In [5]:

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sqlite3
import re,string,math,operator,os
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from sklearn.model_selection import train_test_split, GridSearchCV, RandomizedSearchCV
from sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer,TfidfVectorizer
from sklearn.metrics import accuracy_score,roc_auc_score, confusion_matrix,auc
#from sklearn.preprocessing import
from tqdm import tqdm
from gensim.models import Word2Vec, keyedvectors
import pickle
from chart_studio.plotly import plotly
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
%matplotlib inline
from collections import Counter
```

## 1.1 Reading data

In [6]:

```
import gdown
url = 'https://drive.google.com/uc?id=1bDLwb_Vq7q2W9S89JB96PgmZG3LsLns9'
output = 'train_data.csv' #
https://drive.google.com/file/d/1bDLwb_Vq7q2W9S89JB96PgmZG3LsLns9/view?usp=sharing
gdown.download(url, output, quiet=False)
train=pd.read_csv('train_data.csv',nrows=30000)

url = 'https://drive.google.com/uc?id=1tsD0PCPB1qvwZnaxFxDGiEw0xKlSEFaP'
output= "resources.csv"
gdown.download(url,output,quiet= False )

resources=pd.read_csv('resources.csv')
#data=pd.read_csv('preprocessed_data.csv')
```

```
Downloading...
From: https://drive.google.com/uc?id=1bDLwb_Vq7q2W9S89JB96PgmZG3LsLns9
To: C:\Users\bpash\Downloads\train_data.csv
201MB [00:37, 5.31MB/s]
Downloading...
From: https://drive.google.com/uc?id=1tsD0PCPB1qvwZnaxFxDGiEw0xKlSEFaP
To: C:\Users\bpash\Downloads\resources.csv
127MB [00:21, 5.84MB/s]
```

In [7]:

```
train.shape, train.columns.values
```

Out[7]:

```
((30000, 17),
 array(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
        'project_submitted_datetime', 'project_grade_category',
        'project_subject_categories', 'project_subject_subcategories',
        'project_title', 'project_essay_1', 'project_essay_2',
        'project_essay_3', 'project_essay_4', 'project_resource_summary',
        'teacher_number_of_previously_posted_projects',
        'project_is_approved'], dtype=object))
```

In [8]:

```
resources.shape, resources.columns.values
```

Out[8]:

```
((1541272, 4), array(['id', 'description', 'quantity', 'price'], dtype=object))
```

In [9]:

```
train['date']=pd.to_datetime(train['project_submitted_datetime'].values)
train.drop('project_submitted_datetime',axis=1,inplace=True)
train.sort_values(by=['date'],inplace=True)
```

In [10]:

```
counts=train['project_is_approved'].value_counts()
```

In [11]:

```
train.head(2)
```

Out[11]:

| | Unnamed: 0 | id | teacher_id | teacher_prefix | school_state | project_grade_category | project_subject_ca |
|---|---|---|---|---|---|---|---|
| 473 | 100660 | p234804 | cbc0e38f522143b86d372f8b43d4cff3 | Mrs. | GA | Grades PreK-2 | Applied |
| 29891 | 146723 | p099708 | c0a28c79fe8ad5810da49de47b3fb491 | Mrs. | CA | Grades 3-5 | Math & Science, I |

In [12]:

```
resources.head(2)
```

Out[12]:

| | 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 [13]:

```
train['teacher_prefix'].value_counts()
```

Out[13]:

```
Mrs.        15682
Ms.         10779
Mr.          2895
Teacher       643
Name: teacher_prefix, dtype: int64
```

In [14]:

```
print(train['teacher_prefix'].isna().sum())
train['teacher_prefix']=train['teacher_prefix'].fillna('Mrs.')
print(train['teacher_prefix'].isna().sum())
train['teacher_prefix']=train['teacher_prefix'].str.replace('.','')  # Removing (.) from prefix

#we can also define some function and replace the values and use it as
project_data['teacher_prefix'].astype(str).apply(function)
```

```
1
0
```

In [15]:

```
train['teacher_prefix'].value_counts()
```

Out[15]:

```
Mrs        15683
Ms         10779
Mr          2895
Teacher      643
Name: teacher_prefix, dtype: int64
```

In [16]:

```
train['project_grade_category'].values[:10]
```

Out[16]:

```
array(['Grades PreK-2', 'Grades 3-5', 'Grades PreK-2', 'Grades PreK-2',
       'Grades 3-5', 'Grades PreK-2', 'Grades 6-8', 'Grades 6-8',
       'Grades 3-5', 'Grades 9-12'], dtype=object)
```

In [17]:

```
train['project_grade_category']=train['project_grade_category'].str.replace(' ','')
train['project_grade_category']=train['project_grade_category'].str.replace('-','_')
train['project_grade_category']=train['project_grade_category'].str.replace('Grades','')
```

In [18]:

```
train['project_grade_category'].value_counts()
```

Out[18]:

```
PreK_2    12204
3_5       10160
6_8        4663
9_12       2973
Name: project_grade_category, dtype: int64
```

In [19]:

```
train['project_subject_categories'].values[:100]
```

Out[19]:

```
array(['Applied Learning', 'Math & Science, History & Civics',
       'Literacy & Language', 'Math & Science, Applied Learning',
       'Literacy & Language', 'Literacy & Language', 'Music & The Arts',
       'Applied Learning, Music & The Arts', 'Literacy & Language',
       'Applied Learning', 'Literacy & Language, Special Needs',
       'Literacy & Language', 'Health & Sports',
       'Literacy & Language, Math & Science', 'Literacy & Language',
       'History & Civics, Math & Science',
       'Literacy & Language, Math & Science', 'Health & Sports',
       'Math & Science, Special Needs', 'Literacy & Language',
       'Applied Learning, Music & The Arts',
       'Literacy & Language, Special Needs', 'Health & Sports',
       'Math & Science, Literacy & Language', 'Literacy & Language',
       'Health & Sports', 'Literacy & Language, Special Needs',
       'Health & Sports', 'Literacy & Language, Special Needs',
       'Math & Science', 'Literacy & Language, Math & Science',
       'Music & The Arts', 'Health & Sports, Special Needs',
       'Literacy & Language, Music & The Arts', 'Math & Science',
       'Literacy & Language', 'History & Civics, Literacy & Language',
       'Literacy & Language, Math & Science', 'Literacy & Language',
       'Math & Science', 'History & Civics, Literacy & Language',
       'Math & Science', 'Literacy & Language',
       'Literacy & Language, Math & Science',
       'Literacy & Language, Math & Science',
       'Special Needs, Music & The Arts', 'Literacy & Language',
       'Math & Science', 'Health & Sports, History & Civics',
       'Literacy & Language, Math & Science', 'Literacy & Language',
       'Literacy & Language', 'Math & Science',
       'Literacy & Language, Math & Science',
       'Literacy & Language, Math & Science',
       'Literacy & Language, Math & Science',
       'Applied Learning, Music & The Arts', 'Math & Science',
       'History & Civics, Literacy & Language', 'Special Needs',
       'Literacy & Language, Special Needs', 'Literacy & Language',
       'Applied Learning', 'Math & Science', 'Music & The Arts',
       'Literacy & Language', 'Math & Science, History & Civics',
       'Literacy & Language, Math & Science', 'Literacy & Language',
       'Applied Learning', 'Literacy & Language',
       'Applied Learning, Special Needs', 'Special Needs',
       'Math & Science, History & Civics',
       'Applied Learning, Literacy & Language', 'Math & Science',
       'Literacy & Language, Special Needs', 'Literacy & Language',
       'Literacy & Language, Special Needs', 'Math & Science',
       'Literacy & Language, Special Needs', 'Music & The Arts',
       'Literacy & Language', 'Math & Science', 'Literacy & Language',
       'History & Civics', 'Applied Learning', 'Math & Science',
       'Applied Learning, Music & The Arts',
       'Applied Learning, Special Needs',
       'Literacy & Language, Math & Science',
       'Applied Learning, Literacy & Language',
       'Literacy & Language, Math & Science',
       'Literacy & Language, Math & Science',
       'Literacy & Language, Math & Science', 'Special Needs',
       'Math & Science', 'Math & Science, Literacy & Language',
       'Health & Sports', 'Math & Science, Special Needs',
       'Math & Science', 'Math & Science'], dtype=object)
```

In [20]:

```python
train['project_subject_categories']=train['project_subject_categories'].str.replace(" ",'')
train['project_subject_categories']=train['project_subject_categories'].str.replace("&",'_')
train['project_subject_categories']=train['project_subject_categories'].str.replace("The",'')
cat_list = []
for i in train['project_subject_categories'].values:
    temp = ""
    for j in i.split(','):
        temp+=j.strip()+" "
    cat_list.append(temp.strip().lower())

train['clean_categories'] = cat_list
train.drop('project_subject_categories', axis=1, inplace=True)
# count of all the words in corpus python: https://stackoverflow.com/a/22898595/4084039
from collections import Counter
my_counter = Counter()
for word in train['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]))
print("The Worlds in sorted_cat_dict",sorted_cat_dict)
```

```
The Worlds in sorted_cat_dict {'warmth': 384, 'care_hunger': 384, 'history_civics': 1583,
'music_arts': 2832, 'appliedlearning': 3374, 'specialneeds': 3751, 'health_sports': 3918,
'math_science': 11318, 'literacy_language': 14356}
```

In [21]:

```python
sorted_cat_dict
```

Out[21]:

```
{'warmth': 384,
 'care_hunger': 384,
 'history_civics': 1583,
 'music_arts': 2832,
 'appliedlearning': 3374,
 'specialneeds': 3751,
 'health_sports': 3918,
 'math_science': 11318,
 'literacy_language': 14356}
```

In [22]:

```
train['project_subject_subcategories'].values[:100]
```

Out[22]:

```
array(['Early Development', 'Mathematics, Social Sciences',
       'ESL, Literacy', 'Applied Sciences, Early Development', 'Literacy',
       'Literacy, Literature & Writing', 'Music, Performing Arts',
       'College & Career Prep, Visual Arts',
       'Literacy, Literature & Writing', 'College & Career Prep',
       'Literacy, Special Needs', 'Literature & Writing',
       'Gym & Fitness, Team Sports', 'Literacy, Mathematics', 'Literacy',
       'Economics, Mathematics', 'Literacy, Mathematics',
       'Health & Wellness', 'Mathematics, Special Needs',
       'Literacy, Literature & Writing', 'Early Development, Visual Arts',
       'Literacy, Special Needs', 'Gym & Fitness',
       'Applied Sciences, Literacy', 'Literacy',
       'Gym & Fitness, Team Sports', 'Literacy, Special Needs',
       'Applied Sciences, Environmental Science', 'Literacy, Mathematics',
       'Music', 'Health & Wellness, Special Needs', 'Literacy, Music',
       'Applied Sciences, Mathematics', 'Literacy',
       'History & Geography, Literature & Writing',
       'Literacy, Mathematics', 'Literature & Writing', 'Mathematics',
       'History & Geography, Literacy', 'Applied Sciences, Mathematics',
       'Literacy', 'Literacy, Mathematics',
       'Literature & Writing, Mathematics', 'Special Needs, Visual Arts',
       'Literacy', 'Environmental Science',
       'Health & Wellness, Social Sciences', 'Literacy, Mathematics',
       'Literature & Writing', 'ESL, Literature & Writing',
       'Applied Sciences', 'Literacy, Mathematics',
       'Literature & Writing, Mathematics', 'Literacy, Mathematics',
       'College & Career Prep, Visual Arts', 'Mathematics',
       'History & Geography, Literature & Writing', 'Special Needs',
       'Literature & Writing, Special Needs', 'Literature & Writing',
       'Character Education, Early Development', 'Mathematics',
       'Visual Arts', 'Literacy', 'Applied Sciences, Social Sciences',
       'Literacy, Mathematics', 'Literacy, Literature & Writing', 'Other',
       'Literacy, Literature & Writing', 'Other, Special Needs',
       'Special Needs', 'Applied Sciences, Social Sciences',
       'Early Development, Literacy',
       'Environmental Science, Mathematics', 'Literacy, Special Needs',
       'Literature & Writing', 'Literacy, Special Needs',
       'Applied Sciences, Mathematics', 'Literacy, Special Needs',
       'Visual Arts', 'Literature & Writing', 'Mathematics',
       'ESL, Literacy', 'Civics & Government, Economics',
       'Community Service, Extracurricular', 'Mathematics',
       'Early Development, Visual Arts',
       'Character Education, Special Needs',
       'Literature & Writing, Mathematics', 'Extracurricular, Literacy',
       'Literacy, Mathematics', 'Literacy, Mathematics',
       'Literature & Writing, Mathematics', 'Special Needs',
       'Mathematics', 'Applied Sciences, Literature & Writing',
       'Gym & Fitness', 'Applied Sciences, Special Needs',
       'Applied Sciences', 'Mathematics'], dtype=object)
```

In [23]:

```
train['project_subject_subcategories']=train['project_subject_subcategories'].str.replace(" ",'')
train['project_subject_subcategories']=train['project_subject_subcategories'].str.replace("&",'_')
train['project_subject_subcategories']=train['project_subject_subcategories'].str.replace("The",''
)
```

```
cat_list = []
for i in train['project_subject_subcategories'].values:
    temp = ""
    for j in i.split(','):
        temp+=j.strip()+" "
    cat_list.append(temp.strip().lower())

train['clean_subcategories'] = cat_list
train.drop('project_subject_subcategories', axis=1, inplace=True)
# count of all the words in corpus python: https://stackoverflow.com/a/22898595/4084039
from collections import Counter
my_counter = Counter()
for word in train['clean_categories'].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]))
print("The Worlds in sorted_subcat_dict",sorted_sub_cat_dict)
```

```
The Worlds in sorted_subcat_dict {'warmth': 384, 'care_hunger': 384, 'history_civics': 1583,
'music_arts': 2832, 'appliedlearning': 3374, 'specialneeds': 3751, 'health_sports': 3918,
'math_science': 11318, 'literacy_language': 14356}
```

In [24]:

```
sorted_sub_cat_dict
```

Out[24]:

```
{'warmth': 384,
 'care_hunger': 384,
 'history_civics': 1583,
 'music_arts': 2832,
 'appliedlearning': 3374,
 'specialneeds': 3751,
 'health_sports': 3918,
 'math_science': 11318,
 'literacy_language': 14356}
```

## Adding a new feature Number of words in title

In [25]:

```
train['project_title'].str.len()
```

Out[25]:

```
473      38
29891    34
23374    17
7176     32
5145     35
         ..
18892    28
21335    54
11368    16
27376    17
14678    23
Name: project_title, Length: 30000, dtype: int64
```

In [26]:

```
train["title_word_count"] = train['project_title'].str.split().str.len()
```

In [27]:

```
train.head()
```

Out[27]:

Out[27]:

| | Unnamed: 0 | id | teacher_id | teacher_prefix | school_state | project_grade_category | project_title | proj |
|---|---|---|---|---|---|---|---|---|
| 473 | 100660 | p234804 | cbc0e38f522143b86d372f8b43d4cff3 | Mrs | GA | PreK_2 | Flexible Seating for Flexible Learning | I rec giv |
| 29891 | 146723 | p099708 | c0a28c79fe8ad5810da49de47b3fb491 | Mrs | CA | 3_5 | Breakout Box to Ignite Engagement! | It's tl Rou |
| 23374 | 72317 | p087808 | 598621c141cda5fb184ee7e8ccdd3fcc | Ms | CA | PreK_2 | iPad for Learners | soci |
| 7176 | 79341 | p091436 | bb2599c4a114d211b3381abe9f899bf8 | Mrs | OH | PreK_2 | Robots are Taking over 2nd Grade | Com and sec |
| 5145 | 50256 | p203475 | 63e9a9f2c9811a247f1aa32ee6f92644 | Mrs | CA | 3_5 | Books to Power Powerful Book Clubs! | re b th |

## combining 4 essays into 1 essay

In [28]:

```
train['essay']=train['project_essay_1'].map(str)+train['project_essay_2'].map(str)+train['project_e
ssay_3'].map(str)+train['project_essay_4'].map(str)
```

In [29]:

```
train['essay']
```

Out[29]:

```
473      I recently read an article about giving studen...
29891    It's the end of the school year. Routines have...
23374    Never has society so rapidly changed. Technolo...
7176     Computer coding and robotics, my second grader...
5145     Do you remember the book you read that made yo...
                               ...                        
18892    My first graders are creative, innovative, and...
21335    My school will work with Microsoft's TEALS pro...
11368    I teach 17 amazing students in a Title One sch...
27376    I teach first grade in a Title I school. Altho...
14678    My students range from age four to five years ...
Name: essay, Length: 30000, dtype: object
```

## Adding a new feature Number of words in essay

In [30]:

```
train["essay_word_count"] = train['essay'].str.split().str.len()
```

In [31]:

```
#another way to get count
c=[]
for i in train['essay']:
    a=len(i.split(' '))
    c.append(a)
```

In [32]:

```
train.head(2)
```

Out[32]:

| | Unnamed: 0 | id | teacher_id | teacher_prefix | school_state | project_grade_category | project_title | proj |
|---|---|---|---|---|---|---|---|---|
| 473 | 100660 | p234804 | cbc0e38f522143b86d372f8b43d4cff3 | Mrs | GA | PreK_2 | Flexible Seating for Flexible Learning | I rec giv |
| 29891 | 146723 | p099708 | c0a28c79fe8ad5810da49de47b3fb491 | Mrs | CA | 3_5 | Breakout Box to Ignite Engagement! | It's tl Rou |

## Train Test Split

In [33]:

```
y=train.project_is_approved
X=train.drop('project_is_approved',axis=1)
```

In [34]:

```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.33,stratify=y, random_state=0)
X_train, X_cv, y_train, y_cv =train_test_split(X_train,y_train,test_size=0.33,stratify=y_train,
random_state=0)
```

In [35]:

```
X_train.head(2)
```

Out[35]:

| | Unnamed: 0 | id | teacher_id | teacher_prefix | school_state | project_grade_category | project_title | pr |
|---|---|---|---|---|---|---|---|---|
| 10176 | 79458 | p215379 | 88a608a529899a93032549bc1fd9d844 | Mrs | FL | PreK_2 | Extra! Extra! Headphones Are in Demand! | r e |
| 21272 | 74713 | p126071 | e557f660eb17ed9e674893833246f9d8 | Mrs | WI | PreK_2 | Sensory Activities Support Healthy Development... | Ki |

## Text Preprocessing

In [36]:

```
# printing some random reviews

print(train['essay'].values[0])
print("="*50)
print(train['essay'].values[500])
print("="*50)
```

```
I recently read an article about giving students a choice about how they learn. We already set goa
ls; why not let them choose where to sit, and give them options of what to sit on?I teach at a low
-income (Title 1) school. Every year, I have a class with a range of abilities, yet they are all t
he same age. They learn differently, and they have different interests. Some have ADHD, and some a
```

```
re fast learners. Yet they are eager and active learners that want and need to be able to move aro
und the room, yet have a place that they can be comfortable to complete their work.We need a class
room rug that we can use as a class for reading time, and students can use during other learning t
imes. I have also requested four Kore Kids wobble chairs and four Back Jack padded portable chairs
so that students can still move during whole group lessons without disrupting the class. Having th
ese areas will provide these little ones with a way to wiggle while working.Benjamin Franklin once
said, \"Tell me and I forget, teach me and I may remember, 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, all
by giving them options for comfortable flexible seating.
==================================================
Picture this: A classroom in which all students are engaged in meaningful learning activities, foc
used on their work, and are comfortable with the learning environment that is provided to them. So
unds like a dream, right? This is what I would like to provide for my students with your help!I wo
rk in a very large and diverse elementary school in the heart of West Palm Beach, Florida. Nearly
50% of our students receive free and reduced lunch and come from low income families. My students
come from all different backgrounds. We're somewhat of a melting pot, which I love! Some of my stu
dents come from families that work hard on a daily basis just to get by, while others come from st
able middle class families. I teach students of all cultural and ethnic backgrounds. In my classro
om, I strive to treat all the children who walk through the door as equals. I would love to provid
e the students that can't afford the \"extras\" with what they need to reach their full
potential.Research has proven that students that are given the freedom to choose where and how the
y learn best tend to perform better academically. I would like to provide my students with the
opportunity to be comfortable in their learning environment. Traditional rows of desks are a thing
of the past. In my dream classroom, students would be able to work in the area of the classroom th
at best meets their learning needs. \r\n\r\nStudents who have difficulty focusing will be able to
move as they learn using wobble chairs, stability balls, balance discs, etc., while others may cho
ose a comfortable yoga cushion in a quiet area with soft lighting. As an adult, I often have
difficulty sitting still for extended periods on time, yet we expect this from young children on a
daily basis. The items I have requested will allow my students to move as they work. This will ass
ist with keeping all students engaged and focused on instruction, regardless of their
differentiated abilities.This project will better allow me to engage all the different learners in
my classroom. It will provide all of my students with the opportunity to be successful
academically. I will be able to reach a larger amount of my students by providing an additional ac
commodation for students with ADHD and learning disabilities by implementing a student-centered mo
vement-based learning environment. Thank you so much for your consideration.
==================================================
```

In [37]:

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

def decontracted(phrase):
    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 [38]:

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

In [39]:

```
sent=decontracted("\r\n\r\nEvery week, new technologies emerge that can't could engage students an
d transform their academic understanding into real world action.")
sent=sent.replace('\r',' ')
sent=sent.replace('\n',' ')
sent=sent.replace('\"',' ')
sent
```

Out[39]:

```
'    Every week, new technologies emerge that can not could engage students and transform their ac
ademic understanding into real world action.'
```

In [40]:

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

```
 Every week new technologies emerge that can not could engage students and transform their academi
c understanding into real world action
```

In [41]:

```
preprocessed_essays_train = []
for sentence in tqdm(X_train['essay'].values):
    sentence=' '.join([i for i in sentence.split(' ') if i.lower() not in stopwords])
    sent=decontracted(sentence)
    sent=sentence.replace('\r',' ')
    sent=sentence.replace('\n',' ')
    sent=sentence.replace('\"' ,' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sentence)
    preprocessed_essays_train.append(sent.strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████| 13467/13467
[00:10<00:00, 1260.90it/s]
```

In [42]:

```
preprocessed_essays_train[3]
```

Out[42]:

```
'students come loving homes inner city English Language Learners developing language skills everyd
ay children love read always willing work hard Although reading always come easy children never gi
ve up constantly working improve readers writers often say we never done always read learn r n r n
These children limited resources homes need books resources help promote reading homes well
classroom need materials help keep engaged learners We started integrating art Reading curriculum
students enjoying learning much done acting singing moving visual arts r n r nStudents need constr
uction paper create display learning bring text comprehension life drawing creating scenes texts m
ake three dimensional scenes texts develop character creations using paper requesting r n r nOur c
lass also continues work small groups games requested help us work centers work together develop l
ove reading learning nannan'
```

In [43]:

```
preprocessed_essays_cv= []
for sentence in tqdm(X_cv['essay'].values):
    sentence=' '.join([i for i in sentence.split(' ') if i.lower() not in stopwords])
    sent=decontracted(sentence)
    sent=sentence.replace('\\r',' ')
    sent=sentence.replace('\\n',' ')
    sent=sentence.replace('\\"' ,' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sentence)
    preprocessed_essays_cv.append(sent.strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 6633/6633
[00:03<00:00, 1932.08it/s]
```

In [44]:

```
preprocessed_essays_test = []
for sentence in tqdm(X_test['essay'].values):
    sentence=' '.join([i for i in sentence.split(' ') if i.lower() not in stopwords])
    sent=decontracted(sentence)
    sent=sentence.replace('\\r',' ')
    sent=sentence.replace('\\n',' ')
    sent=sentence.replace('\\"' ,' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sentence)
    preprocessed_essays_test.append(sent.strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 9900/9900
[00:06<00:00, 1452.71it/s]
```

## Preprocessing of project title

In [45]:

```
# printing some randomproject titles.
print(X_train['project_title'].values[0])
print("="*50)
print(train['project_title'].values[150])
print("="*50)
print(train['project_title'].values[1000])
print("="*50)
print(train['project_title'].values[20000])
print("="*50)
```

```
Extra! Extra!  Headphones Are in Demand!
==================================================
Help Keep Us Motivated!
==================================================
Seating to help improve focus
==================================================
White Boards and Markers!
==================================================
```

In [46]:

```
title = decontracted(X_train['project_title'].values[46])
```

In [47]:

```
title
```

Out[47]:

```
'Widescreen Learning'
```

In [48]:

```
X_train['project_title'].values[45:50]
```

Out[48]:

```
array(['Help us Organize our Books', 'Widescreen Learning',
       'Class Headphones', 'Knowledge From Books',
       'Racing to Put an A in STEM!'], dtype=object)
```

In [49]:

```
preprocessed_titles_train = []
for sentence in tqdm(X_train['project_title'].values):
    sentence=' '.join([i for i in sentence.split(' ') if i.lower() not in stopwords])
    sent=decontracted(sentence)
    sent=sentence.replace('\r',' ')
    sent=sentence.replace('\n',' ')
    sent=sentence.replace('\"' ,' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sentence)
    preprocessed_titles_train.append(sent.strip())
```

```
100%|████████████████████████████████████████████████████████████████████████████| 13467/13467
[00:00<00:00, 42596.23it/s]
```

In [50]:

```
preprocessed_titles_cv= []
for sentence in tqdm(X_cv['project_title'].values):
    sentence=' '.join([i for i in sentence.split(' ') if i.lower() not in stopwords])
    sent=decontracted(sentence)
    sent=sentence.replace('\\r',' ')
    sent=sentence.replace('\\n',' ')
    sent=sentence.replace('\\"' ,' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sentence)
    preprocessed_titles_cv.append(sent.strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 6633/6633
[00:00<00:00, 40553.47it/s]
```

In [51]:

```
preprocessed_titles_test = []
for sentence in tqdm(X_test['project_title'].values):
    sentence=' '.join([i for i in sentence.split(' ') if i.lower() not in stopwords])
    sent=decontracted(sentence)
    sent=sentence.replace('\\r',' ')
    sent=sentence.replace('\\n',' ')
    sent=sentence.replace('\\"' ,' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sentence)
    preprocessed_titles_test.append(sent.strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 9900/9900
[00:00<00:00, 37177.89it/s]
```

In [52]:

```
preprocessed_titles_test
```

Out[52]:

```
['Expand Literary Imaginations Fantasy Thrillers',
 'Ready Rug',
 'Roll It Code Learn',
 'Sharp Ears',
 'Ding Dong Merrily',
 'Alternative Seating',
 'Kindergarten Emergent Writers Exploring Expressing Ideas Together',
 'Wreck Problem Rekenrek',
 'Wiggle Wobble Wriggle LEARN',
 'r nReady Read Chapter Books'
 'I hReady Read Chapter Books',
 'Starting Year Right Book',
 'Let s Get Flexible',
 'Staying ACTIVE Even Rainy Days',
 'First Grade Students Need Classroom Area Rug',
 'Finding balance',
 'Individualized Learning Technology Headphones',
 'Help Us Create Comfortable Collaborative Climate',
 'Keeping Kitty Clean Safe',
 'Mini Apples Many Hands',
 'Pop Fly Groundball Catch It',
 'Let Exploring Begin Part 3',
 'Infrared Thermometers Middle School Architects',
 'Give Us iPads Give Us World',
 '3D Doodling Experience',
 'Journalist Block',
 'Enriching Students iPad Exploration',
 'English Language Learner Games Library Aims',
 'Erosion Occur',
 'Flexible Seating Kindergarten',
 'Modern Day Technology Technologically Underserved School',
 'Hands Deck',
 'Piece Together',
 'Need Help Successful',
 'Wiggle Work',
 'No Pencil Challenge',
 'Move Make Room',
 'Rugs Not Floor Coverings but Place Build Community',
 'Getting Comfy Classroom Library',
 'Life 3D',
 'Turning Textbooks Chrome',
 'Leveled Readers Happy Students',
 'Flexible Seating Student Directed Student Centered Learning',
 'Getting Real Scholastic MATH',
 'Ribbons Rulers Please',
 'Building Class Set Chromebooks',
 'What s Happening Today',
 'Learning Interests Us',
 'Stand Desks Life Moves one desk time',
 'Fun Learning',
 'Movement Classroom',
 'Help Students Develop Empathy Combat Bullying Reading',
 'Exploring New World',
 'Hands on Drumming',
 'Cubbies Pre K',
 'Classroom Desires Apple TV Modernize Learning Experience',
 'Technology Future',
 'Think It Print It',
 'Folders Conferences',
 'Grow Green Eat Green',
 'Community Coming Age Literature',
 'Expanding Minds Projector Power',
 'Mr Nisbet s Kids Wants Run Relays',
 'Stop Wiggling',
 'Let s Get Physical Wiggle Learn',
 'Keep Calm Classroom Caddies',
 'Classroom Supplies',
 'Current Events Classroom',
 'Tools New Gadjects',
 'Replenishing Needs',
 'America s Right Fight Revolutionary Perspectives',
 'Wireless Engaged Reading',
 'Flying Macbook Air',
 'Magnetic Personality Part 4',
 '8th graders Reading Cozy Corners lacking materials',
 'Bouncing Classroom',
 'Flexible Seating Focused Students',
 'Financial Literacy All',
 'Using Technology Teach Positive Behavior Social Skills',
 'Kindling Love Knowledge',
 'Solve Seating Paradox Seats Science',
 'Flexible Seating Kindergarteners',
 'School Supplies Scholars',
 'Washing Away Waste Learning Say',
 'Fantasy Rooms Landscapes',
 'Laptops Learning',
 'Thriving Classroom Need Flexible Seating',
 'Academic Success Driven Technology'
 'Academic Success Driven Technology',
 'Chromebook Kids',
 'Coding classroom',
 'Let Us Stand Learn',
 'Shh Trying Concentrate',
 'Chromebooks Kids',
 'Mindfulness CHANGES EVERYTHING r n Health Class',
 'Weather',
 'Take Look Book',
 'Technology Driven Society',
 'Young Architects Work',
 'Entrepreneurs Making',
 'No Frustration New Apple',
 'Integrating Mindfulness Space Breathe',
 'MATHEMATICS GAMES',
 'Wiggling Work',
 'Today reader tomorrow leader Books supplies students',
 'DOTS DASHES CLASSROOM SEATING KINDERGARTEN',
 'Teach students physics music',
 'Bringing Sun Inside',
 'Classroom Fixer Upper Edition',
 'Chromebooks All Chromebooks',
 'Flexible Learning Space',
 'Tech Check',
 'Sky Unlimited HS Students Developing Drone Applications',
 'Scholar Tech',
 'See Eyes',
 'Wiggle Work Empowering Students',
 'Son Lois Lowry',
 'Ready Rock Test',
 'Additional Nonfiction Resource',
 'Inhale Exhale',
 'Classroom Supplies Class 213',
 'Arlington Golden Knights',
 'Reading Space First Grade',
 'Justice Antigone Chicago 2017',
 'Flexible Seating Creating Student Centered Classroom',
 'One Display Read Expand Minds',
 'Unleashing Potential',
 'Special Supplies Bilingual Students',
 'Stability Balls Student Centered Seating',
 'Look Book',
 'Coding Kindles',
 'Nonfiction Bust',
 'Blocks Manipulatives',
 'Science Magnified Looking Glass',
 'SATISFY HUNGER',
 'Help us Hokki pokie',
 'iLove learning iPads 2',
 'Moving Shaking Active stools active kids',
 'Visualizing Electron Emission Spectra',
 'Using FORCE Help Us Read',
 'Theater Arts Without Interface Actors Need Heard',
 'Books Fingertips',
 'BOXES BOOKS NEED',
 'History Making',
 'Literacy Centers Work',
 'Art Literacy Heart',
 'Supplies Supplies Supplies',
 'Good Citizenship Abounds',
 'Wrapping Math Skills',
 'Seating Everyone',
 'Tech Savvy Kinders',
 'Wiggle Wiggle',
 'Discover New World Books',
 'Crafty Kids',
 'Magic Reading',
 'Alternative Seating Options All',
 'Seat',
 'Supplies Needed Successful Middle School Students',
 'Interactive Science Notebooks Millwood Arts Academy',
 'look race relations America r n',
 'Map Skills Today',
 'Thing Fear Not Reading Bud Not Buddy',
 'Graphic Novels Reluctant Readers',
 'Hear Now',
 'COMFY RELAXED HAPPY r nReading Way',
 'Publishing Products'
 'Publishing Products',
 'Learning Movement',
 'Empowering Technology Generation',
 'Economic Decisions Google',
 '100 Books Kindergarten Part 1',
 'Go Go Gadget Lego Robot',
 'Healthy Child Succeeds Academically',
 'Show Wipe',
 'Get Kids Keyboard Connect Community',
 'Beautiful Books Bring Brilliant Brains',
 'Kill Watt Energy',
 'Making Comics Come Life English Classroom',
 'Technology Open Doors',
 'Doodle Knowledge',
 'We ve Barely Scratched SURFACE Potential',
 'Creating Digital Learners',
 'Alternate Seating Active Kids',
 'Hands Math',
 'Need Print Color Pre K',
 'STEM Bundles',
 'Kindergarten Fine Motor Fun',
 'Going Wild',
 'Let s See Big Screen',
 'Creativity Chromebooks Critical Thinking',
 'see it it',
 'Touching Remember',
 'Hot Learning Kindle Fires 2',
 'Journals Kindergarten r nWe Need Write',
 'Wiggle Work Wiggly Seats Wiggly Bodies',
 'Extra Extra Read It My Students Need Almost Everything',
 'Classroom Library English Language Learners',
 'Magazines Curriculum',
 'Paper Projects',
 'Inquiry Based Discovery Laptop Learning',
 'Robots Made Little Bitty Parts',
 'Flexible Seating Attentive Learning',
 'Cooperative Learning 4th Grade',
 'Cultivating Piano Class',
 'Traveling World 3D Printed Monuments',
 'Flexible Seating First Graders',
 'Help Keep Us Track',
 'Help Bronx Student Read A Wrinkle Time',
 'Laptop Kids Special Needs',
 'Discovering Ancient Egypt Art',
 'Exceptional Students Need Opportunities Practice Independent Learning',
 'STEM STEM r nRobotics',
 'Escape Fantasy',
 'Ink Needed First Grade Classroom',
 'TOON Graphic Books',
 'Read It Building Dream Classroom Library',
 'Classroom Call Home',
 'Two Worlds Collide ELA Chinese Immersion Class',
 'Technology Focus',
 'Quality Not Quantity',
 'Flood Kindergarten Class Books',
 'Get Fit XBOX 360 Kinect',
 'Life Gives Lemons Calculate Solve It',
 'perfect tool literacy achievement Kindergarten',
 'Helping 3rd Graders Become Lifelong Readers Learners',
 'Collaborative Learning Table',
 'Move Toward Digital Fluency',
 'Active Kids Healthy Kids',
 'Rosie Rivet',
 'Working Hand Hand Science',
 'Bringing Science Life',
 'Stream Tables Super Scientists',
 'Help Us Bury Purple Projector',
 'Flexible Seating Flexible Learning',
 'Yoga Mats Expanding Mind Body',
 'Flexible Seating Flexible Thinking',
 'Learning Can t Beat Flexible Seat',
 'Write On',
 'Tablet Fun Second Grade',
 'ABCs Kindergarten Literacy Materials',
 'Book Clubs Third Graders',
 'Navigating Native American Folklore Third Grade',
 'Bringing Student Ideas Reality 3D Printing',
 'STEAM Lab wants Cool',
 'STEAM Lab wants Cool',
 'Need Folders Music',
 'Picture Perfect Projects',
 'Maker Space Lego Wall',
 'Popping Workforce',
 'Secret Agents',
 'Mindful Classroom',
 'AAC Ipad Autistic Support Classroom Communication Joy',
 'virtual world oyster',
 'Learning EPIC',
 'Organization Leads Efficiency',
 'Brushing Painting Oh My',
 'Hocus Pocus Revised Right Fit',
 'Fired Science',
 'Support Cause Support Different Learning Styles',
 'Chromebooks Fill Gap',
 'Kindergarten Rug Needed',
 'Students Parents Learning Together Home',
 'Interactive Notebooks New Science Textbook',
 'Library Emerging Readers',
 'Ring Bells Hear',
 'Lego Wall Maker Space',
 'Strengthen Core',
 'Science Prek',
 'Flexible Seating Increase Physical Activity Promote Brain Power',
 'Using Mini Technology Get Maximum Results',
 'Making Day Count Technology',
 'Increasing Engagement Eager Learn Kinders',
 'Fluent First Grade Readers',
 'Makers Gonna Make',
 'Want MOVE',
 'Chromebook Computer Science First',
 'Help Us Hear Tasks',
 'Super Second Graders Need Chromebooks',
 'Hands on Literacy Activities',
 'Way Success',
 'Oh Places Go Good Book',
 'Technology Open Possibilities STEM',
 'Hands on Fun',
 'Rural Remote Small BUT ART',
 'Standing Room Only',
 'Fitness Health Class',
 'Place Wiggle Learn',
 'iPads Individualized Learning',
 'Crazy Chromebooks',
 'Making life long readers',
 'Dire Need Digital Microscopes',
 'Beautiful Books Help Us Grow',
 'Inspiring Positive Academic Devices Students iPads',
 'Masters Math',
 'Improving Literacy Two Year Reading Challenge',
 'Project Polaroid',
 'Let s Hokki Pokey',
 'Chromebooks 21st Century Classroom',
 'Literacy Center Fun',
 'VR Field Trips ESL Students',
 'Books Need Home',
 'Putting Technology Hands 1st Grade Learners',
 'Help engage students education',
 'YoGatta Active',
 'Table Time Ditch Desks',
 'Future NOW',
 'Pottery Club',
 'Science Organized Interactive',
 'Let s Stay Sharp',
 'Comfy Couch Cool Reading Workshop',
 'Bringing Color Comfort r nto the r n Math Science Classroom',
 'Laptop Classroom',
 'Current Events Kids',
 'Wiggle Work',
 'Reach Stars',
 'Pick Seat Works ME',
 'Ever Felt Invisible',
 'Scholars Need Supplies Successful Year',
 'Wiggle Work',
 'Technology Future',
 'Today Guided Reader Tomorrow Leader',
 'Motivating Reluctant Readers Literature Circles',
 'Motivating Reluctant Readers Literature Circles',
 'Healthy Body Healthy Mind',
 'Fun Learning Time',
 'Chromebooks Give Class',
 'Classroom Computer Connection',
 'Rock Without Roll',
 'Learning World Around Us Using Technology',
 'Rolling back Learning Flood 2016',
 'Listen UP Need Headphones',
 'Technology Kindergarten',
 'Third Grade Writing Machines',
 'no wrong answers science',
 'Creating 3D industrial programs',
 'Moving Grooving Helps Us Learn',
 'Creating WONDER ful experience 4th Grade',
 'Clean Germ Free Classroom',
 'Looking Home Classroom Books r n',
 'Apple TV Today Learn Way',
 'Making Good CHOICES',
 'Lifelong Literacy Dreamers',
 'Adventures Reading',
 'Personalized Learning Tools',
 'down hop run ready get fit',
 'Special Needs Read Succeed',
 'Reading Go',
 'Balance Discs',
 'Last Piece Puzzle Document Camera',
 'Making Movement Meaningful',
 'Cultivating Grit Perseverance Learning Art Crochet',
 'Chromebooks Make Difference',
 'middle school Hero s Journey Harry Potter Friends',
 'SUPER SCIENCE MAGAZINES 2',
 'Let s Boogie Kindergarten',
 'Let See Wobble Rock',
 'Display',
 'Get Wiggles Keep Learning',
 'Silence Golden',
 'Students World r nWorld Drumming',
 'Journey Success Reinforcing Character Counts Message',
 'Individual Seating Options',
 'Need Talk',
 'Help Need New Desks',
 'Magical Math Manipulatives',
 'iPad protection continued learning',
 '2nd Graders Become Fluent Reading Math',
 'Stand Mess',
 'Buckets Full Beats Part 2',
 'Right Stuff Write With',
 'Quiet Retreat Another World',
 'Passion Playground',
 'Exploring World Leaving School',
 'Head Shoulders Knees Toes That s Learn',
 'Apple Today Keeps Illiteracy Away',
 'Books Readers',
 'Fine Motor Exploration',
 'Learning Motion Flexible Seating Active Learners',
 'Digital Portfolios Make Learning Fun',
 'Unleashing Creative Writing Round Carpet',
 'Taking Ipad Higher Level Learning',
 'Keeping Organization',
 'Bridging Educational Gap Technology',
 'Lively Little Scientists',
 'Make Middle School Math Interactive',
 'Flexible Seating Focus',
 'Multiplying Minutes add 10 days school year',
 'Lighting Library',
 'Supplies Needed Growing Minds',
 'Place Hands Learning Hands Little Learners',
 'Reading Changes Lives',
 'Table 21st Century Skills',
 'Flexible Seating Transforming Ordinary Extraordinary',
 'Songs trill Ivories tickle',
 'Science Books Beginning Readers',
 'WRAP MULTIPLICATION',
 'Osmo tastic Learning Ipads',
 'Active Sitting Thanks Seat Cushions Increases Student Engagment',
 'Lovin Learnin',
 [illegible]
```

```
 'MAGIC School Club Promote Literacy Numeracy',
 'Reduce Reuse',
 'Involve Us Learn',
 'Surprise Kindles check r nNow time keep surprises protected',
 'Need Paper Pencils Pencil Sharpener',
 'Wiggle Work',
 'Help Gifted Students Soar',
 'Chromebook Collaborations',
 'Special Children Special Needs Sensory Tools',
 'Making Dreams Come True One Book Time',
 'Empowering Students Digital Artistry',
 'Ball STEM',
 'Science Basics Beyond',
 'Busking Way Day',
 'Calming Corner',
 'Stand Sit Lay Creative Seating Options Learning',
 '3D Me 3D Printing Career Skill Development Robotics',
 'Packing Technology',
 'Focus Pay Attention Learn Flexible Seating Options',
 'Grammatically Correct Chromebooks',
 'Flexible Seating Active Learners',
 'Paperless Dream',
 'Chevron Please Fuel School Kindles',
 'Shining Chrome',
 'Journalism 21st Century Students',
 'Today Reader Tomorrow Leader',
 'Comfy Cozy Cooperative learning',
 'Laugh Child Beautiful Music',
 'Yes Want Build Snowman',
 'Learning Fun Let s Play',
 'Me Literature',
 'Today READER Tomorrow LEADER',
 'Kindle Love Reading',
 'Help Us Start 2016 2017 School year RIGHT',
 'Exercise Minds',
 'Chromebooks Learning',
 'Bouncy Chairs Wiggly Bodies',
 'Kid Inspired Walk Learn Brain Breaks',
 'LaBelle High College Decision Day',
 'Upper Case Lower Case Kindle CASE Protecting Tablets',
 'Supply Independence Flexible Classroom',
 'Oh Yes Looking TARGET Health',
 'Give Books Science Math First Graders',
 'Math Games Math Brains',
 'Fund Mrs Fielstra Classroom',
 'Let s Come Carpet Time Learn',
 'Making Math Come Alive Students Special Needs',
 'Dawn Cusick Hooks Readers Animal Snot Spit Slime',
 'Integrating History Literature Art',
 'Comfy Learning',
 'Fueling Learning books',
 'Building STEAM Goose Bay Elementary',
 'Fun Resources Motivates Learning r n',
 'Mathematical Masters',
 'Tumbling Cement Sculptures',
 'Apple Classroom',
 'LapDesks Lead Learners',
 'Flexible Seating Learning Environment Kids Need',
 'Art Across America',
 'Keeping Track Movement K Part II',
 'Wobble Work',
 'Beam Future',
 'Books',
 'Listening Reading',
 'Hands On Osmo r n',
 'Flexible Seating Supporting Differences Learners',
 'Balancing school fun',
 'Neat Seat',
 'Choose Seat r nFocus Learning',
 'Read',
 'Warm Students Winter',
 'Hooked Books',
 'Increasing Engagement Technology',
 'Every Child Abilities Sucessful',
 'Profoundly Impacting Students Technology',
 'Swing Low Sweet Chariot',
 'Welcomed Update',
 'need stencils',
 'Need Read',
 'Help Students Cancer Write Learn New Words',
 'Oh Choices Make',
 'International Garden Club',
 'Wiggle seats wiggly classroom',
 'Outdoor Learning Space',
 'Let Trumpets Sound',
 'Play Play Learn Learning Play',
 'Movement Flexible Seating Classroom',
 'Coding Future',
 'Inspiring Change Love Books',
 'safe quiet space',
 'S O S Saving School Spirit',
 '3rd Grade Balancing Act pt 2',
 'Ready Read Technology',
 'Augmenting FUTURE',
 'Healthy Snacking Produces Quality Performance',
 'Education Gets Techy',
 'Creative Collaborators',
 'Strengthening Brains STEM',
 'Balance Discs Get Students Moving Learning',
 'Splendid Science',
 'TABLETS SHOW US WORLD',
 'AAAAACHHHOOO Where s Kleenex',
 'Moving Towards Technology Based Classroom',
 'Music Ears',
 'Videography Resources Needed',
 'Kinesthetic Kinders Like Move It Move It r n r n',
 'Incorporating Technology Literacy Math',
 'Help We re trapped fluorescent cave',
 'Remember building words refrigerator little r n',
 'Chromebook Classroom Designing Learning',
 'End Stigma Learning Educating Mental Health',
 '3 D Printer GT Projects',
 'Alternative Seating Young Readers',
 'Stop Grumbles',
 'Learning play',
 'Micro Enterprise Project Young Adults Autism',
 'Hopping Around 2nd Grade Hokki Stools',
 'Keep Calm Read',
 'Capturing Special Moments',
 'PreK Listening Lounge',
 'Kinder Learning Activities Practice Pre Reading Math Skills',
 'Let s Get Organized',
 'Projecting Way World',
 'Victorious Volleyball',
 'Wibbly Wobbly Supplies Timey Wimey Work',
 'FLEXIBLE SEATING Alternative seating options 2nd graders',
 'Strengthening STEM Science',
 'Colorful Writing',
 'Technology Tubergen s Tigers',
 'Make Us Techie Busy Bees',
 'Urban Landscape Paintings',
 'High Five Organized Daily Five',
 'Ready Read r nBooks Need',
 'Bringing history life',
 'Second Graders Need Coding Mice Art Science Night',
 'Colossal Intervention',
 'Pre K Targets Healthy Living',
 'Campaigning Earth r nRevisiting 3 R s SIMS',
 'Target Healthy Snacks Part 2',
 'Increasing Achievement Flexible Seating Choices',
 'Snacks Students',
 'Wonder Novel Study',
 'World Instruments Teach Culture',
 'Mr Whitney s Fantastic Fourth',
 'Manipulatives Make Learning Fun',
 'CHARGE IT SURVIVING TECHNOLOGY SAFELY',
 'Flexible Learning Students Sit Anywhere Maximize Learning',
 '21st Century Kinders',
 'Game Time',
 'Lights Camera ACTION',
 'Physical Education all',
 'Preparing Future First Graders',
 'Powerpoint Pupils',
 'Safety Numbers',
 'Technology Interventions Struggling Learners',
 'Exploring Math Home',
 'PMHS Lady Blues Soccer',
 'FALCON Strong',
 'Apple Students',
 'Technology Enhance Learning',
 'Keeping Kids School',
 'Learning Technology Style',
 'Basics Please',
 'Flexible Seating Creative Minds',
 'Staying Active Healthy',
 'Embracing Technology',
 'Supporting Outdoor Learning Experiences',
 'BRINGING MELODY DRUMS',
 'Wiggle Work',
 'Fill Bookshelves New Readers',
 'Time Increase Eat Q 2',
 'Tech ed Ready Learn',
 'Creative Courage Young Hearts',
 'Learning Ball',
 'Rewarding Awesome Cougars',
 'Kinders Love Exciting Colors',
 'High Interest Reading Students',
 'Twinkle Twinkle Little Chromebook',
 'Project Projector',
 'Help Students Project Learning Throughout School',
 'Preparing Future Leaders Technology Driven World',
 'Silly second graders wobble fall down',
 'iRead iWrite iPads',
 'Flexible Seating',
 'Weather Station Outdoor Classroom',
 'Path Good Health r n r nPlay Move Fitness',
 'What s Lung Capacity r n',
 'Experiencing Chicken s Life Cycle',
 'Clay Stars',
 'Read It',
 'Innovative Classroom Seating',
 'Mats Fitness Fun',
 'Going Batty Bees',
 'Beginning Reading Journey Technology',
 'Color World',
 'Building Community',
 'Learning ACTION',
 'Headphones First Graders Digital Age',
 'Listen Learn',
 'Move It Retain IT',
 'Developing Elementary Programmers Using Ozobots',
 'Back School Supplies everyone 2016 2017',
 'Science Class Supplies',
 'Add CHROME Classroom',
 'Give 1st Grader Crayons Journals',
 'Integrated Science Physics Experiment Lab',
 'Classroom Library Extreme Makeover',
 'Moving Learning Go Together',
 'Low Go',
 'Help Us Keep Organized Seat Storage Sacks',
 'Using Technology Means Communicating English Leaners',
 'College Hopeful Geometry Students Need Graphing Calculators',
 'Help us rebuild technology',
 'Arduino Intro Computer Programming',
 'Electricity Craze',
 'Learning Adventure Go Explore',
 'Jump Start Learning Chromebooks',
 'Hands On Science',
 'Robotic Kits High School Inventors Club',
 'Accessorizing Badgy',
 'Supplies 16 17 School Year',
 'Shading Fun Learning Use Tonal Color Values',
 'Scan Print Demand',
 'Supply Right Learn',
 'CHEER CLEAN HEALTHY YEAR',
 'Speech Therapy Language Interventions',
 'Wiggle n Learn Part II',
 'Great Book Discovery',
 'Back School Style',
 'waste time drying rack',
 'Art Meditation Room Coming Together',
 'Creating Communication Success Early Childhood Classroom',
 'Adapting Basketball Everyone',
 'Chromebooks Future Learning',
 'STEM Project Racing Sun Creating Solar Powered Car',
 'Chromebook Math',
 'See Unlimited Possibilities',
 'Kindergarten Techies',
 'Vamos leer',
 'Hokki Pokey Kindergarten',
 'Meeting Individual Needs One Scribble Time',
 'Putting Pro Prototyping',
 'Paperless Classroom Support Argument Driven Inquiry Science',
 'Magnifying Students Ability Learn',
 'Take Look Book',
 'Warmth Care',
 'Happy Readers Retreat',
 'Mathematicians Need Materials Critical Thinking',
 'STEAM Investigation',
 'Hook Reader',
 'Hear Now',
 'Reading Tablets',
 'Enhance Vision firsties',
 'MacBook Professional Learning Center',
 '21st Century Project Based Learning',
 'Readers Become Leaders',
 'Picture This',
 'Floor Tables',
 'Pencils Go',
 'Think Fun Mad Libs',
 'Supplies Engaging Students STEM',
 'Dream It Write It Create',
 'Help Dreams Become Non Fiction',
 'Visual Arts Card Stock',
 'Focus Anything',
 'Talk r nDigital Storytelling Language Acquisition',
 'Study Guides Reading Eyes',
 'Hear book day read right away',
 'Kindle Fire Tablets Interactive Reading Necessary',
 'Manipulating Learning',
 'Life s Great Lessons',
 'Ye Old Book Shop Swap',
 'Read Me Listening Ears On',
 'Learning Fun',
 'Can t Live Without You Supplies',
 'Technology School Home Family Involvement Project',
 'Power Free',
 'Helping Families Work Together Math',
 'Learning History Art Oral Storytelling',
 'Exploring Tech Future Computer Scientists',
 'Books Make World Go Round',
 'Scholars Seek Engaging Reads',
 'Combination Education',
 'Collaborating Technology',
 'Math Minds',
 'Let s get comfy',
 'Keep Moving',
 'Quest 1 1 Classroom',
 'We re Moving First Grade',
 'Filling Classroom Library Books',
 'Joy Sharpened Pencil ESOL Classroom',
 'Hands Tech',
 'Space Cadets Training Future Astronauts',
 'Historical Fiction Helps Us Understand Social Studies',
 'Sunny Day Second Grade',
 'STEM Learning 1st Grade',
 'Rebuilding Flood 2016',
 'Right Books Classroom',
 'Activating Student Learning Standing Desks',
 'Chromebooks Needed 21st Century Learning',
 'Develop Biliteracy Skills',
 'Supplies Donated Violins',
 'Purposeful Positioning Learning',
 'Help Us Fall Love Reading',
 'PROJECT Possible Using technology expand learning potential',
 'Learning Language Tools',
 'Recharged Revamped',
 'Systems Supplies ELA Classroom',
 'Connecting Chromebooks',
 'Stink Ink',
 'Want Shake Wobble Bounce',
 'Hungry Sensory Diet',
 'Materials Military Kids',
 'Let s Get Moving Flexible Seating',
 'Singing Reading Technology',
 'Writing Workshop Open',
 'Wadsworth Students Great Artists',
 'Bouncy Bands',
 'Daily Essentials Galore',
 'Supply Jump Start New Year',
 'Book Bins Organization Make Reading',
 'Recharging Photo Program',
 'Times Mrs Roberson s Kids',
 'Help Keep Classroom Running Like A Well Oiled Machine',
 'Wiggle Work',
 'See Science',
 'Refill Not Landfill',
 'Focused Learning Classroom',
 'Science Technology Math YES PLEASE',
 'Reading Learn Learning Read',
 'Teamwork Dreamwork',
 'What s Inside Box',
 'Sitting standing Moving Staying Focused',
 'Technology Needed Growing 5th Grade Minds',
 'Book Intrigues Mind Captivates Heart',
 'New Teacher Help Students Improve Fine Motor Skills',
 'Building Chromebook Library',
 'Flexible Seating Arrangement Optimal Learning',
 'Bring Excitement 7th Grade',
 'Supplies KinderMath',
 'Future Explorers Light Need Help',
 'Pedaling Way Success',
 'Kindergarten Supplies',
 'Listening Literacy',
 'Move Desk There New Seating Town',
 'Wiggle Worms Bouncy Bottoms',
 'Science Supplemental',
 'STEAM Stations Come',
 'Next Obstacle Please',
 'Open Ears',
 'STEM Classroom Needs Supplies',
 'Please Help Us Honor Students',
 'Electronic Portfolios',
 'Discovering Micro World',
 'Give Access Career',
 'Classroom Library',
 'Teaching Holocaust The Book Thief',
 'Snacks Supporting Students Studies',
 'Prizes Winning Year',
 'Making Reading possibility',
 'Precious Metal lophones',
 'mark get set READ',
 'Zeal Math Tutoring',
 'SAVE SUGAR Containers keep science materials bug free',
 'Big Time Reading Kindergarten',
 'Kindergarten Shuffle',
 'HEAR CLEARLY',
 'iPad Touch Technology',
 'Stage Lighting Theater Production',
 'Kidney table chairs small group instruction',
 'Wanted Cozy Reading Corner',
 'Creating Relaxing Learning Environment',
 'Colorful Rug Shelf Kindergarten Classroom',
 'Reading Writing Go Hand in Hand',
 'Spot Call Home',
 'No Longer Hand Me Down Kids',
 'IB Students Making Mini Inquiries',
 'Make Joyful Noise Choir',
 'Learning COZY Way',
 '10 4 Walkie Talkies Safe School Environment',
 'Alternative seating Miss Huffman s 4th grade class',
 'Moving Grooving',
 'Wiggle Work Way Top',
 'Materials Technology Efficiency',
 'Critical Thinking Begins With',
 'Read Read Read',
 'Kiddies Wobble Fall Down',
 'Engaging Students Kinesthetic Classroom',
 'Involving Parents Technology',
 'Literacy Wiggles',
 'Jump Ropes School Home',
 'Calling WRITERS',
 'Monsters Middle School',
 'Tables FACS Cooking Class',
 'Stability Strength',
 'Learning play The pre k way',
 'Give Giver',
 'Let Exploring Begin',
 'Math Madness',
 'Reading Kindergarteners',
 'iPads Osmo',
 'Lights Camera Action r nShadow Puppet Studio Set',
 'Classroom Comfyness',
 'Motivate Reading',
 'ALTERNATIVE SEATING',
 'Fidget Focus Creating Student Success',
 'Students Need Spanish English Dictionaries',
 'Technology Helps Students Spread Kindness Community',
 'Learning Touch Screen',
 'Working Computers',
 'Moving Bodies Grow Minds',
 'Engaging Students Reading Learning Comfort',
 'Magical Utensils',
 'Costumes Organize',
 'Creative Cuties',
 'Book shopping made easy eager 1st grade readers',
 'Come Kindle Light Fire',
 'STEM Art We re starting STEAM',
 '1 Big iPad 1 Little Robot 25 Excited First Graders',
 'Time Go Teacher Table',
 'Wobble Stools',
 'Classic Literature Empowers High Tech Art',
 'Tackling New Science Standards Hands On',
 'Sensory Success Room',
 'Spanish Language Library',
 'First Grade Flexible Seating',
 'Secret Life Projects',
 'Makerspace Innovation Lab',
 'Help students make 2D designs 3D reality',
 'Read Think Paint',
 'Strengthening Core',
 'Building Love Reading Graphic Novels',
 'Leveled Library Super Readers',
 'Growing Maker Minds',
 'No Skateboarding Campus Says Who',
 'Back Packs Success',
 'Engaging STEM Laboratory Activities',
 'Kill Mockingbird Kill Racism',
 'Endless Supply Books',
 'Wiggle Work Part 3',
 'Personalized Learning 1 1 Chromebooks',
 'Prizes Prizes',
 'Putting Best Foot Forward',
 'Get Bodies Movin',
 'Keeping Reading Flame Alive Kindle Fire',
 'Chromebooks Classics',
 'Keep STEAM Fun',
 'Figure Drawing Proportion Focus',
 'Osmo Ipad STEM Centers',
 'Building Unity Active Play',
 'iMac Creating Learning Opportunities',
 'love technology',
 'Broadcast Journalism Media Art Part 2',
 'Science Materials Pre K',
 'Keeping Kids Focus',
 'Printing Masterpiece Title 1',
 'Let s Give Children Best',
 'Build Vocabulary',
 'Fourth Graders Crave Nonfiction Titles',
 'Therapeutic School Needs Laminator',
 'Help Us Read Write Nonfiction Books',
 'Loving Literacy',
 'Booming Technology',
 'Active Seating Success',
 'Listen This',
 'Becoming Media Literate Nonfiction Books',
 'Science Hands',
 'Butterfly Release Party',
 'Small Group Work Fresh Space',
 'Games Away',
 'Sensory Sensations',
 'Take Seat Enjoy Book Title 1',
 'Help Hawks Soar',
 'Connecting Chromebooks',
 'Teaching Technology',
 'Miss have',
 'Making Speech Language Therapy Successful',
 'need TIME work reading skills',
 'Hip Hop Hooray Let s Play',
 'Need Books Centers Multiple Different Levels Please',
 'Caterpillar Crawl Us',
 'Celebrating Success',
 'Pompoms Friendship Bracelets Weaving Crocheting Fun',
 'Stage Piano',
 'Cheerleaders Rock',
 'Galore Math Games',
 'Make Learning Fun',
 'Digital Kindergarten',
 'TOON Graphics Promote Comprehension',
 'Small Scholars Need iPad Pro',
 'Read Listen Learn 2nd Grade',
 'Creative Writing Chromebooks Part 2',
 'Book Clubs Need Books',
 'Current Events Current Students',
 'Floodzilla Flounders',
 'Getting Parents Involved Reading',
 'Turning Tables CARE Room',
 'STEM Keep Calm Build Something',
 'Boo Boo Fixers',
 'Positive Reinforcement',
 'May Succeed',
 'Help Enrich Stimulate Learning Environment',
 'Lights Camera Action',
 'Green Screen Dream',
 'Headsets Reading Fluencey',
 'Osmo Learn Play doh',
 'Looking School Work Close',
 'Reading Fire Kindle Fire',
 'What s Going Work TEAMWORK',
 'Miss Jessamyn s Musical Classroom',
 'Soccer Gear Year',
 'Keep computer lab nice',
 'Write On',
 'Small Group learning big brains',
 'Feel Rhythm Beat Getting Stronger',
 'Flexible Seating Five Year Olds',
 'Riding Along Stay Fit Healthy',
 'Moving Grooving Learning',
 'Can hear hear',
 'Keep Calm Hydrate Nourish',
 'New Equipment New Found Inspiration',
 'Motivate Readers Technology',
 'Organized Students Free Focus Learning',
 'DREAMING POEMS',
 'Van Buren Community Garden',
 'better student s science abilities',
 'Empowering Students Integration Science Technology Art',
 'Currently Classmates Studying Currents',
 'Tablets Help Us Learning',
 'Structure Function First Grade Science Social Studies',
 'Leveling Playing Field BYOD',
 'Tablet Time',
 'Engineers Future',
 'Touch Tomorrow',
 'Rockin Readers',
 'Magnificent Music Makers 2016',
 'Makeblock Robots Score Goal',
 'Coding Osmo',
 'Standing Desks Movers Shakers',
 'Nonfiction Biographies Historical Informational Books Books Books',
 'Look Who s Listening',
 'Projecting Excellent Year Ahead',
 'Motivating Readers Manipulates',
 'Art Ask',
 'Kindles Kindergarten',
 'Calm Upside World',
 'Bringing Thinking Life',
 'Let s Listen Good Book',
 'Chromebook 3rd Grade Learners 3',
 'Want Omnikin Ball',
 'Sew What',
 'Reading Interactive Fun',
 'Singing Way Sight Words',
 'Mind Blowing Math Motivating Young Mathematicians',
 'Reading Level',
 'Power Art Helps Student Afford College',
 'Breaking Breakout Edu',
 'IPads Mean Can',
 'Boredom Busters Building Baking Bringing Families Together',
 'Books Hands Students',
 'Targeting Fun Recess',
 'Flexible Seating Oasis',
 'Limitless Library',
 'Eat',
 'Learning Around Room',
 'Learning Fun Let s learn social skills games',
 'Living Digital World',
 'Stand Surf Sail',
 'Good Readers Act Stories',
 'Tools Active Learners',
 'Pencils Crayons Glue make Kindergarteners Dreams Come True',
 'Enriching Literacy Art Science Materials',
 'Keep Calm Love Reading',
 'Life Essentials',
 'Visualize Organize',
 'Succeed Reading',
 'Early Explorers Finding Way World',
 'Finding Voices Silent No More',
 'Picture Perfect',
 'iPads 2nd',
 '3Doodling Math',
 'Leaping Technology',
 'Comics Favorite Genre',
 'Seeing Succeed',
 'Centered Around Reading Table Stools',
 'Blossoming Artists',
 'Full STEAM Ahead Young Engineers Need Materials',
 'Comfy Place Gather',
 'Healthy Minds Bodies',
 'Excite Enhance using Technology',
 'Growing Reading Technology',
 'Chromebooks Mini Researchers',
 'Leaders Technology',
 'Mix Illuminate Enhancing Shows Mixing Board Lights',
 'Healthy Kids Smart Kids',
 'Organized Centers',
 'Could Pablo Picasso Create r nMagic Without Canvas',
 'Seating Success',
 'English Classroom Needs Print Words',
 'Flexible Seating Fun Functional Formative',
 'Help Us Calclulate',
 'Help Us Wiggle Wobble Hokki Stools',
 'Supply Learning',
 'Tech Gives Quietest Student Voice',
 'Creative Printing',
 'Listening',
 ...]
```

In [ ]:

```

```

In [53]:

```
X_train['school_state'].value_counts()
vec_state=CountVectorizer()
vec_state.fit(X_train['school_state'].values)

X_train_state_ohe=vec_state.transform(X_train['school_state'].values)
X_cv_state_ohe=vec_state.transform(X_cv['school_state'].values)
X_test_state_ohe=vec_state.transform(X_test['school_state'].values)



print(X_train_state_ohe.shape,y_train.shape)
print(X_cv_state_ohe.shape,y_cv.shape)
print(X_test_state_ohe.shape, y_test.shape)
```

```
(13467, 51) (13467,)
(6633, 51) (6633,)
(9900, 51) (9900,)
```

In [54]:

```
X_train['teacher_prefix'].value_counts()
vec_prefix=CountVectorizer()
vec_prefix.fit(X_train['teacher_prefix'].values)
X_train_teacher_ohe=vec_prefix.transform(X_train['teacher_prefix'].values)
X_cv_teacher_ohe=vec_prefix.transform(X_cv['teacher_prefix'].values)
X_test_teacher_ohe=vec_prefix.transform(X_test['teacher_prefix'].values)

print(X_train_teacher_ohe.shape,y_train.shape)
print(X_cv_teacher_ohe.shape,y_cv.shape)
print(X_test_teacher_ohe.shape, y_test.shape)
print(vec_prefix.get_feature_names())
```

```
(13467, 4) (13467,)
(6633, 4) (6633,)
(9900, 4) (9900,)
['mr', 'mrs', 'ms', 'teacher']
```

In [55]:

```
X_train.project_grade_category.value_counts()
vec_grade_cat=CountVectorizer()
vec_grade_cat.fit(X_train['project_grade_category'].values)
X_train_grade_ohe=vec_grade_cat.transform(X_train['project_grade_category'].values)
X_cv_grade_ohe=vec_grade_cat.transform(X_cv['project_grade_category'].values)
X_test_grade_ohe=vec_grade_cat.transform(X_test['project_grade_category'].values)

print(X_train_grade_ohe.shape,y_train.shape)
print(X_cv_grade_ohe.shape,y_cv.shape)
print(X_test_grade_ohe.shape, y_test.shape)
print(vec_grade_cat.get_feature_names())
```

```
(13467, 4) (13467,)
(6633, 4) (6633,)
(9900, 4) (9900,)
['3_5', '6_8', '9_12', 'prek_2']
```

In [56]:

```
X_train.project_grade_category.value_counts()
vec_clean_sub_cat=CountVectorizer()
vec_clean_sub_cat.fit(X_train['clean_subcategories'].values)
X_train_clean_subcategories_ohe=vec_clean_sub_cat.transform(X_train['clean_subcategories'].values)
X_cv_clean_subcategories_ohe=vec_clean_sub_cat.transform(X_cv['clean_subcategories'].values)
X_clean_subcategories_grade_ohe=vec_clean_sub_cat.transform(X_test['clean_subcategories'].values)

print(X_train_clean_subcategories_ohe.shape,y_train.shape)
print(X_cv_clean_subcategories_ohe.shape,y_cv.shape)
print(X_clean_subcategories_grade_ohe.shape, y_test.shape)
print(vec_clean_sub_cat.get_feature_names())
```

```
(13467, 30) (13467,)
```

```
(6633, 30) (6633,)
(9900, 30) (9900,)
['appliedsciences', 'care_hunger', 'charactereducation', 'civics_government',
'college_careerprep', 'communityservice', 'earlydevelopment', 'economics', 'environmentalscience',
'esl', 'extracurricular', 'financialliteracy', 'foreignlanguages', 'gym_fitness',
'health_lifescience', 'health_wellness', 'history_geography', 'literacy', 'literature_writing', 'm
athematics', 'music', 'nutritioneducation', 'other', 'parentinvolvement', 'performingarts', 'socia
lsciences', 'specialneeds', 'teamsports', 'visualarts', 'warmth']
```

In [57]:

```
X_train.project_grade_category.value_counts()
vec_clean_cat=CountVectorizer()
vec_clean_cat.fit(X_train['clean_categories'].values)
X_train_clean_categories_ohe=vec_clean_cat.transform(X_train['clean_categories'].values)
X_cv_clean_categories_ohe=vec_clean_cat.transform(X_cv['clean_categories'].values)
X_test_clean_categories_ohe=vec_clean_cat.transform(X_test['clean_categories'].values)

print(X_train_clean_categories_ohe.shape,y_train.shape)
print(X_cv_clean_categories_ohe.shape,y_cv.shape)
print(X_test_clean_categories_ohe.shape, y_test.shape)
print(vec_clean_cat.get_feature_names())
```

```
(13467, 9) (13467,)
(6633, 9) (6633,)
(9900, 9) (9900,)
['appliedlearning', 'care_hunger', 'health_sports', 'history_civics', 'literacy_language',
'math_science', 'music_arts', 'specialneeds', 'warmth']
```

## Vectorizing Numerical features

Various numerical feautures are :

1.Price

2.Quantity

3.Number of Projects previously proposed by Teacher

4.Title word Count ( introduced by us)

5.Essay word Count ( introduced by us)

In [58]:

```
resourc_data=resources.groupby('id').agg({'price':sum, 'quantity':sum}).reset_index()
resourc_data.head()
```

Out[58]:

| | id | price | quantity |
|---|---|---|---|
| 0 | p000001 | 459.56 | 7 |
| 1 | p000002 | 515.89 | 21 |
| 2 | p000003 | 298.97 | 4 |
| 3 | p000004 | 1113.69 | 98 |
| 4 | p000005 | 485.99 | 8 |

In [59]:

```
X_train=pd.merge(X_train,resourc_data,on='id',how='left')
X_cv=pd.merge(X_cv,resourc_data,on='id',how='left')
X_test=pd.merge(X_test,resourc_data,on='id',how='left')
```

In [60]:

```
#Price
from sklearn.preprocessing import Normalizer
```

```
X_train['price'].value_counts()

#from sklearn.preprocessing import Normalizer
nm_price=Normalizer()

nm_price.fit(X_train['price'].values.reshape(-1,1))

X_train_price_norm = nm_price.transform(X_train['price'].values.reshape(1,-1).T)
X_cv_price_norm = nm_price.transform(X_cv['price'].values.reshape(1,-1).T)
X_test_price_norm = nm_price.transform(X_test['price'].values.reshape(1,-1).T)


print(X_train_price_norm.shape,y_train.shape)
print(X_cv_price_norm.shape,y_cv.shape)
print(X_test_price_norm.shape,y_test.shape)
```

```
(13467, 1) (13467,)
(6633, 1) (6633,)
(9900, 1) (9900,)
```

In [61]:

```
nm_Teachers=Normalizer()

nm_Teachers.fit(X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1,1))
X_train_No_of_teachers_norm=nm_Teachers.transform(X_train['teacher_number_of_previously_posted_proje
cts'].values.reshape(-1,1))
X_cv_No_of_teachers_norm =
nm_Teachers.transform(X_cv['teacher_number_of_previously_posted_projects'].values.reshape(1,-1).T)
X_test_No_of_teachers_norm =
nm_Teachers.transform(X_test['teacher_number_of_previously_posted_projects'].values.reshape(1,-1).T
)
print("After vectorizations")
print(X_train_No_of_teachers_norm.shape, y_train.shape)
print(X_cv_No_of_teachers_norm.shape, y_cv.shape)
print(X_test_No_of_teachers_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(13467, 1) (13467,)
(6633, 1) (6633,)
(9900, 1) (9900,)
====================================================================================================
```

In [62]:

```
X_train.head(2)
```

Out[62]:

| | Unnamed: 0 | id | teacher_id | teacher_prefix | school_state | project_grade_category | project_title | projec |
|---|---|---|---|---|---|---|---|---|
| 0 | 79458 | p215379 | 88a608a529899a93032549bc1fd9d844 | Mrs | FL | PreK_2 | Extra! Extra! Headphones Are in Demand! | The my st econo |
| 1 | 74713 | p126071 | e557f660eb17ed9e674893833246f9d8 | Mrs | WI | PreK_2 | Sensory Activities Support Healthy Development... | I te Kinder |

2 rows × 21 columns

In [63]:

```
#Quantity
nm_quantity=Normalizer()
nm_quantity.fit(X_train['quantity'].values.reshape(-1,1))
```

```
X_train_quantity_norm = nm_quantity.transform(X_train['price'].values.reshape(1,-1).T)
X_cv_quantity_norm = nm_quantity.transform(X_cv['price'].values.reshape(1,-1).T)
X_test_quantity_norm = nm_quantity.transform(X_test['price'].values.reshape(1,-1).T)


print(X_train_quantity_norm.shape,y_train.shape)
print(X_cv_quantity_norm.shape,y_cv.shape)
print(X_test_quantity_norm.shape,y_test.shape)
```

```
(13467, 1) (13467,)
(6633, 1) (6633,)
(9900, 1) (9900,)
```

In [64]:

```
#title_word_count

nm_tcount=Normalizer()
nm_tcount.fit(X_train.title_word_count.values.reshape(-1,1))
X_train_title_word_count_norm = nm_tcount.transform(X_train['price'].values.reshape(1,-1).T)
X_cv_title_word_count_norm = nm_tcount.transform(X_cv['price'].values.reshape(1,-1).T)
X_test_title_word_count_norm = nm_tcount.transform(X_test['price'].values.reshape(1,-1).T)


print(X_train_title_word_count_norm.shape,y_train.shape)
print(X_cv_title_word_count_norm.shape,y_cv.shape)
print(X_test_title_word_count_norm.shape,y_test.shape)
```

```
(13467, 1) (13467,)
(6633, 1) (6633,)
(9900, 1) (9900,)
```

In [65]:

```
#essay_word_count

nm_ecount=Normalizer()
nm_ecount.fit(X_train.essay_word_count.values.reshape(-1,1))
X_train_essay_word_count_norm = nm_ecount.transform(X_train['price'].values.reshape(1,-1).T)
X_cv_essay_word_count_norm = nm_ecount.transform(X_cv['price'].values.reshape(1,-1).T)
X_test_essay_word_count_norm = nm_ecount.transform(X_test['price'].values.reshape(1,-1).T)


print(X_train_essay_word_count_norm.shape,y_train.shape)
print(X_cv_essay_word_count_norm.shape,y_cv.shape)
print(X_test_essay_word_count_norm.shape,y_test.shape)
```

```
(13467, 1) (13467,)
(6633, 1) (6633,)
(9900, 1) (9900,)
```

## Vectorizing text data

## BAg of Words

In [0]:

```
"""#train essays
#preprocessed_essays=X_train['essay'].values

print("before fitting")
print(X_train.shape,y_train.shape)
print(X_cv.shape,y_cv.shape)
print(X_test.shape, y_test.shape)

print("="*100)
```

```
vector_essay=CountVectorizer(min_df=10,ngram_range=(1,4),max_features=5000)
vector_essay.fit(preprocessed_essays_train)
#we are fitting train essays only and we will transform the train, text and cv data bases on this
model we have fitted here


X_train_essay_bow=vector_essay.transform(preprocessed_essays_train)
X_train_cv_bow=vector_essay.transform(preprocessed_essays_cv)
X_test_bow=vector_essay.transform(preprocessed_essays_test)

print("="*100)
print("after fitting")
print(X_train_essay_bow.shape,y_train.shape)
print(X_train_cv_bow.shape,y_cv.shape)
print(X_test_bow.shape, y_test.shape)"""
```

Out[0]:

```
'#train essays\n#preprocessed_essays=X_train[\'essay\'].values\n\nprint("before
fitting")\nprint(X_train.shape,y_train.shape)\nprint(X_cv.shape,y_cv.shape)\nprint(X_test.shape, y
_test.shape)\n\nprint("="*100)\n\nvector_essay=CountVectorizer(min_df=10,ngram_range=
(1,4),max_features=5000)\nvector_essay.fit(preprocessed_essays_train)\n#we are fitting train
essays only and we will transform the train, text and cv data bases on this model we have fitted h
ere\n\n\nX_train_essay_bow=vector_essay.transform(preprocessed_essays_train)\nX_train_cv_bow=vector
ay.transform(preprocessed_essays_cv)\nX_test_bow=vector_essay.transform(preprocessed_essays_test)\n
int("="*100)\nprint("after
fitting")\nprint(X_train_essay_bow.shape,y_train.shape)\nprint(X_train_cv_bow.shape,y_cv.shape)\npr
X_test_bow.shape, y_test.shape)'
```

In [0]:

```
#titles
```

In [0]:

```
"""print("before fitting")
print(X_train.shape,y_train.shape)
print(X_cv.shape,y_cv.shape)
print(X_test.shape, y_test.shape)

print("="*100)

vector_title=CountVectorizer(min_df=10,max_features=5000)
vector_title.fit(preprocessed_titles_train)
#we are fitting train essays only and we will transform the train, text and cv data bases on this
model we have fitted here


X_train_title_bow=vector_title.transform(preprocessed_titles_train)
X_cv_title_cv_bow=vector_title.transform(preprocessed_titles_cv)
X_test_title_bow=vector_title.transform(preprocessed_titles_test)

print("="*100)
print("after fitting")
print(X_train_title_bow.shape,y_train.shape)
print(X_cv_title_cv_bow.shape,y_cv.shape)
print(X_test_title_bow.shape, y_test.shape)"""
```

Out[0]:

```
'print("before
fitting")\nprint(X_train.shape,y_train.shape)\nprint(X_cv.shape,y_cv.shape)\nprint(X_test.shape, y
_test.shape)\n\nprint("="*100)\n\nvector_title=CountVectorizer(min_df=10,max_features=5000)\nvector
le.fit(preprocessed_titles_train)\n#we are fitting train essays only and we will transform the tra
in, text and cv data bases on this model we have fitted
here\n\n\nX_train_title_bow=vector_title.transform(preprocessed_titles_train)\nX_cv_title_cv_bow=ve
_title.transform(preprocessed_titles_cv)\nX_test_title_bow=vector_title.transform(preprocessed_titl
est)\n\nprint("="*100)\nprint("after
fitting")\nprint(X_train_title_bow.shape,y_train.shape)\nprint(X_cv_title_cv_bow.shape,y_cv.shape)\
nt(X_test_title_bow.shape, y_test.shape)'
```

## Tfidf

In [66]:

```
vec_tfidf_essay=TfidfVectorizer(min_df=10,max_features=5000)
vec_tfidf_essay.fit(preprocessed_titles_train)
X_train_title_tfidf=vec_tfidf_essay.transform(preprocessed_titles_train)
X_train_title_cv_tfidf=vec_tfidf_essay.transform(preprocessed_titles_cv)
X_test_title_tfidf=vec_tfidf_essay.transform(preprocessed_titles_test)

print("="*100)
print("after fitting")
print(X_train_title_tfidf.shape,y_train.shape)
print(X_train_title_cv_tfidf.shape,y_cv.shape)
print(X_test_title_tfidf.shape, y_test.shape)
```

```
====================================================================================================

after fitting
(13467, 796) (13467,)
(6633, 796) (6633,)
(9900, 796) (9900,)
```

In [67]:

```
vec_tfidf_title=TfidfVectorizer(min_df=10,max_features=5000)
vec_tfidf_title.fit(preprocessed_essays_train)
X_train_essay_tfidf=vec_tfidf_title.transform(preprocessed_essays_train)
X_train_essay_cv_tfidf=vec_tfidf_title.transform(preprocessed_essays_cv)
X_test_essay_tfidf=vec_tfidf_title.transform(preprocessed_essays_test)

print("="*100)
print("after fitting")
print(X_train_essay_tfidf.shape,y_train.shape)
print(X_train_essay_cv_tfidf.shape,y_cv.shape)
print(X_test_essay_tfidf.shape, y_test.shape)
```

```
====================================================================================================

after fitting
(13467, 5000) (13467,)
(6633, 5000) (6633,)
(9900, 5000) (9900,)
```

In [68]:

```
# stronging variables into pickle files python: http://www.jessicayung.com/how-to-use-pickle-to-sa
ve-and-load-variables-in-python/
# make sure you have the glove_vectors file

import gdown
url = 'https://drive.google.com/uc?id=1MqUasf7jYoPbG35MJ28VQcOjjNp-ZDDp'
output = 'glove_vectors'
gdown.download(url, output, quiet=False)
with open('glove_vectors', 'rb') as f:
    model = pickle.load(f)
    glove_words =  set(model.keys())
```

```
Downloading...
From: https://drive.google.com/uc?id=1MqUasf7jYoPbG35MJ28VQcOjjNp-ZDDp
To: C:\Users\bpash\Downloads\glove_vectors
128MB [00:25, 5.07MB/s]
```

In [0]:

```
# average Word2Vec for train essays
# computing average word2vec for each review.
avg_w2v_essays=[]
for sentence in tqdm(preprocessed_essays_train):
    vector=np.zeros(300)
    for word in sentence.split(' '):
        if word in glove_words:
            cnt_wrds+=1
            vector+=model[word]
    if cnt_wrds:
            vector/=cnt_wrds

    avg_w2v_essays.append(vector)
```

```
100%|██████████| 22445/22445 [00:15<00:00, 1407.28it/s]
```

In [0]:

```
# average Word2Vec for cv essays
# computing average word2vec for each review.
avg_w2v_essays_cv=[]
for sentence in tqdm(preprocessed_essays_cv):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if word in glove_words:
            cnt_wrds+=1
            vector+=model[word]
    if cnt_wrds:
            vector/=cnt_wrds

    avg_w2v_essays_cv.append(vector)
```

```
100%|██████████| 11055/11055 [00:07<00:00, 1428.14it/s]
```

In [0]:

```
# average Word2Vec for test essays
# computing average word2vec for each review.
avg_w2v_essays_test=[]
for sentence in tqdm(preprocessed_essays_test):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if word in glove_words:
            cnt_wrds+=1
            vector+=model[word]
    if cnt_wrds:
            vector/=cnt_wrds

    avg_w2v_essays_test.append(vector)
```

```
100%|██████████| 16500/16500 [00:11<00:00, 1403.28it/s]
```

In [0]:

```
# average Word2Vec for train titles
# computing average word2vec for each review.
avg_w2v_titles_train=[]
for sentence in tqdm(preprocessed_titles_train):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if word in glove_words:
            cnt_wrds+=1
            vector+=model[word]
    if cnt_wrds:
            vector/=cnt_wrds

    avg_w2v_titles_train.append(vector)
```

```
100%|██████████| 22445/22445 [00:00<00:00, 131439.28it/s]
```

In [0]:

```
# average Word2Vec for cv titles
# computing average word2vec for each review.
avg_w2v_titles_cv=[]
for sentence in tqdm(preprocessed_titles_cv):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if word in glove_words:
            cnt_wrds+=1
            vector+=model[word]
    if cnt_wrds:
            vector/=cnt_wrds

    avg_w2v_titles_cv.append(vector)
```

```
100%|██████████| 11055/11055 [00:00<00:00, 111494.31it/s]
```

In [0]:

```
# average Word2Vec for test titles
# computing average word2vec for each review.
avg_w2v_titles_test=[]
for sentence in tqdm(preprocessed_titles_test):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if word in glove_words:
            cnt_wrds+=1
            vector+=model[word]
    if cnt_wrds:
            vector/=cnt_wrds

    avg_w2v_titles_test.append(vector)
```

```
100%|██████████| 16500/16500 [00:00<00:00, 132797.87it/s]
```

## weighted tfidf w2v

In [69]:

```
vec_tfidf_w2v=TfidfVectorizer()
t=vec_tfidf_w2v.fit(preprocessed_essays_train)
dictionary=dict(zip(vec_tfidf_w2v.get_feature_names(),list(vec_tfidf_w2v.idf_)))
tfidf_words=set(vec_tfidf_w2v.get_feature_names())
```

In [71]:

```
tfidf_w2v_essays_train=[]
for sentence in tqdm(preprocessed_essays_train):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if (word in glove_words) and (word in tfidf_words):
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split(' ')))


            cnt_wrds+=tf_idf
            vector+=(model[word]*tf_idf)
    if cnt_wrds:
            vector/=cnt_wrds

    tfidf_w2v_essays_train.append(vector)
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 13467/13467 [00:
```

```
29<00:00, 455.36it/s]
```

In [72]:

```
len(tfidf_w2v_essays_train)
```

Out[72]:

```
13467
```

In [73]:

```
tfidf_w2v_essays_test=[]
for sentence in tqdm(preprocessed_essays_test):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if (word in glove_words) and (word in tfidf_words):
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split(' ')))


            cnt_wrds+=tf_idf
            vector+=(model[word]*tf_idf)
    if cnt_wrds:
            vector/=cnt_wrds

    tfidf_w2v_essays_test.append(vector)
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 9900/9900
[00:21<00:00, 466.63it/s]
```

In [74]:

```
tfidf_w2v_essays_cv=[]
for sentence in tqdm(preprocessed_essays_cv):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if (word in glove_words) and (word in tfidf_words):
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split(' ')))


            cnt_wrds+=tf_idf
            vector+=(model[word]*tf_idf)
    if cnt_wrds:
            vector/=cnt_wrds

    tfidf_w2v_essays_cv.append(vector)
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 6633/6633
[00:14<00:00, 454.74it/s]
```

In [75]:

```
vec_tfidf_w2v_title=TfidfVectorizer()
t=vec_tfidf_w2v_title.fit(preprocessed_titles_train)
dictionary=dict(zip(vec_tfidf_w2v_title.get_feature_names(),list(vec_tfidf_w2v_title.idf_)))
tfidf_words=set(vec_tfidf_w2v_title.get_feature_names())
```

In [76]:

```
tfidf_w2v_title_essays=[]
for sentence in tqdm(preprocessed_titles_train):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if (word in glove_words) and (word in tfidf_words):
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split(' ')))


            cnt_wrds+=tf_idf
            cnt_wrds+=tf_idf
            vector+=(model[word]*tf_idf)
    if cnt_wrds:
            vector/=cnt_wrds

    tfidf_w2v_title_essays.append(vector)
```

```
100%|██████████████████████████████████████████████████████████████████████████| 13467/13467
[00:00<00:00, 83842.87it/s]
```

In [77]:

```
len(tfidf_w2v_title_essays)
```

Out[77]:

```
13467
```

In [78]:

```
tfidf_w2v_title_test=[]
for sentence in tqdm(preprocessed_titles_test):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if (word in glove_words) and (word in tfidf_words):
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split(' ')))


            cnt_wrds+=tf_idf
            vector+=(model[word]*tf_idf)
    if cnt_wrds:
            vector/=cnt_wrds

    tfidf_w2v_title_test.append(vector)
```

```
100%|███████████████████████████████████████████████████████████████████████████| 9900/9900
[00:00<00:00, 194538.27it/s]
```

In [79]:

```
tfidf_w2v_title_cv=[]
for sentence in tqdm(preprocessed_titles_cv):
    vector=np.zeros(300)
    cnt_wrds=0
    for word in sentence.split(' '):
        if (word in glove_words) and (word in tfidf_words):
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split(' ')))


            cnt_wrds+=tf_idf
            vector+=(model[word]*tf_idf)
    if cnt_wrds:
        vector/=cnt_wrds

    tfidf_w2v_title_cv.append(vector)
```

```
100%|███████████████████████████████████████████████████████████████████████████| 6633/6633
[00:00<00:00, 190029.02it/s]
```

In [80]:

```
len(tfidf_w2v_title_cv)
```

Out[80]:

```
6633
```

## Apply Decision Tree on these feature sets

## Apply Decision Tree on these feature sets

categorical, numerical features + preprocessed_essay (TFIDF W2V)

In [81]:

```
from scipy.sparse import hstack
```

In [82]:

```
X_cr=hstack((X_cv_state_ohe,X_cv_teacher_ohe,X_cv_grade_ohe,X_cv_clean_subcategories_ohe,X_cv_clean
_categories_ohe,X_cv_price_norm,X_cv_No_of_teachers_norm,X_cv_quantity_norm,X_cv_title_word_count_r
orm,X_cv_essay_word_count_norm,
tfidf_w2v_essays_cv,tfidf_w2v_title_cv)).tocsr()

X_tr=hstack((X_train_state_ohe,X_train_teacher_ohe,X_train_grade_ohe,X_train_clean_subcategories_oh
,
             X_train_clean_categories_ohe,X_train_price_norm,X_train_No_of_teachers_norm,X_train_qu
antity_norm,
             X_train_title_word_count_norm,X_train_essay_word_count_norm,
tfidf_w2v_essays_train,tfidf_w2v_title_essays)).tocsr()
X_te=hstack((X_test_state_ohe,X_test_teacher_ohe,X_test_grade_ohe,X_clean_subcategories_grade_ohe,
X_test_clean_categories_ohe,X_test_price_norm,X_test_No_of_teachers_norm,X_test_quantity_norm,X_te
st_title_word_count_norm,X_test_essay_word_count_norm,
tfidf_w2v_essays_test,tfidf_w2v_title_test)).tocsr()

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

```
Final Data matrix
(13467, 703) (13467,)
(6633, 703) (6633,)
(9900, 703) (9900,)
====================================================================================================
```

In [83]:

```
%%time
from sklearn.tree import DecisionTreeClassifier
tree=DecisionTreeClassifier(class_weight='balanced',random_state=0)
#The hyper parameter tuning (best `depth` in range [1, 5, 10, 50], and the best
`min_samples_split` in range [5, 10, 100, 500])
paramaters={'max_depth':[1, 5, 10, 50],'min_samples_split': [5, 10,100,500]}

#clf=GridSearchCV(tree,paramaters,cv=10,scoring='roc_auc',return_train_score=True,verbose=2)
clf1=GridSearchCV(tree,paramaters,cv=10,scoring='roc_auc',n_jobs=-1,return_train_score=True,verbose
=2)
clf1.fit(X_tr,y_train)
```

```
Fitting 10 folds for each of 16 candidates, totalling 160 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed:    9.0s
[Parallel(n_jobs=-1)]: Done 160 out of 160 | elapsed:  3.7min finished
```

```
Wall time: 3min 54s
```

Out[83]:

```
GridSearchCV(cv=10, error_score=nan,
             estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                              class_weight='balanced',
                                              criterion='gini', max_depth=None,
                                              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,
                                                              presort='deprecated',
                                                              random_state=0, splitter='best'),
                       iid='deprecated', 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=True,
                       scoring='roc_auc', verbose=2)
```

In [84]:

```
train_auc=clf1.cv_results_['mean_train_score']
train_auc_std=clf1.cv_results_['std_train_score']
cv_auc=clf1.cv_results_['mean_test_score']
cv_auc_std=clf1.cv_results_['std_test_score']
print(clf1.best_estimator_)
print("Using the best parametrs predict the best score for Test", clf1.score(X_tr,y_train))
print("Using the best parametrs predict the best score for Test",clf1.score(X_te,y_test))
```

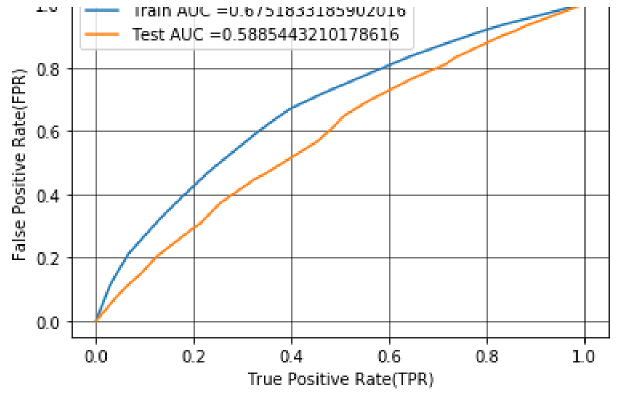
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight='balanced', criterion='gini',
                       max_depth=5, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=500,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=0, splitter='best')
Using the best parametrs predict the best score for Test 0.6751833185902016
Using the best parametrs predict the best score for Test 0.5885443210178616
```

In [85]:

```
def batch_predict(clf, data):
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the posi
tive 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 [86]:

```
%%time
from sklearn.metrics import roc_curve
#{'max_depth': 10, 'min_samples_split': 500}

tree_wtfidf=DecisionTreeClassifier(class_weight='balanced',max_depth=5,min_samples_split=500,
random_state=0)
tree_wtfidf.fit(X_tr,y_train)
y_train_pred = batch_predict(tree_wtfidf, X_tr)
y_test_pred = batch_predict(tree_wtfidf, X_te)

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_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("True Positive Rate(TPR)")
plt.ylabel("False Positive Rate(FPR)")
plt.title("AUC")
plt.grid(color='black', linestyle='-', linewidth=0.5)
plt.show()
```



```
Wall time: 6.56 s
```

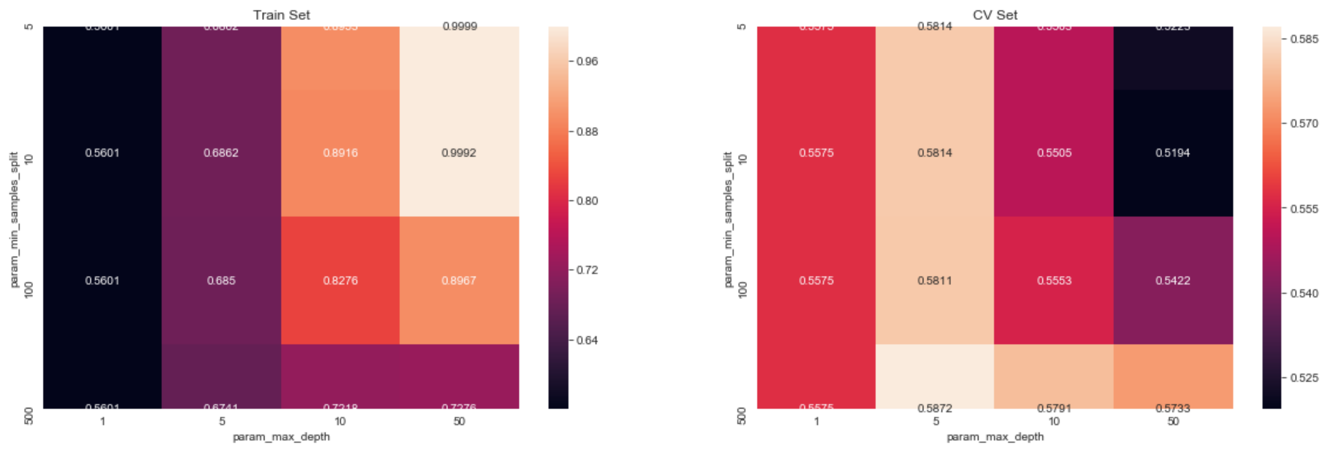
## printing the confusion matrix with predicted and original labels of test data points

In [87]:

```
def predict(proba, threshold, fpr, tpr):

    t = threshold[np.argmax(fpr*(1-tpr))]

    # (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 threshold", np.round(t,3))
    predictions = []
    for i in proba:
        if i>=t:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
```

In [88]:

```
%matplotlib inline
sns.set_style('whitegrid')
#max_score=pd.DataFrame(clf.cv_results_).groupby(['param_max_depth','param_min_samples_split'])

max_scores1 = pd.DataFrame(clf1.cv_results_).groupby(['param_min_samples_split', 'param_max_depth'
]).max().unstack()[['mean_test_score', 'mean_train_score']]
fig, ax = plt.subplots(1,2, figsize=(20,6))
sns.heatmap(max_scores1.mean_train_score, annot = True, fmt='.4g', ax=ax[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()
```



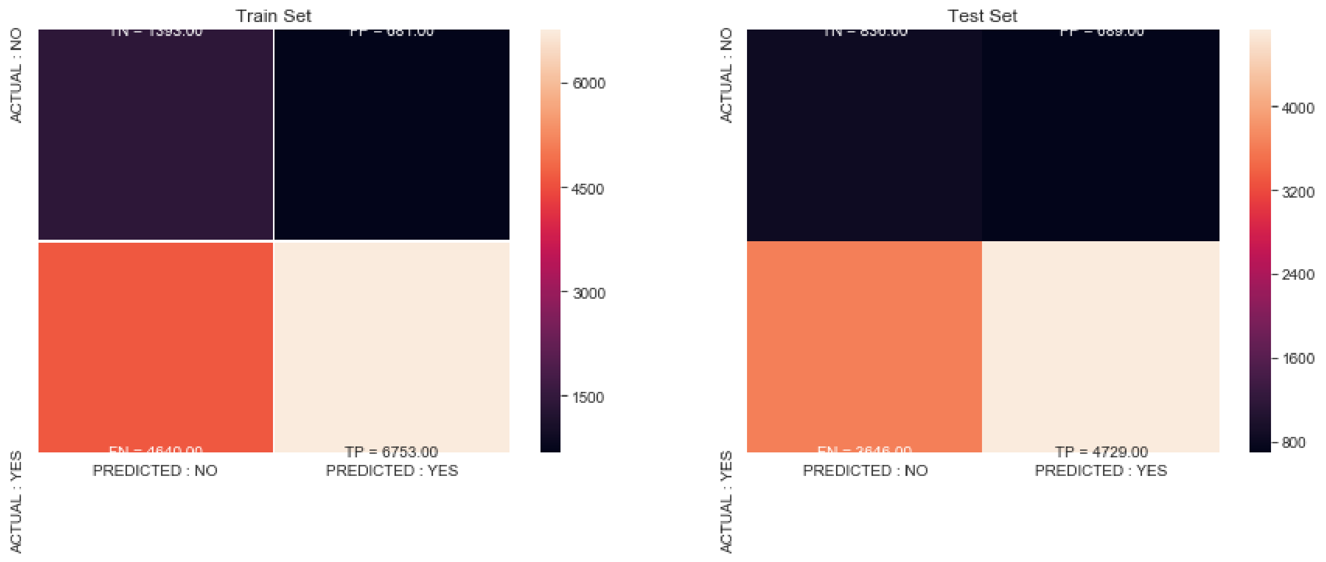
In [89]:

```
def predict(proba, threshold, fpr, tpr):
    t = threshold[np.argmax(fpr*(1-tpr))]
    print("the maximum value of tpr*(1-fpr)", np.round(max(tpr*(1-fpr)),2) , "for threshold", np.ro
und(t,2))
    predictions = []
    global predictions1 # making it global
    for i in proba:
        if i>=t:
            predictions.append(1)
        else:
            predictions.append(0)
    predictions1= predictions
    return predictions
```

In [90]:

```
conf_matr_df_train=confusion_matrix(y_train,predict(y_train_pred,tr_thresholds,train_fpr,train_tpr
))
conf_matr_df_test=confusion_matrix(y_test,predict(y_test_pred,te_thresholds,test_fpr, test_tpr))
embed=(np.asarray([['TN','FP'],['FN','TP']]))
fig,ax=plt.subplots(1,2,figsize=(15,5))
labels_train = (np.asarray(["{0} = {1:.2f}" .format(key, value) for key, value in zip(embed.flatten
(), conf_matr_df_train.flatten())])).reshape(2,2)
labels_test = (np.asarray(["{0} = {1:.2f}" .format(key, value) for key, value in zip(embed.flatten(
),conf_matr_df_test.flatten())])).reshape(2,2)

sns.heatmap(conf_matr_df_train,linewidths=0.5,xticklabels=['PREDICTED : NO', 'PREDICTED : YES'],yt
icklabels=['ACTUAL : NO', 'ACTUAL : YES'],annot=labels_train,fmt='',ax=ax[0])

sns.heatmap(conf_matr_df_test,xticklabels=['PREDICTED : NO', 'PREDICTED :
YES'],yticklabels=['ACTUAL : NO', 'ACTUAL : YES'],annot=labels_test,fmt='',ax=ax[1])
#print("train confusion matrix")
#print(confusion_matrix(y_train,predict(y_train_pred,tr_thresholds,train_fpr,train_tpr)))

ax[0].set_title('Train Set')
ax[1].set_title('Test Set')
plt.show()
```

```
the maximum value of tpr*(1-fpr) 0.4 for threshold 0.56
the maximum value of tpr*(1-fpr) 0.32 for threshold 0.56
```



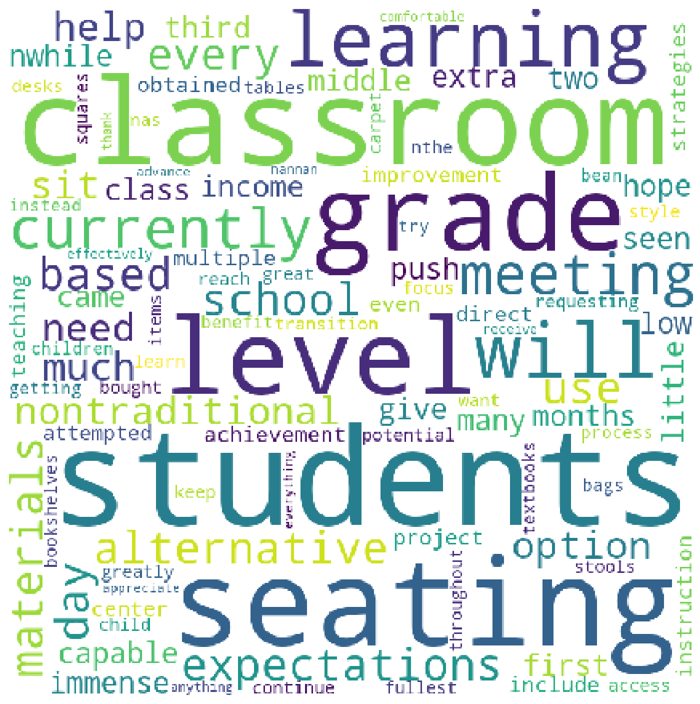
In [91]:

```
#Analysis on the False positives
fpi = []
for i in range(len(y_test)) :
  if (y_test.values[i] == 0) & (predictions1[i] == 1) :
    fpi.append(i)
fp_essay1 = []
```

```
for i in fpi :
  fp_essay1.append(X_test['essay'].values[i])
```

In [92]:

```
#WORD CLOUD OF ESSAY
from wordcloud import WordCloud, STOPWORDS
comment_words = ' '
stopwords = set(STOPWORDS)
for val in fp_essay1 :
  val = str(val)
  tokens = val.split()
for i in range(len(tokens)):
  tokens[i] = tokens[i].lower()
for words in tokens :
  comment_words = comment_words + words + ' '

wordcloud = WordCloud(width = 800, height = 800, background_color ='white', stopwords = stopwords,
min_font_size = 10).generate(comment_words)

plt.figure(figsize = (6, 6), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()
```

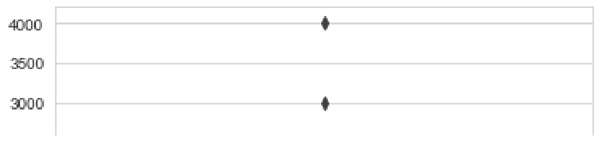


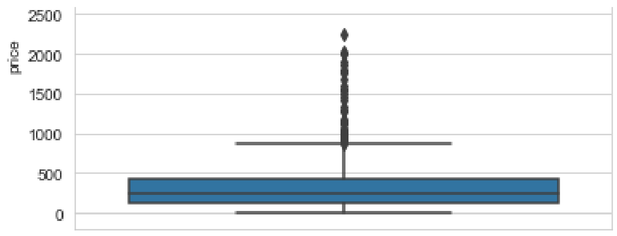
In [93]:

```
#Box Plot (FP 'price')
# first get the columns:
cols = X_test.columns
X_test_falsePos1 = pd.DataFrame(columns=cols)
# get the data of the false pisitives
for i in fpi : # (in fpi all the false positives data points indexes)
  X_test_falsePos1 = X_test_falsePos1.append(X_test.filter(items=[i], axis=0))
sns.boxplot(y='price', data=X_test_falsePos1)
```

Out[93]:

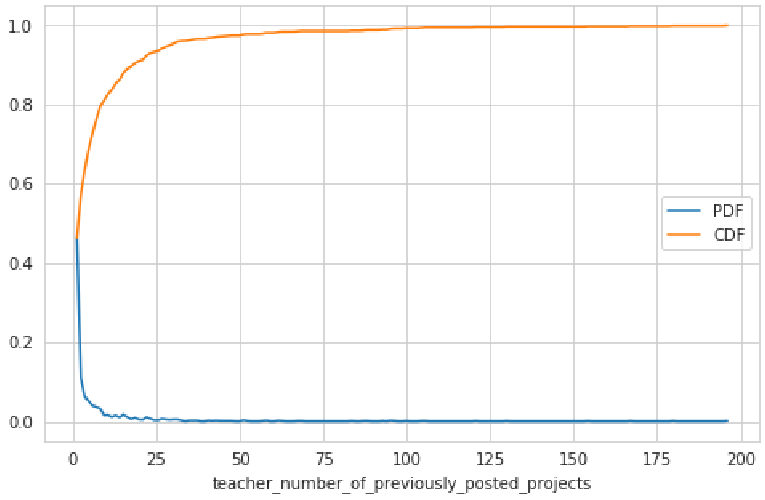
```
<matplotlib.axes._subplots.AxesSubplot at 0x13e38058f88>
```





In [0]:

```
#PDF (FP ,teacher_number_of_previously_posted_projects)
plt.figure(figsize=(8,5))
counts, bin_edges = np.histogram(X_test_falsePos1['teacher_number_of_previously_posted_projects'],
bins='auto', density=True)
pdf = counts/sum(counts)
cdf = np.cumsum(pdf)
pdfP, = plt.plot(bin_edges[1:], pdf)
cdfP, = plt.plot(bin_edges[1:], cdf)
plt.legend([pdfP, cdfP], ["PDF", "CDF"])
plt.xlabel('teacher_number_of_previously_posted_projects')
plt.show()
```



In [0]:

```

```

In [0]:

```

```

## categorical, numerical features + project_title(TFIDF) + preprocessed_essay (TFIDF)

In [94]:

```
X_tr1=hstack((X_train_state_ohe,X_train_teacher_ohe,X_train_grade_ohe,X_train_clean_subcategories_o
e,X_train_clean_categories_ohe,X_train_price_norm,X_train_No_of_teachers_norm,X_train_quantity_norm
,X_train_title_word_count_norm,X_train_essay_word_count_norm,
X_train_essay_tfidf,X_train_title_tfidf)).tocsr()

X_cr1=hstack((X_cv_state_ohe,X_cv_teacher_ohe,X_cv_grade_ohe,X_cv_clean_subcategories_ohe,X_cv_clea
n_categories_ohe,X_cv_price_norm,X_cv_No_of_teachers_norm,X_cv_quantity_norm,X_cv_title_word_count_
norm,X_cv_essay_word_count_norm,
X_train_essay_cv_tfidf,X_train_title_cv_tfidf)).tocsr()

X_te1=hstack((X_test_state_ohe,X_test_teacher_ohe,X_test_grade_ohe,X_clean_subcategories_grade_ohe
,X_test_clean_categories_ohe,X_test_price_norm,X_test_No_of_teachers_norm,X_test_quantity_norm,X_t
est_title_word_count_norm,X_test_essay_word_count_norm,
X_test_essay_tfidf,X_test_title_tfidf)).tocsr()
```

```
print("Final Data matrix")
print(X_tr1.shape, y_train.shape)
print(X_cr1.shape, y_cv.shape)
print(X_te1.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix
(13467, 5899) (13467,)
(6633, 5899) (6633,)
(9900, 5899) (9900,)
====================================================================================================
```

In [96]:

```
%%time
from sklearn.tree import DecisionTreeClassifier
tree=DecisionTreeClassifier(class_weight='balanced',random_state=0)
#The hyper parameter tuning (best `depth` in range [1, 5, 10, 50], and the best
`min_samples_split` in range [5, 10, 100, 500])
paramaters={'max_depth':[1, 5, 10, 50],'min_samples_split': [5, 10,100,500]}

clf=GridSearchCV(tree,paramaters,cv=10,scoring='roc_auc',return_train_score=True,verbose=2)
#clf=GridSearchCV(tree,paramaters,cv=10,scoring='roc_auc',n_jobs=-
1,return_train_score=True,verbose=2)
clf.fit(X_tr1,y_train)
```

```
Fitting 10 folds for each of 16 candidates, totalling 160 fits
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
```

```
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.1s remaining:    0.0s
```

```
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=5 ................................
[CV] ................. max_depth=1, min_samples_split=5, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
```

```
[CV] max_depth=1, min_samples_split=10 ...............................
[CV] ................ max_depth=1, min_samples_split=10, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=100 ..............................
[CV] ............... max_depth=1, min_samples_split=100, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=1, min_samples_split=500 ..............................
[CV] ............... max_depth=1, min_samples_split=500, total=   0.2s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=5 ................................
[CV] ................. max_depth=5, min_samples_split=5, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
```

```
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=10 ...............................
[CV] ................ max_depth=5, min_samples_split=10, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.9s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.8s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=100 ..............................
[CV] ............... max_depth=5, min_samples_split=100, total=   0.7s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.7s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.9s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=5, min_samples_split=500 ..............................
[CV] ............... max_depth=5, min_samples_split=500, total=   0.8s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   1.7s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   1.8s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   2.0s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   1.6s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   1.6s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   3.1s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   2.8s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   1.5s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   1.7s
[CV] max_depth=10, min_samples_split=5 ...............................
[CV] ................ max_depth=10, min_samples_split=5, total=   3.5s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] ............... max_depth=10, min_samples_split=10, total=   1.9s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] ............... max_depth=10, min_samples_split=10, total=   1.7s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] ............... max_depth=10, min_samples_split=10, total=   2.4s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] ............... max_depth=10, min_samples_split=10, total=   2.9s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] ............... max_depth=10, min_samples_split=10, total=   1.5s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] ............... max_depth=10, min_samples_split=10, total=   1.6s
```

```
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] .............. max_depth=10, min_samples_split=10, total=   2.9s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] .............. max_depth=10, min_samples_split=10, total=   2.4s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] .............. max_depth=10, min_samples_split=10, total=   1.7s
[CV] max_depth=10, min_samples_split=10 ..............................
[CV] .............. max_depth=10, min_samples_split=10, total=   1.6s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.5s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.8s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.7s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.4s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.5s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.4s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.7s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.6s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.7s
[CV] max_depth=10, min_samples_split=100 .............................
[CV] ............. max_depth=10, min_samples_split=100, total=   1.6s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   1.2s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   1.2s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   2.1s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   2.4s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   1.1s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   1.2s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   1.1s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   2.3s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   2.3s
[CV] max_depth=10, min_samples_split=500 .............................
[CV] ............. max_depth=10, min_samples_split=500, total=   1.3s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   9.1s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   8.4s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   6.0s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   6.4s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   6.5s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   6.5s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   6.8s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   8.5s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   8.4s
[CV] max_depth=50, min_samples_split=5 ...............................
[CV] ................ max_depth=50, min_samples_split=5, total=   6.7s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] .............. max_depth=50, min_samples_split=10, total=   6.7s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] .............. max_depth=50, min_samples_split=10, total=   6.5s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] .............. max_depth=50, min_samples_split=10, total=   5.7s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] .............. max_depth=50, min_samples_split=10, total=   8.5s
[CV] max_depth=50, min_samples_split=10 ..............................
```

```
[CV] ............... max_depth=50, min_samples_split=10, total=   6.3s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] ............... max_depth=50, min_samples_split=10, total=   8.3s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] ............... max_depth=50, min_samples_split=10, total=   6.1s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] ............... max_depth=50, min_samples_split=10, total=   8.3s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] ............... max_depth=50, min_samples_split=10, total=   7.9s
[CV] max_depth=50, min_samples_split=10 ..............................
[CV] ............... max_depth=50, min_samples_split=10, total=   8.3s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   6.0s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   5.5s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   4.5s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   4.7s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   4.5s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   6.7s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   5.1s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   5.9s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   4.1s
[CV] max_depth=50, min_samples_split=100 .............................
[CV] .............. max_depth=50, min_samples_split=100, total=   6.7s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   5.0s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   2.5s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   3.4s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   4.2s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   2.4s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   5.2s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   3.0s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   4.7s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   2.3s
[CV] max_depth=50, min_samples_split=500 .............................
[CV] .............. max_depth=50, min_samples_split=500, total=   3.3s
```

```
[Parallel(n_jobs=1)]: Done 160 out of 160 | elapsed:  5.8min finished
```

```
Wall time: 5min 52s
```

Out[96]:

```
GridSearchCV(cv=10, error_score=nan,
             estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                              class_weight='balanced',
                                              criterion='gini', max_depth=None,
                                              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,
                                              presort='deprecated',
                                              random_state=0, splitter='best'),
             iid='deprecated', n_jobs=None,
             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=True,
             scoring='roc_auc', verbose=2)
```

In [98]:

```
print(clf.best_estimator_)
print("Using the best parametrs predict the best score for Test", clf.score(X_tr1,y_train))
print("Using the best parametrs predict the best score for Test",clf.score(X_te1,y_test))
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight='balanced', criterion='gini',
                       max_depth=10, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=500,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=0, splitter='best')
Using the best parametrs predict the best score for Test 0.6989715046907027
Using the best parametrs predict the best score for Test 0.5886741766576952
```

In [99]:

```
train_auc=clf.cv_results_['mean_train_score']
train_auc_std=clf.cv_results_['std_train_score']
cv_auc=clf.cv_results_['mean_test_score']
cv_auc_std=clf.cv_results_['std_test_score']
```

In [100]:

```
clf.cv_results_
```

Out[100]:

```
{'mean_fit_time': array([0.15688035, 0.1775243 , 0.17912085, 0.18041761, 0.69765182,
        0.70634391, 0.73107138, 0.77364094, 2.12494173, 2.05691297,
        1.58818433, 1.62448711, 7.32946877, 7.27162879, 5.36430733,
        3.58951762]),
 'std_fit_time': array([0.00571235, 0.02233706, 0.01608277, 0.02004333, 0.01507851,
        0.008465  , 0.06818632, 0.05288474, 0.69370447, 0.53344012,
        0.11172908, 0.54875224, 1.06535076, 1.04938946, 0.88087677,
        1.05840207]),
 'mean_score_time': array([0.00249383, 0.00269418, 0.00289257, 0.00269322, 0.00259745,
        0.00239351, 0.00249524, 0.00319893, 0.00318468, 0.00299482,
        0.00269005, 0.00299389, 0.0025008 , 0.00290749, 0.00320423,
        0.00311365]),
 'std_score_time': array([0.00066905, 0.00063715, 0.0008289 , 0.00045683, 0.00049255,
        0.00048818, 0.00050271, 0.0006006 , 0.00153043, 0.00109281,
        0.00077511, 0.0010949 , 0.00050589, 0.00112266, 0.00124605,
        0.00104889]),
 'param_max_depth': masked_array(data=[1, 1, 1, 1, 5, 5, 5, 5, 10, 10, 10, 10, 50, 50, 50, 50],
              mask=[False, False, False, False, False, False, False, False,
                    False, False, False, False, False, False, False, False],
        fill_value='?',
             dtype=object),
 'param_min_samples_split': masked_array(data=[5, 10, 100, 500, 5, 10, 100, 500, 5, 10, 100, 500,
5,
                    10, 100, 500],
              mask=[False, False, False, False, False, False, False, False,
                    False, False, False, False, False, False, False, False],
        fill_value='?',
             dtype=object),
 'params': [{'max_depth': 1, 'min_samples_split': 5},
  {'max_depth': 1, 'min_samples_split': 10},
  {'max_depth': 1, 'min_samples_split': 100},
  {'max_depth': 1, 'min_samples_split': 500},
  {'max_depth': 5, 'min_samples_split': 5},
  {'max_depth': 5, 'min_samples_split': 10},
  {'max_depth': 5, 'min_samples_split': 100},
  {'max_depth': 5, 'min_samples_split': 500},
  {'max_depth': 10, 'min_samples_split': 5},
  {'max_depth': 10, 'min_samples_split': 10},
  {'max_depth': 10, 'min_samples_split': 100},
  {'max_depth': 10, 'min_samples_split': 500},
  {'max_depth': 50, 'min_samples_split': 5},
  {'max_depth': 50, 'min_samples_split': 10},
  {'max_depth': 50, 'min_samples_split': 100},
  {'max_depth': 50, 'min_samples_split': 500}],
 'split0_test_score': array([0.56709891, 0.56709891, 0.56709891, 0.56709891, 0.5761823 ,
        0.5801212 , 0.58097084, 0.59361387, 0.55453004, 0.55182643,
        0.56147767, 0.60013772, 0.49974362, 0.53023985, 0.53173786,
        0.57015213]),
 'split1_test_score': array([0.56137808, 0.56137808, 0.56137808, 0.56137808, 0.59688745,
        0.59688745, 0.59809306, 0.61057505, 0.58302822, 0.5823841 ,
        0.59554411, 0.61967116, 0.54836003, 0.55458725, 0.56863505,
        0.6081829 ]),
 'split2_test_score': array([0.54270913, 0.54270913, 0.54270913, 0.54270913, 0.57220315,
        0.57220315, 0.57147852, 0.57245317, 0.57546826, 0.57130477,
        0.56899102, 0.56897195, 0.51162599, 0.52200822, 0.53363844,
        0.53522121]),
 'split3_test_score': array([0.53162356, 0.53162356, 0.53162356, 0.53162356, 0.57551538,
        0.57559136, 0.57592271, 0.57449602, 0.60286731, 0.5846538 ,
        0.61202261, 0.61444123, 0.52838396, 0.54373776, 0.56952582,
        0.58549377]),
 'split4_test_score': array([0.53105795, 0.53105795, 0.53105795, 0.53105795, 0.58629364,
        0.58629364, 0.5865469 , 0.58913858, 0.58564784, 0.57977646,
        0.57853338, 0.60613012, 0.5440269 , 0.54587568, 0.55362751,
        0.58989625]),
 'split5_test_score': array([0.53659798, 0.53659798, 0.53659798, 0.53659798, 0.56549267,
        0.56549267, 0.5609108 , 0.56732247, 0.56493761, 0.56422638,
        0.56786697, 0.59472716, 0.51733555, 0.51832537, 0.56606884,
        0.60156725]),
 'split6_test_score': array([0.54903508, 0.54903508, 0.54903508, 0.54903508, 0.59804273,
        0.5971711 , 0.59492765, 0.58935174, 0.55484315, 0.55334681,
        0.56016791, 0.56179932, 0.50697727, 0.49845723, 0.54594533,
        0.58309414]),
 'split7_test_score': array([0.54329588, 0.54329588, 0.54329588, 0.54329588, 0.57912484,
        0.57912484, 0.57891701, 0.58020214, 0.56441789, 0.56676549,
        0.57017131, 0.58892664, 0.50225217, 0.52430304, 0.52638343,
        0.55761262]),
 'split8_test_score': array([0.54791049, 0.54791049, 0.54791049, 0.54791049, 0.57869434,
        0.57869434, 0.58040785, 0.57453992, 0.55684069, 0.56175644,
        0.56810788, 0.56910885, 0.51479601, 0.533475  , 0.513636  ,
        0.53482375]),
 'split9_test_score': array([0.56657463, 0.56657463, 0.56657463, 0.56657463, 0.59373847,
        0.59373847, 0.59386359, 0.59845275, 0.5782193 , 0.5747944 ,
        0.56777282, 0.58115433, 0.52782337, 0.52798667, 0.52032039,
        0.53569323]),
 'mean_test_score': array([0.54772817, 0.54772817, 0.54772817, 0.54772817, 0.5822175 ,
        0.58253182, 0.58220389, 0.58501457, 0.57208003, 0.56908351,
        0.57506557, 0.59050685, 0.52013249, 0.52989961, 0.54295187,
        0.57017372]),
 'std_test_score': array([0.01274538, 0.01274538, 0.01274538, 0.01274538, 0.01048007,
        0.01018559, 0.01092063, 0.01287636, 0.01499349, 0.01095139,
        0.01555011, 0.01900994, 0.01584963, 0.01512927, 0.01970212,
        0.02653084]),
 'rank_test_score': array([10, 10, 10, 10,  4,  3,  5,  2,  7,  9,  6,  1, 16, 15, 14,  8]),
 'split0_train_score': array([0.55090456, 0.55090456, 0.55090456, 0.55090456, 0.62579395,
        0.62540361, 0.62297849, 0.61633892, 0.73125976, 0.72790422,
        0.71112496, 0.6789025 , 0.9917746 , 0.98670208, 0.91956087,
        0.79576884]),
 'split1_train_score': array([0.55218513, 0.55218513, 0.55218513, 0.55218513, 0.64482811,
        0.64434593, 0.6414891 , 0.63588809, 0.76926766, 0.76616843,
        0.74610992, 0.69756918, 0.99936604, 0.99737054, 0.94732622,
        0.78519216]),
 'split2_train_score': array([0.55427964, 0.55427964, 0.55427964, 0.55427964, 0.64874772,
        0.64874772, 0.64636029, 0.6347918 , 0.7802091 , 0.77761352,
        0.74821024, 0.70274647, 0.99021445, 0.98535425, 0.92936048,
        0.81197526]),
 'split3_train_score': array([0.55026334, 0.55026334, 0.55026334, 0.55026334, 0.62307489,
        0.62239714, 0.61756693, 0.61281734, 0.74441318, 0.74163971,
        0.72083189, 0.68884158, 0.99189794, 0.98668877, 0.91706729,
        0.81197809]),
 'split4_train_score': array([0.55490619, 0.55490619, 0.55490619, 0.55490619, 0.63506841,
        0.63506841, 0.63162709, 0.62526633, 0.76267683, 0.75997974,
        0.73700635, 0.68987156, 0.99435394, 0.99061663, 0.93225601,
        0.7957879 ]),
 'split5_train_score': array([0.55164764, 0.55164764, 0.55164764, 0.55164764, 0.64888967,
        0.64888967, 0.64168726, 0.63750211, 0.74937431, 0.74773359,
        0.72421669, 0.69933245, 0.98863505, 0.98407923, 0.936349  ,
        0.82163482]),
 'split6_train_score': array([0.55290744, 0.55290744, 0.55290744, 0.55290744, 0.64563456,
        0.64524594, 0.64098555, 0.63592931, 0.76117738, 0.75942139,
        0.73143636, 0.6922997 , 0.99384343, 0.98999039, 0.92833226,
        0.80333809]),
 'split7_train_score': array([0.55421556, 0.55421556, 0.55421556, 0.55421556, 0.63930368,
        0.63930368, 0.63364505, 0.63016656, 0.75658601, 0.75380632,
        0.72954453, 0.70597869, 0.99270704, 0.9890783 , 0.92946719,
        0.79066254]),
 'split8_train_score': array([0.55370546, 0.55370546, 0.55370546, 0.55370546, 0.64860889,
        0.64860889, 0.6460793 , 0.63654031, 0.7876533 , 0.78663547,
        0.76139109, 0.69660787, 0.99403415, 0.99052638, 0.93324063,
        0.79036138]),
 'split9_train_score': array([0.55096066, 0.55096066, 0.55096066, 0.55096066, 0.64247592,
        0.64247592, 0.63961223, 0.6337526 , 0.75766401, 0.75509969,
        0.73742808, 0.69550114, 0.99333828, 0.98842371, 0.91797416,
        0.8169862 ]),
 'mean_train_score': array([0.55259756, 0.55259756, 0.55259756, 0.55259756, 0.64024258,
        0.64004869, 0.63620313, 0.62989934, 0.76002815, 0.75760021,
        0.73473001, 0.69476511, 0.99301649, 0.98888303, 0.92909341,
        0.80236853]),
 'std_train_score': array([0.00155458, 0.00155458, 0.00155458, 0.00155458, 0.00896292,
        0.00910986, 0.00918096, 0.00842692, 0.01570514, 0.01606412,
        0.01390366, 0.00734059, 0.00271783, 0.00352589, 0.00878752,
        0.01197301])}
```

In [0]:

```

```

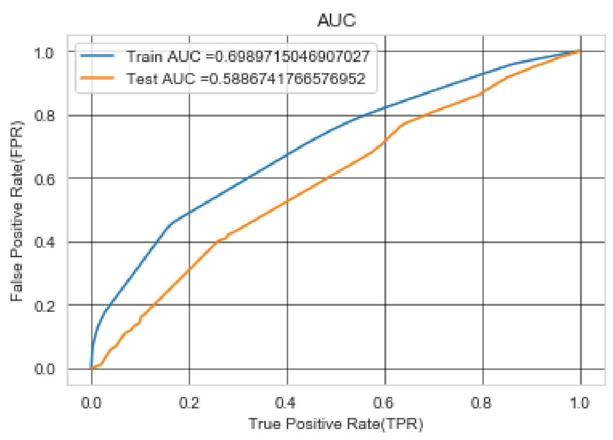
In [101]:

```
%%time
from sklearn.metrics import roc_curve
#{'max_depth': 10, 'min_samples_split': 500}

tree_tfidf=DecisionTreeClassifier(class_weight='balanced',max_depth=10,min_samples_split=500, rand
om_state=0)
tree_tfidf.fit(X_tr1,y_train)
y_train_pred = batch_predict(tree_tfidf, X_tr1)
y_test_pred = batch_predict(tree_tfidf, X_te1)

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
```

```
Wall time: 1.4 s
```

In [102]:

```
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("True Positive Rate(TPR)")
plt.ylabel("False Positive Rate(FPR)")
plt.title("AUC")
plt.grid(color='black', linestyle='-', linewidth=0.5)
plt.show()
```



[illegible]

From given plot we observe that at {'max_depth': 10, 'min_samples_split': 500} we get train AUC of 0.6617 and test AUC of 0.6224

## Representation of results

3D plot with X-axis as min_sample_split, Y-axis as max_depth, and Z-axis as AUC Score

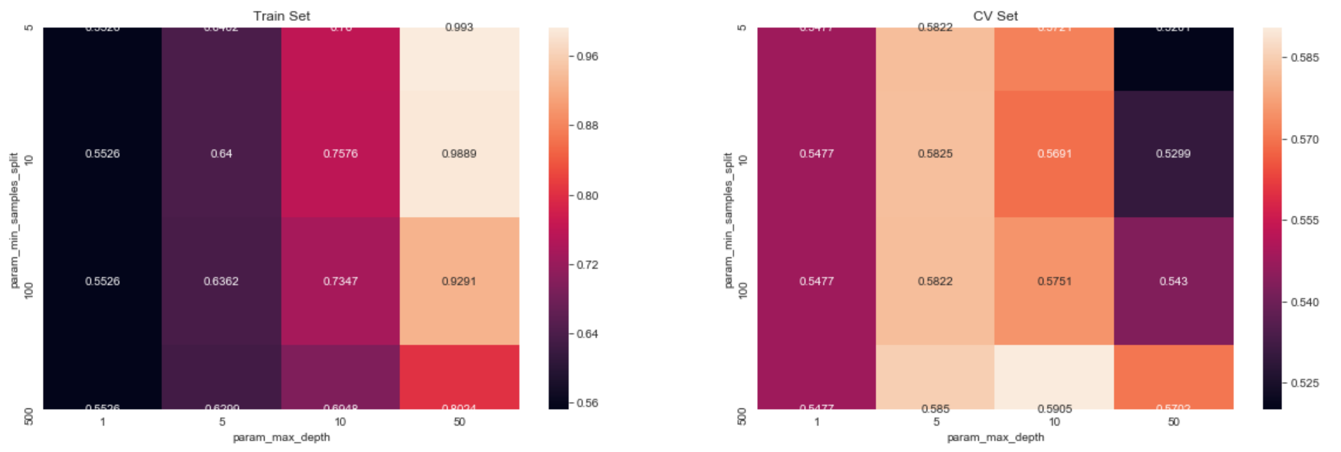
In [103]:

```
paramaters={'max_depth':[1, 5, 10, 50],'min_samples_split': [5, 10,100,500]}
trace1 = go.Scatter3d(x=paramaters['min_samples_split'], y=paramaters['max_depth'],z= train_auc, na
me = 'train')
trace2 = go.Scatter3d(x=paramaters['min_samples_split'], y=paramaters['max_depth'],z= cv_auc, name
= 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='min_sample_split'),
        yaxis = dict(title='max_depth'),
        zaxis = dict(title='AUC'),))

fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

## seaborn heat maps with rows as n_estimators, columns as max_depth, and values inside the cell representing AUC Score

In [104]:

```
%matplotlib inline
sns.set_style('whitegrid')
#max_score=pd.DataFrame(clf.cv_results_).groupby(['param_max_depth','param_min_samples_split'])

max_scores1 = pd.DataFrame(clf.cv_results_).groupby(['param_min_samples_split', 'param_max_depth']
).max().unstack()[['mean_test_score', 'mean_train_score']]
fig, ax = plt.subplots(1,2, figsize=(20,6))
sns.heatmap(max_scores1.mean_train_score, annot = True, fmt='.4g', ax=ax[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()
```

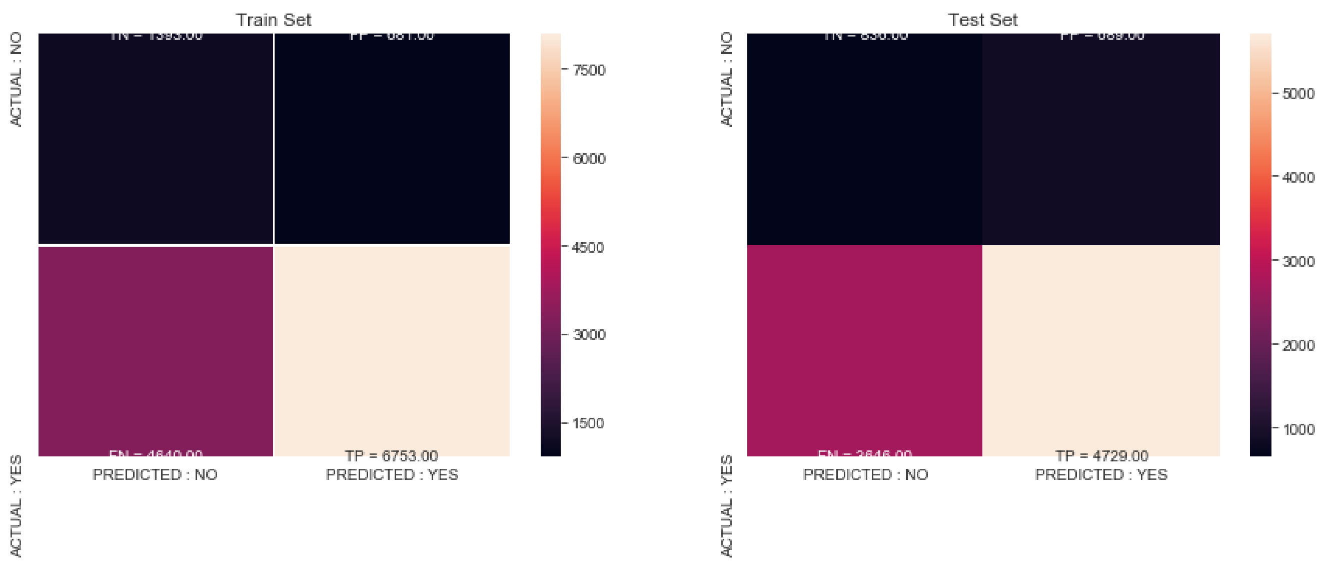


In [ ]:

In [105]:

```
conf_matr_df_train_1=confusion_matrix(y_train,predict(y_train_pred,tr_thresholds,train_fpr,train_tp
r))
conf_matr_df_test_1=confusion_matrix(y_test,predict(y_test_pred,te_thresholds,test_fpr, test_tpr))
embed=(np.asarray([['TN','FP'],['FN','TP']]))
fig,ax=plt.subplots(1,2,figsize=(15,5))
labels_train = (np.asarray(["{0} = {1:.2f}" .format(key, value) for key, value in zip(embed.flatten
(), conf_matr_df_train.flatten())])).reshape(2,2)
labels_test = (np.asarray(["{0} = {1:.2f}" .format(key, value) for key, value in zip(embed.flatten(
),conf_matr_df_test.flatten())])).reshape(2,2)

sns.heatmap(conf_matr_df_train_1,linewidths=0.5,xticklabels=['PREDICTED : NO', 'PREDICTED : YES'],
yticklabels=['ACTUAL : NO', 'ACTUAL : YES'],annot=labels_train,fmt='',ax=ax[0])

sns.heatmap(conf_matr_df_test_1,xticklabels=['PREDICTED : NO', 'PREDICTED :
YES'],yticklabels=['ACTUAL : NO', 'ACTUAL : YES'],annot=labels_test,fmt='',ax=ax[1])
#print("train confusion matrix")
#print(confusion_matrix(y_train,predict(y_train_pred,tr_thresholds,train_fpr,train_tpr)))

ax[0].set_title('Train Set')
ax[1].set_title('Test Set')
plt.show()
```

```
the maximum value of tpr*(1-fpr) 0.4 for threshold 0.48
the maximum value of tpr*(1-fpr) 0.3 for threshold 0.48
```



In [106]:

```
#Feature aggregation
f1=vec_state.get_feature_names()
f2=vec_prefix.get_feature_names()
f3=vec_grade_cat.get_feature_names()
f4=vec_clean_sub_cat.get_feature_names()
f5=vec_clean_cat.get_feature_names()

fb1=vec_tfidf_title.get_feature_names()
ft1=vec_tfidf_essay.get_feature_names()
feature_agg_tfidf = f1 + f2 + f3 + f4 + f5 + fb1 + ft1
# p is price, q is quantity, t is teacher previous year projects

feature_agg_tfidf.append('price')
feature_agg_tfidf.append('quantity')
feature_agg_tfidf.append('teacher_previous_projects')
feature_agg_tfidf.append('title_word_count')
feature_agg_tfidf.append('essay_word_count')
```

In [0]:

```
# this model is created just to visualize the tree
clfVisual=DecisionTreeClassifier(class_weight = 'balanced',max_depth=3,min_samples_split=500)
clfVisual.fit(X_tr1, y_train)
```

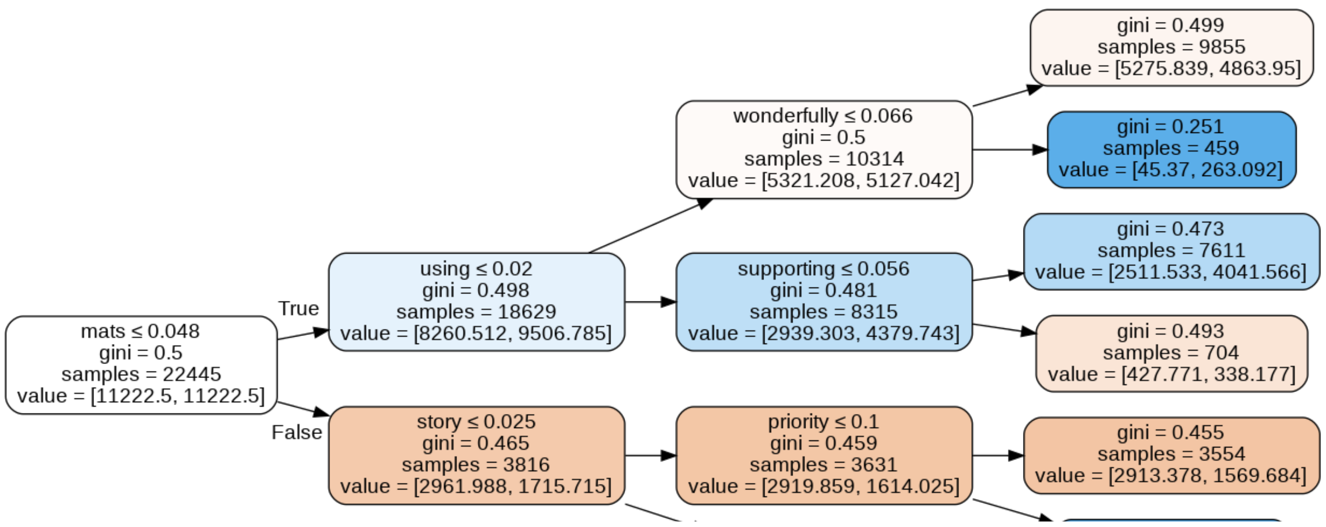
Out[0]:

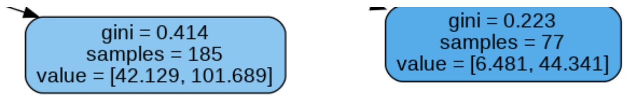
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight='balanced', criterion='gini',
                       max_depth=3, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=500,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

In [0]:

```
# https://scikit-
learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTr
assifier
# the code implimented here can be found in the SKlearn Documentation
import warnings
warnings.filterwarnings("ignore")
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(clfVisual, out_file=dot_data, filled=True, rounded=True, special_characters=True, f
eature_names=feature_agg_tfidf,rotate=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

Out[0]:





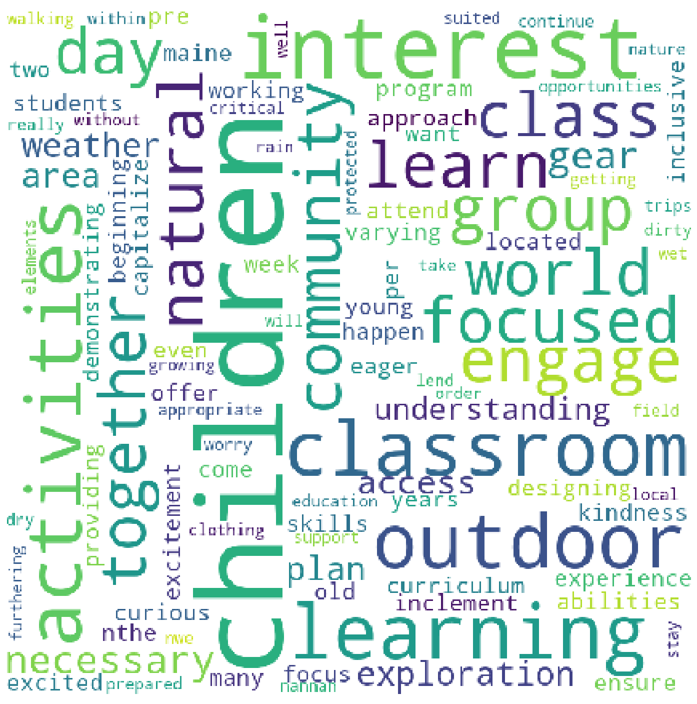
Analysis of the false positive rate

In [107]:

```
# The code for this taken as a reference from the links given below .
#https://www.google.com/search?
q=geeks+for+geeks+false+positive&rlz=1C1SQJL_enIN849IN849&oq=geeks+for+geeks+false+positive&aqs=chi
.69i57j3315.6431j0j7&sourceid=chrome&ie=UTF-8
#https://github.com/pskadasi/DecisionTrees_DonorsChoose/blob/master/Copy_of_8_DonorsChoose_DT_(1).:

fpi = []
for i in range(len(y_test)):
  if (y_test.values[i] == 0) & (predictions1[i] == 1) :
    fpi.append(i)
fp_essay1 = []
for i in fpi :
  fp_essay1.append(X_test['essay'].values[i])
```

In [108]:

```
from wordcloud import WordCloud, STOPWORDS
comment_words = ' '
stopwords = set(STOPWORDS)
for val in fp_essay1 :
  val = str(val)
  tokens = val.split()
for i in range(len(tokens)):
  tokens[i] = tokens[i].lower()
for words in tokens :
  comment_words = comment_words + words + ' '
wordcloud = WordCloud(width = 800, height = 800, background_color ='white', stopwords = stopwords,m
in_font_size = 10).generate(comment_words)

plt.figure(figsize = (6, 6), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()
```



In [109]:

```
# create the DataFrame for False Positive
# https://www.geeksforgeeks.org/python-pandas-dataframe-filter
cols = X_test.columns
X_test_falsePos1 = pd.DataFrame(columns=cols)
for i in fpi :
  X_test_falsePos1 = X_test_falsePos1.append(X_test.filter(items=[i], axis=0))


print(X_test_falsePos1.head(10))
print(len(X_test_falsePos1))
```

```
    Unnamed: 0       id                        teacher_id teacher_prefix  \
6       155477  p134986  9cb8846b548404438c81aaa02eda4f0f            Mrs
43       74901  p185850  106d8124b4fac334dedff8fe5387ee68            Mrs
48       25829  p165533  8a86f0c8089b3f9b4547e425a194e3d9            Mrs
64      136729  p130983  950a34ec96979e36e7ff62c29af80255            Mrs
67      104230  p127231  ad15fbaf2466bbbddc641969812e6a9d            Mrs
69       49359  p032480  a4186153269d371d738d29f35c735ef1             Mr
89      145757  p059301  cfa5bab6882bd6cb4da3899fa99e7b75             Ms
93       44751  p016650  2fd1fea2cbfeb473ffbcea8eb78cc38d            Mrs
99       89280  p002492  98e4590e4e65e7ceb58b3925f0fb5dc6            Mrs
106      49203  p160009  3b8b1099b4c6e22592b3a2c12ff12d05             Ms

    school_state project_grade_category  \
6             OH                PreK_2
43            CA                PreK_2
48            FL                   3_5
64            MI                   3_5
67            MD                PreK_2
69            CA                   3_5
89            CA                   3_5
93            CA                   3_5
99            NC                  9_12
106           NY                   3_5

                                         project_title  \
6    Kindergarten Emergent Writers Exploring and Ex...
43                          Ribbons and Rulers Please!
48                                     Fun in Learning
64                      \"Keep Calm\" Classroom Caddies
67                          Tools for Our New Gadjects
69   America's Right To Fight: Revolutionary Perspe...
89                             Let Us Stand and Learn!
93                             It's All in the Weather
99        Integrating Mindfulness - A Space to Breathe
106              My Classroom: \"Fixer Upper\" Edition

                                       project_essay_1  \
6    My students come from such varied backgrounds,...
43   I have a kindergarten class with 24 students. ...
48   You ask me to describe my students. Well, I ca...
64   Our students are loving, creative, and demonst...
67   My students come from all of the world! In my ...
69   Many hold true to our mission which is \"to in...
89   My students come into my room excited to learn...
93   Our school is set in a suburban community, in ...
99   Creative, intelligent, brave, endearing, and r...
106  I am fortunate enough to teach a bilingual cla...

                                       project_essay_2 project_essay_3  ...  \
6    I have been privileged to have worked in a pro...             NaN  ...
43   My class is requesting a variety of ribbons an...             NaN  ...
48   My theory behind my project is that my student...             NaN  ...
64   With a Calm Caddy in every classroom, students...             NaN  ...
67   Earlier this year our class got some new Chrom...             NaN  ...
69   No child should be deprived of having a real, ...             NaN  ...
89   Teachers are now encouraged to have multiple a...             NaN  ...
93   This year in the science lab, we are focusing ...             NaN  ...
99   As our school increases in size, we are seeing...             NaN  ...
106  Do you ever look around when you are ordering ...             NaN  ...

                              project_resource_summary  \
6    My students need materials to launch our Write...
43   My students need rulers and ribbons to integr...
48   My students need more fun. I am looking to mak...
64   My students need coping tools like stress ball...
67   My students need headphones, rechargeable batt...
```

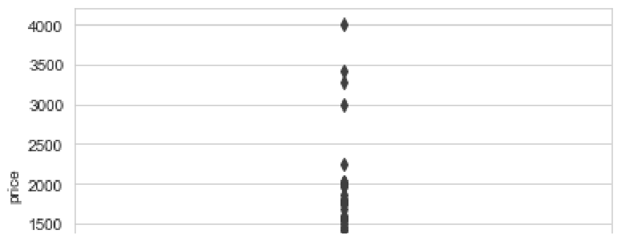
```
67   My students need headphones, rechargeable batt...
69   My students need class sets of  empowering tex...
89   My students need a standing desk to help provi...
93   My students need weather instruments and other...
99   My students need a mindful/calming/meditation ...
106  My students need a newly designed setting cond...

    teacher_number_of_previously_posted_projects                date  \
6                                              0 2016-07-19 21:21:12
43                                            61 2016-11-20 20:20:14
48                                             2 2017-02-26 20:37:36
64                                             0 2016-09-02 15:52:07
67                                            17 2016-12-16 15:08:46
69                                             0 2017-02-08 16:47:32
89                                             6 2016-08-09 15:17:49
93                                            22 2016-09-29 17:45:51
99                                             2 2016-08-27 13:48:04
106                                            7 2016-08-10 18:20:26

                   clean_categories                   clean_subcategories  \
6                 literacy_language           literacy literature_writing
43           math_science music_arts               mathematics visualarts
48                literacy_language                    literature_writing
64     health_sports appliedlearning                 health_wellness other
67     literacy_language math_science                literacy mathematics
69                literacy_language                    literature_writing
89                  appliedlearning                                 other
93                     math_science  appliedsciences environmentalscience
99                     specialneeds                          specialneeds
106               literacy_language                          esl literacy

    title_word_count                                              essay  \
6                  9  My students come from such varied backgrounds,...
43                 4  I have a kindergarten class with 24 students. ...
48                 3  You ask me to describe my students. Well, I ca...
64                 4  Our students are loving, creative, and demonst...
67                 5  My students come from all of the world! In my ...
69                 6  Many hold true to our mission which is \"to in...
89                 5  My students come into my room excited to learn...
93                 5  Our school is set in a suburban community, in ...
99                 7  Creative, intelligent, brave, endearing, and r...
106                5  I am fortunate enough to teach a bilingual cla...

    essay_word_count   price  quantity
6                427  160.96         8
43               198  152.28        14
48               402  411.83        10
64               327  102.48        59
67               212  121.64        17
69               189   28.58        65
89               239  289.98         3
93               284  421.05        13
99               213  414.02        19
106              483  128.30        10

[10 rows x 21 columns]
875
```

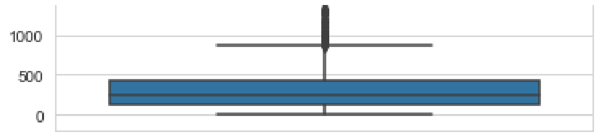
In [112]:

```python
# https://seaborn.pydata.org/generated/seaborn.boxplot.html
sns.boxplot(y='price', data=X_test_falsePos1)
```

Out[112]:

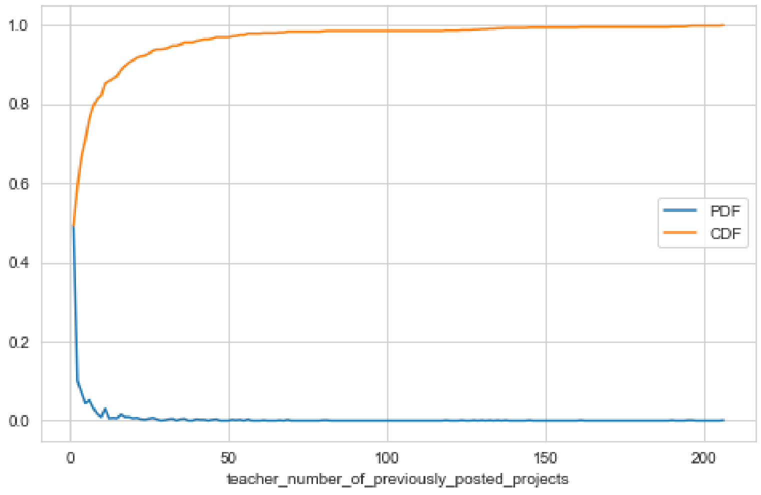
```
<matplotlib.axes._subplots.AxesSubplot at 0x13e5e240d88>
```





In [113]:

```
##PDF (FP ,teacher_number_of_previously_posted_projects)
plt.figure(figsize=(8,5))
counts, bin_edges = np.histogram(X_test_falsePos1['teacher_number_of_previously_posted_projects'],
bins='auto', density=True)
pdf = counts/sum(counts)
cdf = np.cumsum(pdf)
pdfP, = plt.plot(bin_edges[1:], pdf)
cdfP, = plt.plot(bin_edges[1:], cdf)
plt.legend([pdfP, cdfP], ["PDF", "CDF"])
plt.xlabel('teacher_number_of_previously_posted_projects')
plt.show()
print("The plot above shows that the projects submetted by the teachers are low and very few have
projects with high submission rate ")
```



```
The plot above shows that the projects submetted by the teachers are low and very few have project
s with high submission rate
```

Task 2: For this task consider set-1 features. Select all the features which are having non-zero feature importance.You can get the feature importance using 'feature*importances*` (https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html), discard the all other remaining features and then apply any of the model of you choice i.e. (Dession tree, Logistic Regression, Linear SVM), you need to do hyperparameter tuning corresponding to the model you selected and procedure in step 2 and step 3 Note: when you want to find the feature importance make sure you don't use max_depth parameter keep it None.

# The below references give very useful information about the extraction of features.

https://stackoverflow.com/questions/47111434/randomforestregressor-and-feature-importances-error
https://stackoverflow.com/questions/48377296/get-feature-importance-from-gridsearchcv
https://datascience.stackexchange.com/questions/31406/tree-decisiontree-feature-importances-numbers-correspond-to-how-features

In [114]:

```
from sklearn.feature_selection import SelectKBest
```

In [115]:

```
def selectKFeatures(model, X, k=5):
    return X[:,model.best_estimator_.feature_importances_.argsort()[::-1][:k]]
```

In [116]:

```
X_tr1=X_tr1.tocsr()
X_set_train = selectKFeatures(clf, X_tr1,5000)
X_set_test = selectKFeatures(clf, X_te1, 5000)

print(X_set_train.shape)
print(X_set_test.shape)
```

```
(13467, 5000)
(9900, 5000)
```

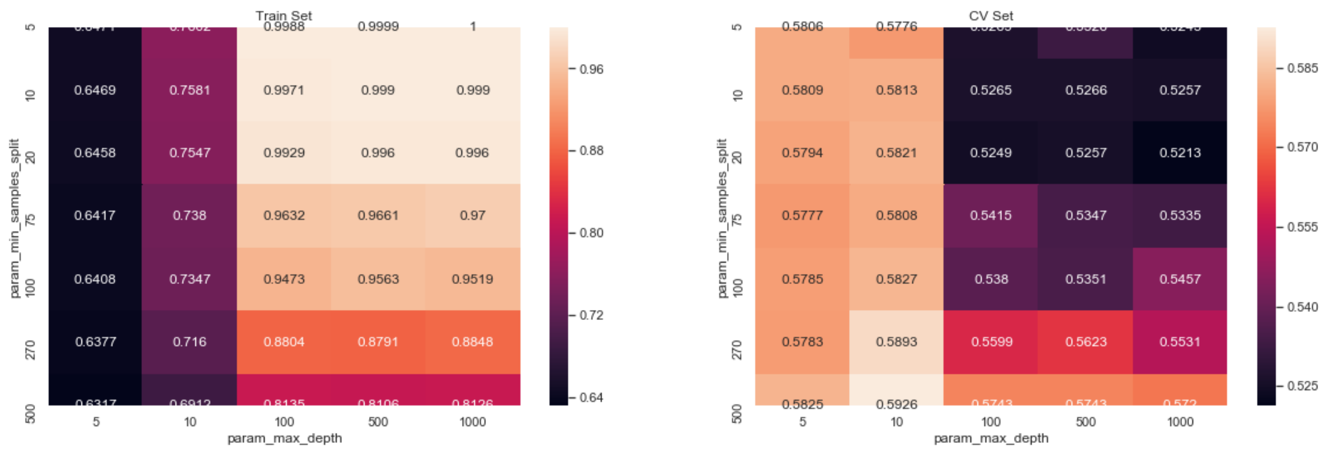
Applying Decision trees on the Important features selected .

Not applying any general linear model as the no. of data points are low

As the distance between the Auc-Roc curve between the train and test are wider from here we know that it is a high variance model , Decision trees work well on high variance and low bias model.

In [121]:

```
from sklearn.metrics import roc_auc_score
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
dt5= DecisionTreeClassifier(class_weight = 'balanced')
parameters = {'max_depth': [5, 10, 100, 500, 1000], 'min_samples_split': [5, 10, 20,75, 100,270,
500]}
clf5 = GridSearchCV(dt5, parameters, cv=3,n_jobs=-1, scoring='roc_auc',return_train_score=True)
set5= clf5.fit(X_set_train, y_train)


import seaborn as sns; sns.set()
max_scores1 = pd.DataFrame(clf5.cv_results_).groupby(['param_min_samples_split', 'param_max_depth'
]).max().unstack()[['mean_test_score', 'mean_train_score']]
fig, ax = plt.subplots(1,2, figsize=(20,6))
sns.heatmap(max_scores1.mean_train_score, annot = True, fmt='.4g', ax=ax[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()
```



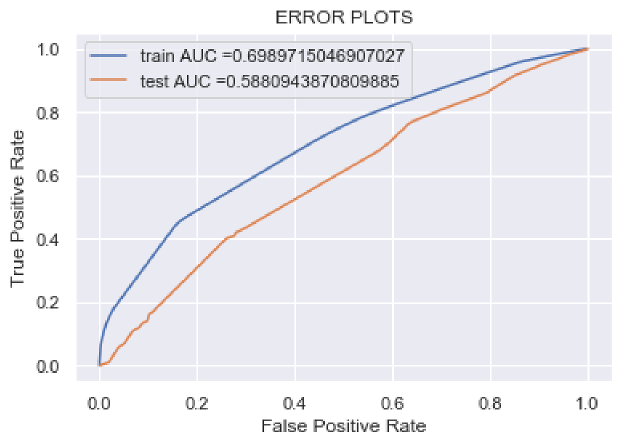
In [123]:

```
#Best Estimator and Best tune parameters
print(clf5.best_estimator_)
#Mean cross-validated score of the best_estimator
print(clf5.score(X_set_train,y_train))
print(clf5.score(X_set_test,y_test))
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight='balanced', criterion='gini',
                       max_depth=10, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=500,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
0.6989715046907027
0.5886741766576952
```

In [122]:

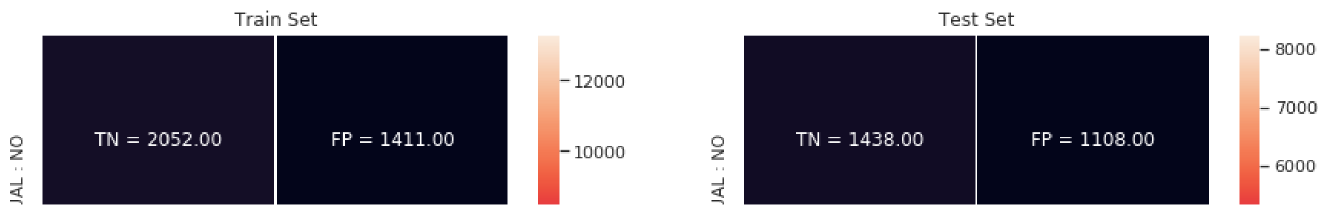
```
clfV1=DecisionTreeClassifier (class_weight = 'balanced',max_depth=10,min_samples_split=500)
clfV1.fit(X_set_train, y_train)
y_train_pred1 = clfV1.predict_proba(X_set_train) [:,1]
y_test_pred1 = clfV1.predict_proba(X_set_test) [:,1]
train_fpr1, train_tpr1, tr_thresholds1 = roc_curve(y_train, y_train_pred1)
test_fpr1, test_tpr1, te_thresholds1 = roc_curve(y_test, y_test_pred1)
plt.plot(train_fpr1, train_tpr1, label="train AUC ="+str(auc(train_fpr1, train_tpr1)))
plt.plot(test_fpr1, test_tpr1, label="test AUC ="+str(auc(test_fpr1, test_tpr1)))
plt.legend()
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ERROR PLOTS")
plt.grid(True)
plt.show()
```

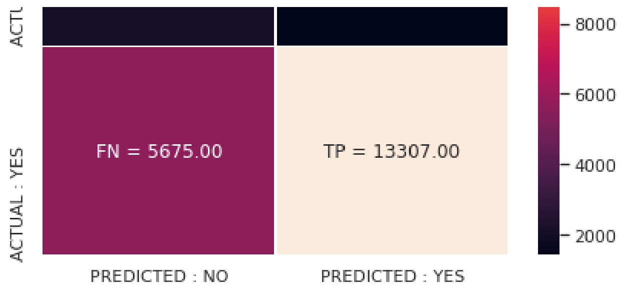


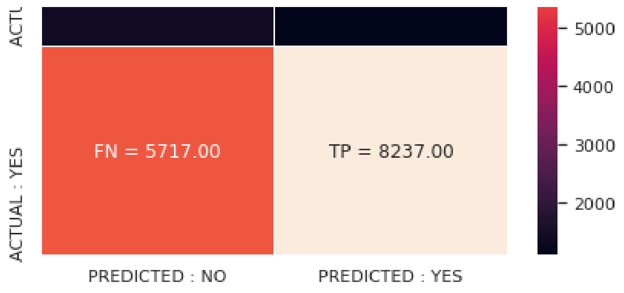
In [0]:

```
#CONFUSION MATRIX
#https://www.quantinsti.com/blog/creating-heatmap-using-python-seaborn
import seaborn as sns; sns.set()
con_m_train = confusion_matrix(y_train, predict(y_train_pred1, tr_thresholds1, train_fpr1, train_tp
r1))
con_m_test = confusion_matrix(y_test, predict(y_test_pred1, te_thresholds1, test_fpr1, test_tpr1))
key = (np.asarray([['TN','FP'], ['FN', 'TP']]))
fig, ax = plt.subplots(1,2, figsize=(15,5))
labels_train = (np.asarray(["{0} = {1:.2f}" .format(key, value) for key, value in zip(key.flatten()
, con_m_train.flatten())])).reshape(2,2)
labels_test = (np.asarray(["{0} = {1:.2f}" .format(key, value) for key, value in zip(key.flatten(),
con_m_test.flatten())])).reshape(2,2)
sns.heatmap(con_m_train, linewidths=.5, xticklabels=['PREDICTED : NO', 'PREDICTED :
YES'],yticklabels=['ACTUAL : NO', 'ACTUAL : YES'], annot = labels_train, fmt = '', ax=ax[0])
sns.heatmap(con_m_test, linewidths=.5, xticklabels=['PREDICTED : NO', 'PREDICTED :
YES'],yticklabels=['ACTUAL : NO', 'ACTUAL : YES'], annot = labels_test, fmt = '', ax=ax[1])
ax[0].set_title('Train Set')
ax[1].set_title('Test Set')
plt.show()
```

```
the maximum value of tpr*(1-fpr) 0.43 for threshold 0.41
the maximum value of tpr*(1-fpr) 0.33 for threshold 0.5
```





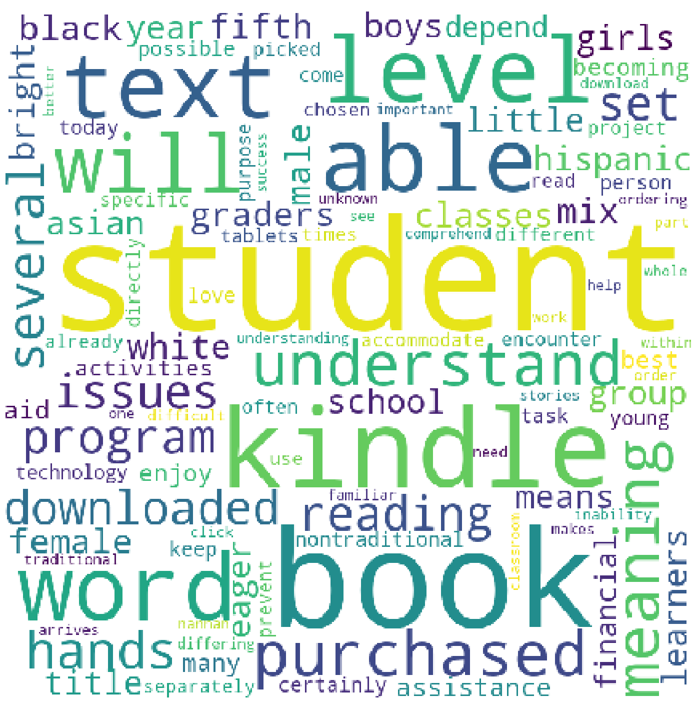


In [0]:

```
#Analysis on the False positives

fpi = []
for i in range(len(y_test)) :
    if (y_test.values[i] == 0) & (predictions1[i] == 1) :
      fpi.append(i)
fp_essay1 = []
for i in fpi :
  fp_essay1.append(X_test['essay'].values[i])
```

In [0]:

```
# Word cloud of essay
from wordcloud import WordCloud, STOPWORDS
comment_words = ' '
stopwords = set(STOPWORDS)
for val in fp_essay1 :
  val = str(val)
  tokens = val.split()
for i in range(len(tokens)):
  tokens[i] = tokens[i].lower()
for words in tokens :
  comment_words = comment_words + words + ' '

wordcloud = WordCloud(width = 800, height = 800, background_color ='white', stopwords = stopwords,m
in_font_size = 10).generate(comment_words)

plt.figure(figsize = (6, 6), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()
```

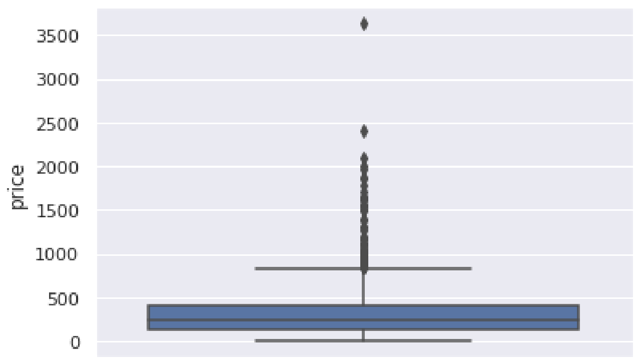


In [0]:

```
#Box Plot (FP 'price')
# first get the columns:
cols = X_test.columns
X_test_falsePos1 = pd.DataFrame(columns=cols)
# get the data of the false pisitives
for i in fpi : # (in fpi all the false positives data points indexes)
  X_test_falsePos1 = X_test_falsePos1.append(X_test.filter(items=[i], axis=0))
sns.boxplot(y='price', data=X_test_falsePos1)
```

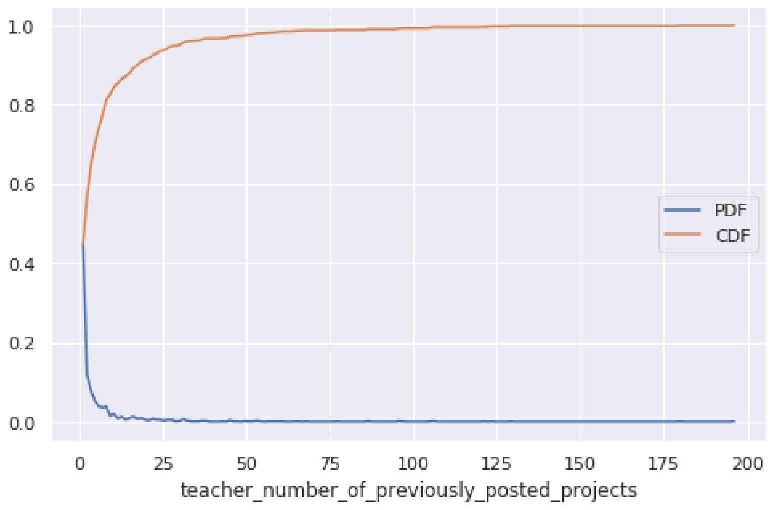
Out[0]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f82ee7f4b38>
```



In [0]:

```
#PDF (FP ,teacher_number_of_previously_posted_projects)
plt.figure(figsize=(8,5))
counts, bin_edges = np.histogram(X_test_falsePos1['teacher_number_of_previously_posted_projects'],
bins='auto', density=True)
pdf = counts/sum(counts)
cdf = np.cumsum(pdf)
pdfP, = plt.plot(bin_edges[1:], pdf)
cdfP, = plt.plot(bin_edges[1:], cdf)
plt.legend([pdfP, cdfP], ["PDF", "CDF"])
plt.xlabel('teacher_number_of_previously_posted_projects')
plt.show()
```



In [124]:

```
# http://zetcode.com/python/prettytable/
from prettytable import PrettyTable
tb = PrettyTable()
tb.field_names= (" Vectorizer ", " Max_depth ", " Min_sample_split "," Test -AUC " , "Train - AUc")
tb.add_row([" Tf - Idf", 10 ,500,69.83,58.86 ])
tb.add_row(["A VG - Tf - Idf", 5 , 500 ,67.5,58.85])
tb.add_row(["Top 5000 Features", 10, 500 ,69.89,58.80])
print(tb.get_string(titles = "Decision trees- Observations"))
```

```
+--------------------+-------------+---------------------+-------------+-------------+
|     Vectorizer     |  Max_depth  |  Min_sample_split   |  Test -AUC  | Train - AUc |
+--------------------+-------------+---------------------+-------------+-------------+
|      Tf - Idf      |      10     |         500         |    69.83    |    58.86    |
|  A VG - Tf - Idf   |      5      |         500         |     67.5    |    58.85    |
| Top 5000 Features  |      10     |         500         |    69.89    |     58.8    |
+--------------------+-------------+---------------------+-------------+-------------+
```

In [0]:

```
```