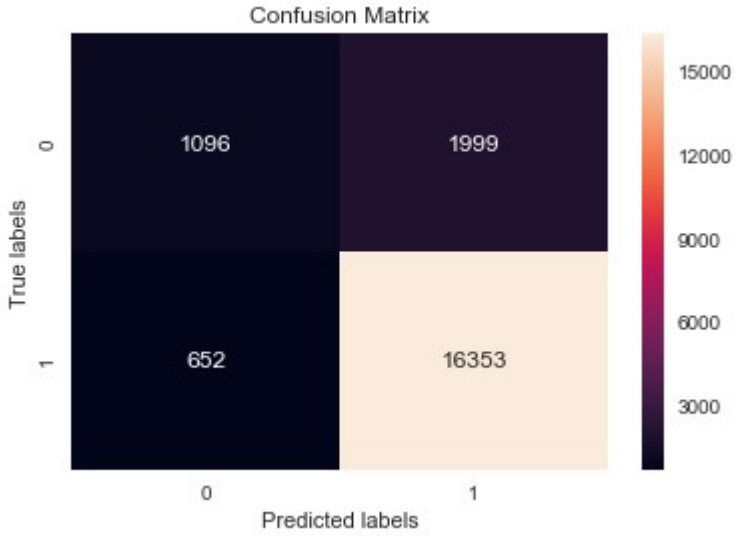


In [123]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is
very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for t
hreshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

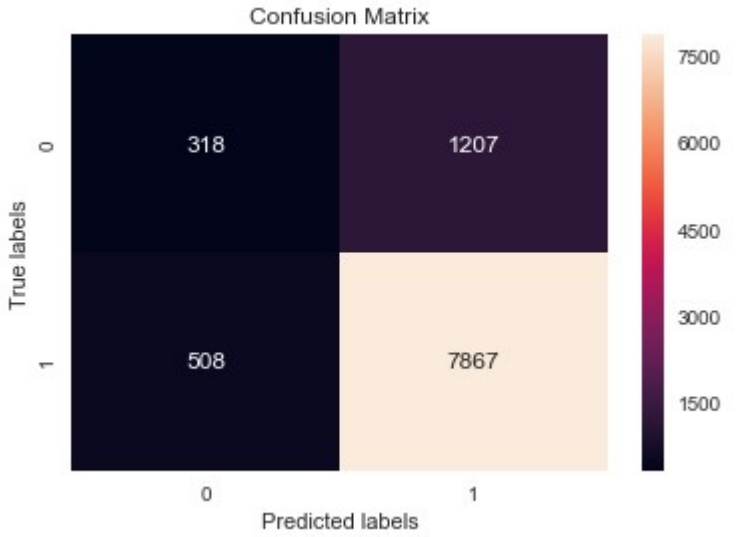
In [124]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pre
d, best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



In [125]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pre
d, best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```




In [126]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```



## Set 4: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF W2V)+ preprocessed_eassay (TFIDF W2V). Here for this set take 20K datapoints only.
</ul>

In [186]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train = hstack((X_train_clean_cat_ohe, X_train_clean_subcat_ohe, X_train_state_ohe, X_train_teacher_ohe, X_train_grade_ohe, train_tfidf_w2v_titles, train_tfidf_w2v_titles, previously_posted_projects_normalized_train, price_normalized_train)).tocsr()
X_test = hstack((X_test_clean_cat_ohe, X_test_clean_subcat_ohe, X_test_state_ohe, X_test_teacher_ohe, X_test_grade_ohe,test_tfidf_w2v_essays, test_tfidf_w2v_essays,   previously_posted_projects_normalized_test, price_normalized_test)).tocsr()

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
type(X_train)
```

```
(20100, 612) (20100,)
(9900, 612) (9900,)
```

Out[186]: scipy.sparse.csr.csr_matrix

In [ ]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier

GBDT = GradientBoostingClassifier()

parameters = {'learning_rate' :  [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3] , 'n_estimators' : [5, 10, 50, 75, 100]}

clf =  GridSearchCV(GBDT, parameters, cv=3)

clf.fit(X_train, y_train)
```

In [ ]:
```
iLearning_rate = clf.best_params_['learning_rate']
iN_estimators = clf.best_params_['n_estimators']
```

In [ ]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve

classifier = GradientBoostingClassifier(learning_rate = iLearning_rate, n_estimators = iN_estimators)
classifier.fit(X_train, y_train)

# roc_auc_score(y_true, y_score) the 2nd parameter should be probabilit y estimates of the positive class
# not the predicted output
y_train_pred =  batch_predict(classifier, X_train)#Return probability e stimates for the  set1x ,for the class label 1 or +ve.
y_test_pred =  batch_predict(classifier, X_test)#Return probability est imates for the setcvx,for the class label 1 or +ve .

train_fpr, train_tpr, thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, thresholds = roc_curve(y_test, y_test_pred)

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, t rain_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_ tpr)))
plt.legend()
plt.xlabel("False Positive Rate(FPR)")
plt.ylabel("True Positive Rate(TPR)")
plt.title("ROC PLOTS")
plt.show()
```

In [ ]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for th reshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    global predictions_

    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions_ = predictions
    return predictions
```

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

ax= plt.subplot()
sns.heatmap(confusion_matrix(y_train, predict_with_best_t(y_train_pred,
best_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```

In [ ]:
```
ax= plt.subplot()
sns.heatmap(confusion_matrix(y_test, predict_with_best_t(y_test_pred, b
est_t)), annot=True, ax = ax,fmt='g');
ax.set_xlabel('Predicted labels');
ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```

# Conclusion

In [195]:
```
# Please compare all your models using Prettytable library
# http://zetcode.com/python/prettytable/
from prettytable import PrettyTable
TB = PrettyTable()

TB.field_names = ["Rand_Forest - MODEL", "HyperparameterS", "Train_AU
C", "Test_Auc"]
TB.title = "Decision Tree"
TB.add_row(["BOW-ENC-RF", "Depth:50  | Samp_Split:5", 0.99543,0.6182
7])
TB.add_row(["TFIDF-ENC-RF", "Depth:50  | Samp_Split:5", 0.99760, 0-613
62])
TB.add_row(["AvgW2V-ENC-RF", "Depth:50   | Samp_Split:5", 0.99963, 0.5
6722])
TB.add_row(["Tf-Idf-ENC-RF", "Depth:50   | Samp_Split:5", 0.997219,0.6
1038])
print(TB)

TB1 = PrettyTable()

TB1.field_names = ["GBDT - MODEL", "HyperparameterS", "Train_AUC", "Te
st_Auc"]
TB1.title = "Gradient Boosting Decision Tree"
TB1.add_row(["BOW-ENC-GBDT", "learning rate:0.2  | n_estimators:10",
0.94,0.71])
TB1.add_row(["TFIDF-ENC-GBDT", "learning rate:0.2  | n_estimators:10",
0.92, 0.68])
TB1.add_row(["AvgW2V-ENC-GBDT", "learning rate:0.2  | n_estimators:1
0", 0.88, 0.67])
TB1.add_row(["Tf-Idf-ENC-GBDT", "learning rate:0.2  | n_estimators:1
0", 0.88, 0.69])
print(TB1)
```

| Rand_Forest - MODEL | HyperparameterS | Train_AUC | Test_Auc |
|---|---|---|---|
| BOW-ENC-RF | Depth:50 | Samp_Split:5 | 0.99543 | 0.61827 |
| TFIDF-ENC-RF | Depth:50 | Samp_Split:5 | 0.9976 | -61362 |
| AvgW2V-ENC-RF | Depth:50 | Samp_Split:5 | 0.99963 | 0.56722 |
| Tf-Idf-ENC-RF | Depth:50 | Samp_Split:5 | 0.997219 | 0.61038 |

| GBDT - MODEL | HyperparameterS | | Train_AUC | Test_Auc |
|---|---|---|---|---|
| BOW-ENC-GBDT | learning rate:0.2 | n_estimators:10 | 0.94 | 0.71 |
| TFIDF-ENC-GBDT | learning rate:0.2 | n_estimators:10 | 0.92 | 0.68 |
| AvgW2V-ENC-GBDT | learning rate:0.2 | n_estimators:10 | 0.88 | 0.67 |
| Tf-Idf-ENC-GBDT | learning rate:0.2 | n_estimators:10 | 0.88 | 0.69 |