In [61]:
```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer


import re

from nltk.corpus import stopwords
import pickle

from tqdm import tqdm
import os
```

1. Reading Data

In [62]:
```
project_data=pd.read_csv("C:/Users/91888/Desktop/Assignment/NaiveBayes Assignment/train_data.csv")
resource_data=pd.read_csv("C:/Users/91888/Desktop/Assignment/NaiveBayes Assignment/resources.csv")
```

In [63]:
```
print("Number of data points in train data", project_data.shape)
print('_'*50)
print("The attributes of data :", project_data.columns.values)
```

```
Number of data points in train data (109248, 17)
__________________________________________________
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_prefix' 'school_state'
 'project_submitted_datetime' 'project_grade_category'
 'project_subject_categories' 'project_subject_subcategories'
 'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects' 'project_is_approved']
```

In [64]:
```
print("Number of data points in resource data",resource_data.shape)
print('_'*50)
print("The attributes of data :",resource_data.columns.values)
```

```
Number of data points in resource data (1541272, 4)
__________________________________________________
The attributes of data : ['id' 'description' 'quantity' 'price']
```

2. Preprocessing Categorical Features

teacher_prefix

In [65]:
```
project_data['teacher_prefix'].value_counts()
```

Out[65]:
```
Mrs.       57269
Ms.        38955
Mr.        10648
Teacher     2360
Dr.           13
Name: teacher_prefix, dtype: int64
```

In [66]:
```
print(project_data['teacher_prefix'].isnull().values.any())
print("Number of nan values", project_data['teacher_prefix'].isnull().values.sum())
```

```
True
Number of nan values 3
```

In [67]:
```
#replace msiing values with Mrs
project_data['teacher_prefix']=project_data['teacher_prefix'].fillna('Mrs.')
```

In [68]:
```
project_data['teacher_prefix'].value_counts()
```

Out[68]:
```
Mrs.       57272
Ms.        38955
Mr.        10648
Teacher     2360
Dr.           13
Name: teacher_prefix, dtype: int64
```

In [69]:
```
project_data['teacher_prefix']=project_data['teacher_prefix'].str.replace('.','')
project_data['teacher_prefix']=project_data['teacher_prefix'].str.lower()
project_data['teacher_prefix'].value_counts()
```

Out[69]:
```
mrs        57272
ms         38955
mr         10648
teacher     2360
dr            13
Name: teacher_prefix, dtype: int64
```

project_grade_category

In [70]:
```
project_data['project_grade_category'].value_counts()
```

Out[70]:
```
Grades PreK-2    44225
Grades 3-5       37137
Grades 6-8       16923
Grades 9-12      10963
Name: project_grade_category, dtype: int64
```

In [71]:
```
print(project_data['project_grade_category'].isnull().values.any())
print("Number of nan values", project_data['project_grade_category'].isnull().values.sum())
```

```
False
Number of nan values 0
```

In [72]:
```
project_data['project_grade_category']=project_data['project_grade_category'].str.replace(' ','_')
project_data['project_grade_category']=project_data['project_grade_category'].str.replace('-','_')
project_data['project_grade_category']=project_data['project_grade_category'].str.lower()
project_data['project_grade_category'].value_counts()
```

Out[72]:
```
grades_prek_2    44225
grades_3_5       37137
grades_6_8       16923
grades_9_12      10963
Name: project_grade_category, dtype: int64
```

school_state

In [73]:
```
project_data['school_state'].value_counts()
```

Out[73]:
CA	15388
TX	7396
NY	7318
FL	6185
NC	5091
IL	4350
GA	3963
SC	3936
MI	3161
PA	3109
IN	2620
MO	2576
OH	2467
LA	2394
MA	2389
WA	2334
OK	2276
NJ	2237
AZ	2147
VA	2045
WI	1827
AL	1762
UT	1731
TN	1688
CT	1663
MD	1514
NV	1367
MS	1323
KY	1304
OR	1242
MN	1208
CO	1111
AR	1049
ID	693
IA	666
KS	634
NM	557
DC	516
HI	507
ME	505
WV	503

```
NH      348
AK      345
DE      343
NE      309
SD      300
RI      285
MT      245
ND      143
WY       98
VT       80
Name: school_state, dtype: int64
```

In [74]:
```
project_data['school_state'].isnull().values.any()
```

Out[74]:
```
False
```

In [75]:
```
project_data['school_state']=project_data['school_state'].str.lower()
project_data['school_state'].value_counts()
```

Out[75]:
ca	15388
tx	7396
ny	7318
fl	6185
nc	5091
il	4350
ga	3963
sc	3936
mi	3161
pa	3109
in	2620
mo	2576
oh	2467
la	2394
ma	2389
wa	2334
ok	2276
nj	2237
az	2147
va	2045
wi	1827
al	1762
ut	1731
tn	1688
ct	1663
md	1514
nv	1367
ms	1323
ky	1304
or	1242
mn	1208
co	1111
ar	1049
id	693
ia	666
ks	634
nm	557
dc	516
hi	507
me	505
wv	503

```
nh      348
ak      345
de      343
ne      309
sd      300
ri      285
mt      245
nd      143
wy       98
vt       80
Name: school_state, dtype: int64
```

project_subject_categories

In [76]:
```
project_data['project_subject_categories'].value_counts()
```

Out[76]:

Literacy & Language	23655
Math & Science	17072
Literacy & Language, Math & Science	14636
Health & Sports	10177
Music & The Arts	5180
Special Needs	4226
Literacy & Language, Special Needs	3961
Applied Learning	3771
Math & Science, Literacy & Language	2289
Applied Learning, Literacy & Language	2191
History & Civics	1851
Math & Science, Special Needs	1840
Literacy & Language, Music & The Arts	1757
Math & Science, Music & The Arts	1642
Applied Learning, Special Needs	1467
History & Civics, Literacy & Language	1421
Health & Sports, Special Needs	1391
Warmth, Care & Hunger	1309
Math & Science, Applied Learning	1220
Applied Learning, Math & Science	1052
Literacy & Language, History & Civics	809
Health & Sports, Literacy & Language	803
Applied Learning, Music & The Arts	758
Math & Science, History & Civics	652
Literacy & Language, Applied Learning	636
Applied Learning, Health & Sports	608
Math & Science, Health & Sports	414
History & Civics, Math & Science	322
History & Civics, Music & The Arts	312
Special Needs, Music & The Arts	302
Health & Sports, Math & Science	271
History & Civics, Special Needs	252
Health & Sports, Applied Learning	192
Applied Learning, History & Civics	178
Health & Sports, Music & The Arts	155
Music & The Arts, Special Needs	138
Literacy & Language, Health & Sports	72
Health & Sports, History & Civics	43
Special Needs, Health & Sports	42
History & Civics, Applied Learning	42
Health & Sports, Warmth, Care & Hunger	23

```
Special Needs, Warmth, Care & Hunger            23
Music & The Arts, Health & Sports               19
Music & The Arts, History & Civics              18
History & Civics, Health & Sports               13
Math & Science, Warmth, Care & Hunger           11
Music & The Arts, Applied Learning              10
Applied Learning, Warmth, Care & Hunger         10
Literacy & Language, Warmth, Care & Hunger       9
Music & The Arts, Warmth, Care & Hunger          2
History & Civics, Warmth, Care & Hunger          1
Name: project_subject_categories, dtype: int64
```

In [77]:
```
print(project_data['project_subject_categories'].isnull().values.any())
print("Number of nan values", project_data['project_subject_categories'].isnull().values.sum())
```

```
False
Number of nan values 0
```

In [78]:
```
project_data['project_subject_categories'] = project_data['project_subject_categories'].str.replace(' The ','')
project_data['project_subject_categories'] = project_data['project_subject_categories'].str.replace(' ','')
project_data['project_subject_categories'] = project_data['project_subject_categories'].str.replace('&','_')
project_data['project_subject_categories'] = project_data['project_subject_categories'].str.replace(',','_')
project_data['project_subject_categories'] = project_data['project_subject_categories'].str.lower()
project_data['project_subject_categories'].value_counts()
```

Out[78]:

literacy_language	23655
math_science	17072
literacy_language_math_science	14636
health_sports	10177
music_arts	5180
specialneeds	4226
literacy_language_specialneeds	3961
appliedlearning	3771
math_science_literacy_language	2289
appliedlearning_literacy_language	2191
history_civics	1851
math_science_specialneeds	1840
literacy_language_music_arts	1757
math_science_music_arts	1642
appliedlearning_specialneeds	1467
history_civics_literacy_language	1421
health_sports_specialneeds	1391
warmth_care_hunger	1309
math_science_appliedlearning	1220
appliedlearning_math_science	1052
literacy_language_history_civics	809
health_sports_literacy_language	803
appliedlearning_music_arts	758
math_science_history_civics	652
literacy_language_appliedlearning	636
appliedlearning_health_sports	608
math_science_health_sports	414
history_civics_math_science	322
history_civics_music_arts	312
specialneeds_music_arts	302
health_sports_math_science	271
history_civics_specialneeds	252
health_sports_appliedlearning	192
appliedlearning_history_civics	178
health_sports_music_arts	155
music_arts_specialneeds	138
literacy_language_health_sports	72
health_sports_history_civics	43
specialneeds_health_sports	42
history_civics_appliedlearning	42
health_sports_warmth_care_hunger	23

```
specialneeds_warmth_care_hunger            23
music_arts_health_sports                   19
music_arts_history_civics                  18
history_civics_health_sports               13
math_science_warmth_care_hunger            11
appliedlearning_warmth_care_hunger         10
music_arts_appliedlearning                 10
literacy_language_warmth_care_hunger        9
music_arts_warmth_care_hunger               2
history_civics_warmth_care_hunger           1
Name: project_subject_categories, dtype: int64
```

project_subject_subcategories

In [79]:
```
project_data['project_subject_subcategories'].value_counts()
```

```
Out[79]: Literacy                                          9486
         Literacy, Mathematics                             8325
         Literature & Writing, Mathematics                 5923
         Literacy, Literature & Writing                    5571
         Mathematics                                       5379
         Literature & Writing                              4501
         Special Needs                                     4226
         Health & Wellness                                 3583
         Applied Sciences, Mathematics                     3399
         Applied Sciences                                  2492
         Literacy, Special Needs                           2440
         Gym & Fitness, Health & Wellness                  2264
         ESL, Literacy                                     2234
         Visual Arts                                       2217
         Music                                             1472
         Warmth, Care & Hunger                             1309
         Literature & Writing, Special Needs               1306
         Gym & Fitness                                     1195
         Health & Wellness, Special Needs                  1189
         Mathematics, Special Needs                        1187
         Environmental Science                             1079
         Team Sports                                       1061
         Applied Sciences, Environmental Science            984
         Environmental Science, Health & Life Science       964
         Music, Performing Arts                             948
         Early Development                                  905
         Environmental Science, Mathematics                 838
         Other                                              831
         Health & Life Science                              827
         Health & Wellness, Nutrition Education             797
                                                           ...
         College & Career Prep, Team Sports                   2
         Nutrition Education, Social Sciences                 2
         Civics & Government, Health & Wellness               2
         Financial Literacy, Health & Wellness                2
         Visual Arts, Warmth, Care & Hunger                   2
         Economics, Health & Life Science                     2
         History & Geography, Warmth, Care & Hunger           1
         Community Service, Gym & Fitness                     1
         Parent Involvement, Team Sports                      1
         Other, Warmth, Care & Hunger                         1
```

```
Economics, Foreign Languages                       1
Gym & Fitness, Parent Involvement                  1
Literature & Writing, Nutrition Education          1
Economics, Other                                   1
ESL, Economics                                     1
Gym & Fitness, Warmth, Care & Hunger               1
Community Service, Music                           1
Civics & Government, Nutrition Education           1
Parent Involvement, Warmth, Care & Hunger          1
Financial Literacy, Performing Arts                1
Economics, Nutrition Education                     1
Community Service, Financial Literacy              1
Civics & Government, Foreign Languages             1
Gym & Fitness, Social Sciences                     1
Financial Literacy, Foreign Languages              1
College & Career Prep, Warmth, Care & Hunger       1
Economics, Music                                   1
ESL, Team Sports                                   1
Extracurricular, Financial Literacy                1
Civics & Government, Parent Involvement            1
Name: project_subject_subcategories, Length: 401, dtype: int64
```

In [80]:
```
print(project_data['project_subject_subcategories'].isnull().values.any())
print("Number of nan values", project_data['project_subject_subcategories'].isnull().values.sum())
```

```
False
Number of nan values 0
```

In [81]:
```
project_data['project_subject_subcategories'] = project_data['project_subject_subcategories'].str.replace(' The ','')
project_data['project_subject_subcategories'] = project_data['project_subject_subcategories'].str.replace(' ','')
project_data['project_subject_subcategories'] = project_data['project_subject_subcategories'].str.replace('&','_')
project_data['project_subject_subcategories'] = project_data['project_subject_subcategories'].str.replace(',','_')
project_data['project_subject_subcategories'] = project_data['project_subject_subcategories'].str.lower()
project_data['project_subject_subcategories'].value_counts()
```

Out[81]:

literacy	9486
literacy_mathematics	8325
literature_writing_mathematics	5923
literacy_literature_writing	5571
mathematics	5379
literature_writing	4501
specialneeds	4226
health_wellness	3583
appliedsciences_mathematics	3399
appliedsciences	2492
literacy_specialneeds	2440
gym_fitness_health_wellness	2264
esl_literacy	2234
visualarts	2217
music	1472
warmth_care_hunger	1309
literature_writing_specialneeds	1306
gym_fitness	1195
health_wellness_specialneeds	1189
mathematics_specialneeds	1187
environmentalscience	1079
teamsports	1061
appliedsciences_environmentalscience	984
environmentalscience_health_lifescience	964
music_performingarts	948
earlydevelopment	905
environmentalscience_mathematics	838
other	831
health_lifescience	827
health_wellness_nutritioneducation	797
...	...
civics_government_health_wellness	2
environmentalscience_teamsports	2
earlydevelopment_economics	2
extracurricular_foreignlanguages	2
foreignlanguages_gym_fitness	2
college_careerprep_teamsports	2
parentinvolvement_warmth_care_hunger	1
communityservice_financialliteracy	1
economics_foreignlanguages	1
civics_government_parentinvolvement	1

```
economics_other                              1
communityservice_music                       1
other_warmth_care_hunger                     1
parentinvolvement_teamsports                 1
esl_economics                                1
history_geography_warmth_care_hunger         1
communityservice_gym_fitness                 1
gym_fitness_socialsciences                   1
financialliteracy_performingarts             1
civics_government_foreignlanguages           1
gym_fitness_parentinvolvement                1
esl_teamsports                               1
economics_music                              1
gym_fitness_warmth_care_hunger               1
extracurricular_financialliteracy            1
literature_writing_nutritioneducation        1
economics_nutritioneducation                 1
civics_government_nutritioneducation         1
financialliteracy_foreignlanguages           1
college_careerprep_warmth_care_hunger        1
Name: project_subject_subcategories, Length: 401, dtype: int64
```

3. Text Processing

project_essay

In [82]:
```
import re

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

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

In [83]:
```
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', 'furthe
r',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'mor
e',\
            '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', "doesn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "were
n't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [84]:
```
from tqdm import tqdm
def preprocess_text(text_data):
    preprocessed_text = []
    # tqdm is for printing the status bar
    for sentance in tqdm(text_data):
        sent = decontracted(sentance)
        sent = sent.replace('\\r', ' ')
        sent = sent.replace('\\n', ' ')
        sent = sent.replace('\\"', ' ')
        sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
        # https://gist.github.com/sebleier/554280
        sent = ' '.join(e for e in sent.split() if e.lower() not in stopwords)
        preprocessed_text.append(sent.lower().strip())
    return preprocessed_text
```

In [85]:
```
project_data["essay"]=project_data["project_essay_1"].map(str) +\
                      project_data["project_essay_2"].map(str) +\
                      project_data["project_essay_3"].map(str) +\
                      project_data["project_essay_4"].map(str)
```

In [86]:
```
print("Printing some random essays")
print(9,project_data["essay"].values[9])
print('-'*100)
print(34,project_data["essay"].values[34])
print('-'*100)
print(147,project_data["essay"].values[147])
```

Printing some random essays
9 Over 95% of my students are on free or reduced lunch. I have a few who are homeless, but despite that, they come t
o school with an eagerness to learn. My students are inquisitive eager learners who embrace the challenge of not ha
ving great books and other resources every day. Many of them are not afforded the opportunity to engage with these
big colorful pages of a book on a regular basis at home and they don't travel to the public library. \r\nIt is my du
ty as a teacher to do all I can to provide each student an opportunity to succeed in every aspect of life. \r\nReadin
g is Fundamental! My students will read these books over and over again while boosting their comprehension skills. Th
ese books will be used for read alouds, partner reading and for Independent reading. \r\nThey will engage in reading
to build their \"Love for Reading\" by reading for pure enjoyment. They will be introduced to some new authors as wel
l as some old favorites. I want my students to be ready for the 21st Century and know the pleasure of holding a good
hard back book in hand. There's nothing like a good book to read! \r\nMy students will soar in Reading, and more bec
ause of your consideration and generous funding contribution. This will help build stamina and prepare for 3rd grade.
Thank you so much for reading our proposal!nannan
---
34 My students mainly come from extremely low-income families, and the majority of them come from homes where both pa
rents work full time. Most of my students are at school from 7:30 am to 6:00 pm (2:30 to 6:00 pm in the after-school
program), and they all receive free and reduced meals for breakfast and lunch. \r\n\r\n\r\nI want my students to feel
as comfortable in my classroom as they do at home. Many of my students take on multiple roles both at home as well as
in school. They are sometimes the caretakers of younger siblings, cooks, babysitters, academics, friends, and most of
all, they are developing who they are going to become as adults. I consider it an essential part of my job to model
helping others gain knowledge in a positive manner. As a result, I have a community of students who love helping each
other in and outside of the classroom. They consistently look for opportunities to support each other's learning in a
kind and helpful way.I am excited to be experimenting with alternative seating in my classroom this school year. Stud
ies have shown that giving students the option of where they sit in a classroom increases focus as well as motivatio
n. \r\n\r\nBy allowing students choice in the classroom, they are able to explore and create in a welcoming environm
ent. Alternative classroom seating has been experimented with more frequently in recent years. I believe (along with
many others), that every child learns differently. This does not only apply to how multiplication is memorized, or a
paper is written, but applies to the space in which they are asked to work. I have had students in the past ask \"Can
I work in the library? Can I work on the carpet?\" My answer was always, \"As long as you're learning, you can work w
herever you want!\" \r\n\r\nWith the yoga balls and the lap-desks, I will be able to increase the options for seating
in my classroom and expand its imaginable space.nannan
---
147 My students are eager to learn and make their mark on the world.\r\n\r\nThey come from a Title 1 school and need
extra love.\r\n\r\nMy fourth grade students are in a high poverty area and still come to school every day to get thei
r education. I am trying to make it fun and educational for them so they can get the most out of their schooling. I c
reated a caring environment for the students to bloom! They deserve the best.\r\nThank you!\r\nI am requesting 1 Chro
mebook to access online interventions, differentiate instruction, and get extra practice. The Chromebook will be used
to supplement ELA and math instruction. Students will play ELA and math games that are engaging and fun, as well as p
articipate in assignments online. This in turn will help my students improve their skills. Having a Chromebook in the
classroom would not only allow students to use the programs at their own pace, but would ensure more students are get
ting adequate time to use the programs. The online programs have been especially beneficial to my students with speci

```
al needs. They are able to work at their level as well as be challenged with some different materials. This is making these students more confident in their abilities.\r\n\r\nThe Chromebook would allow my students to have daily access to computers and increase their computing skills.\r\nThis will change their lives for the better as they become more successful in school. Having access to technology in the classroom would help bridge the achievement gap.nannan
```

In [87]:
```
preprocessed_essays = preprocess_text(project_data['essay'].values)
```

```
100%|██████████| 109248/109248 [01:41<00:00, 1079.10it/s]
```

In [88]:
```
print("printing some random essay")
print(9, preprocessed_essays[9])
print('-'*50)
print(34, preprocessed_essays[34])
print('-'*50)
print(147, preprocessed_essays[147])
```

```
printing some random essay
9 95 students free reduced lunch homeless despite come school eagerness learn students inquisitive eager learners emb
race challenge not great books resources every day many not afforded opportunity engage big colorful pages book regul
ar basis home not travel public library duty teacher provide student opportunity succeed every aspect life reading fu
ndamental students read books boosting comprehension skills books used read alouds partner reading independent readin
g engage reading build love reading reading pure enjoyment introduced new authors well old favorites want students re
ady 21st century know pleasure holding good hard back book hand nothing like good book read students soar reading con
sideration generous funding contribution help build stamina prepare 3rd grade thank much reading proposal nannan
--------------------------------------------------
34 students mainly come extremely low income families majority come homes parents work full time students school 7 30
6 00 pm 2 30 6 00 pm school program receive free reduced meals breakfast lunch want students feel comfortable classro
om home many students take multiple roles home well school sometimes caretakers younger siblings cooks babysitters ac
ademics friends developing going become adults consider essential part job model helping others gain knowledge positi
ve manner result community students love helping outside classroom consistently look opportunities support learning k
ind helpful way excited experimenting alternative seating classroom school year studies shown giving students option
sit classroom increases focus well motivation allowing students choice classroom able explore create welcoming enviro
nment alternative classroom seating experimented frequently recent years believe along many others every child learns
differently not apply multiplication memorized paper written applies space asked work students past ask work library
work carpet answer always long learning work wherever want yoga balls lap desks able increase options seating classro
om expand imaginable space nannan
--------------------------------------------------
147 students eager learn make mark world come title 1 school need extra love fourth grade students high poverty area
still come school every day get education trying make fun educational get schooling created caring environment studen
ts bloom deserve best thank requesting 1 chromebook access online interventions differentiate instruction get extra p
ractice chromebook used supplement ela math instruction students play ela math games engaging fun well participate as
signments online turn help students improve skills chromebook classroom would not allow students use programs pace wo
uld ensure students getting adequate time use programs online programs especially beneficial students special needs a
ble work level well challenged different materials making students confident abilities chromebook would allow student
s daily access computers increase computing skills change lives better become successful school access technology cla
ssroom would help bridge achievement gap nannan
```

In [89]:
```
#adding processed essays to project_data
project_data['processed_essay']=preprocessed_essays
```

4. Preprocessing Numerical Features

In [90]:
```
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'}).reset_index()
price_data.head(2)
```

Out[90]:

	id	price	quantity
0	p000001	459.56	7
1	p000002	515.89	21

In [91]:
```
# join two dataframes in python:
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

In [92]:
```
project_data['price'].head()
```

Out[92]:
```
0    154.60
1    299.00
2    516.85
3    232.90
4     67.98
Name: price, dtype: float64
```

In [93]:
```
project_data.columns.values
```

Out[93]:
```
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', 'essay', 'processed_essay', 'price',
       'quantity'], dtype=object)
```

removing unnecessary columns

In [94]:
```
#removing columns : https://www.geeksforgeeks.org/how-to-drop-one-or-multiple-columns-in-pandas-dataframe/

project_data = project_data.drop(project_data.columns[[0,1,2,5,9,10,11,12,13,14,17,20]], axis=1)
```

In [95]:
```
project_data.head()
```

Out[95]:

	teacher_prefix	school_state	project_grade_category	project_subject_categories	project_subject_subcategories	teacher_numb
0	mrs	in	grades_prek_2	literacy_language	esl_literacy	0
1	mr	fl	grades_6_8	history_civics_health_sports	civics_government_teamsports	7
2	ms	az	grades_6_8	health_sports	health_wellness_teamsports	1
3	mrs	ky	grades_prek_2	literacy_language_math_science	literacy_mathematics	4
4	mrs	tx	grades_prek_2	math_science	mathematics	1

In [96]:
```
#renaming some columns
project_data = project_data.rename(index=str, columns=
                                   {'project_subject_categories':'clean_categories',
                                    'project_subject_subcategories':'clean_subcategories'})
```

In [97]:
```
project_data.head()
```

Out[97]:

	teacher_prefix	school_state	project_grade_category	clean_categories	clean_subcategories	teacher_numbe
0	mrs	in	grades_prek_2	literacy_language	esl_literacy	0
1	mr	fl	grades_6_8	history_civics_health_sports	civics_government_teamsports	7
2	ms	az	grades_6_8	health_sports	health_wellness_teamsports	1
3	mrs	ky	grades_prek_2	literacy_language_math_science	literacy_mathematics	4
4	mrs	tx	grades_prek_2	math_science	mathematics	1

In [ ]:
```
#changing position of columns : https://stackoverflow.com/questions/41968732/set-order-of-columns-in-pandas-dataframe
```

In [98]:
```
project_data = project_data[['processed_essay', 'teacher_prefix','project_grade_category',
                             'school_state','clean_categories','clean_subcategories',
                             'teacher_number_of_previously_posted_projects','price','project_is_approved']]
```

In [99]:
```
project_data.head()
```

Out[99]:

	processed_essay	teacher_prefix	project_grade_category	school_state	clean_categories	clean_subcategorie
0	students english learners working english seco...	mrs	grades_prek_2	in	literacy_language	esl_literacy
1	students arrive school eager learn polite gene...	mr	grades_6_8	fl	history_civics_health_sports	civics_government_teamspor
2	true champions not always ones win guts mia ha...	ms	grades_6_8	az	health_sports	health_wellness_teamsports
3	work unique school filled esl english second l...	mrs	grades_prek_2	ky	literacy_language_math_science	literacy_mathematics
4	second grade classroom next year made around 2...	mrs	grades_prek_2	tx	math_science	mathematics

converting dataframe to csv

In [ ]:
```
#https://www.codegrepper.com/code-examples/python/pandas+save+as+csv
```

In [176]:
```
project_data.to_csv(r'C:/Users/91888/Desktop/Assignment/NaiveBayes Assignment/preprocessed1_data.csv', index = False)
```

5. Loading data

In [177]:
```
data=pd.read_csv('C:/Users/91888/Desktop/Assignment/NaiveBayes Assignment/preprocessed1_data.csv',nrows=100000)
```

In [178]:
```
data.head()
```

Out[178]:

	processed_essay	teacher_prefix	project_grade_category	school_state	clean_categories	clean_subcategorie
0	students english learners working english seco...	mrs	grades_prek_2	in	literacy_language	esl_literacy
1	students arrive school eager learn polite gene...	mr	grades_6_8	fl	history_civics_health_sports	civics_government_teamspor
2	true champions not always ones win guts mia ha...	ms	grades_6_8	az	health_sports	health_wellness_teamsports
3	work unique school filled esl english second l...	mrs	grades_prek_2	ky	literacy_language_math_science	literacy_mathematics
4	second grade classroom next year made around 2...	mrs	grades_prek_2	tx	math_science	mathematics

In [103]:
```
data.shape
```

Out[103]:
```
(100000, 9)
```

In [104]:
```
y = data['project_is_approved'].values
X = data.drop(['project_is_approved'], axis=1)
X.head(1)
```

Out[104]:

	processed_essay	teacher_prefix	project_grade_category	school_state	clean_categories	clean_subcategories	teacher_number_o
0	students english learners working english seco...	mrs	grades_prek_2	in	literacy_language	esl_literacy	0

5.1 Splitting data into Train and Cross Validation

In [105]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, stratify=y)
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.33, stratify=y_train)
```

5.2 Encoding essays

BOW

In [187]:
```
print(X_train.shape, y_train.shape)
print(X_cv.shape, y_cv.shape)
print(X_test.shape, y_test.shape)

print("="*100)


vectorizer1 = CountVectorizer(min_df=5, max_features=5000)
vectorizer1.fit(X_train['processed_essay'].values) # fit has to happen only on train data


X_train_essay_bow = vectorizer1.transform(X_train['processed_essay'].values)
X_cv_essay_bow = vectorizer1.transform(X_cv['processed_essay'].values)
X_test_essay_bow = vectorizer1.transform(X_test['processed_essay'].values)

print("After vectorizations")
print(X_train_essay_bow.shape, y_train.shape)
print(X_cv_essay_bow.shape, y_cv.shape)
print(X_test_essay_bow.shape, y_test.shape)
print("="*100)
```

```
(44890, 8) (44890,)
(22110, 8) (22110,)
(33000, 8) (33000,)
====================================================================================================
After vectorizations
(44890, 5000) (44890,)
(22110, 5000) (22110,)
(33000, 5000) (33000,)
====================================================================================================
```

TFIDF

In [188]:
```
print(X_train.shape, y_train.shape)
print(X_cv.shape, y_cv.shape)
print(X_test.shape, y_test.shape)

print("="*100)


vectorizer2 = TfidfVectorizer(min_df=5, max_features=5000)
vectorizer2.fit(X_train['processed_essay'].values) # fit has to happen only on train data


X_train_essay_tfidf = vectorizer2.transform(X_train['processed_essay'].values)
X_cv_essay_tfidf = vectorizer2.transform(X_cv['processed_essay'].values)
X_test_essay_tfidf = vectorizer2.transform(X_test['processed_essay'].values)

print("After vectorizations")
print(X_train_essay_tfidf.shape, y_train.shape)
print(X_cv_essay_tfidf.shape, y_cv.shape)
print(X_test_essay_tfidf.shape, y_test.shape)
print("="*100)
```

```
(44890, 8) (44890,)
(22110, 8) (22110,)
(33000, 8) (33000,)
====================================================================================================
After vectorizations
(44890, 5000) (44890,)
(22110, 5000) (22110,)
(33000, 5000) (33000,)
====================================================================================================
```

5.3 Encoding Categorical Feature

teacher_prefix

In [190]:
```
vectorizer3 = CountVectorizer()
vectorizer3.fit(X_train['teacher_prefix'].values)


X_train_teacher_ohe = vectorizer3.transform(X_train['teacher_prefix'].values)
X_cv_teacher_ohe = vectorizer3.transform(X_cv['teacher_prefix'].values)
X_test_teacher_ohe = vectorizer3.transform(X_test['teacher_prefix'].values)

print("After vectorizations")
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(vectorizer3.get_feature_names())
print("="*100)
```

```
After vectorizations
(44890, 5) (44890,)
(22110, 5) (22110,)
(33000, 5) (33000,)
['dr', 'mr', 'mrs', 'ms', 'teacher']
====================================================================================================
```

project_grade_category

In [191]:
```
vectorizer4 = CountVectorizer()
vectorizer4.fit(X_train['project_grade_category'].values)


X_train_grade_ohe = vectorizer4.transform(X_train['project_grade_category'].values)
X_cv_grade_ohe = vectorizer4.transform(X_cv['project_grade_category'].values)
X_test_grade_ohe = vectorizer4.transform(X_test['project_grade_category'].values)

print("After vectorizations")
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(vectorizer4.get_feature_names())
print("="*100)
```

```
After vectorizations
(44890, 4) (44890,)
(22110, 4) (22110,)
(33000, 4) (33000,)
['grades_3_5', 'grades_6_8', 'grades_9_12', 'grades_prek_2']
====================================================================================================
```

school_state

In [193]:
```
vectorizer5 = CountVectorizer()
vectorizer5.fit(X_train['school_state'].values)


X_train_state_ohe = vectorizer5.transform(X_train['school_state'].values)
X_cv_state_ohe = vectorizer5.transform(X_cv['school_state'].values)
X_test_state_ohe = vectorizer5.transform(X_test['school_state'].values)

print("After vectorizations")
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)
print(vectorizer5.get_feature_names())
print("="*100)
```

```
After vectorizations
(44890, 51) (44890,)
(22110, 51) (22110,)
(33000, 51) (33000,)
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga', 'hi', 'ia', 'id', 'il', 'in', 'ks', 'ky', 'la', 'm
a', 'md', 'me', 'mi', 'mn', 'mo', 'ms', 'mt', 'nc', 'nd', 'ne', 'nh', 'nj', 'nm', 'nv', 'ny', 'oh', 'ok', 'or', 'pa',
'ri', 'sc', 'sd', 'tn', 'tx', 'ut', 'va', 'vt', 'wa', 'wi', 'wv', 'wy']
====================================================================================================
```

clean_categories

In [194]:
```
vectorizer6 = CountVectorizer()
vectorizer6.fit(X_train['clean_categories'].values)


X_train_category_ohe = vectorizer6.transform(X_train['clean_categories'].values)
X_cv_category_ohe = vectorizer6.transform(X_cv['clean_categories'].values)
X_test_category_ohe = vectorizer6.transform(X_test['clean_categories'].values)

print("After vectorizations")
print(X_train_category_ohe.shape, y_train.shape)
print(X_cv_category_ohe.shape, y_cv.shape)
print(X_test_category_ohe.shape, y_test.shape)
print(vectorizer6.get_feature_names())
print("="*100)
```

```
After vectorizations
(44890, 50) (44890,)
(22110, 50) (22110,)
(33000, 50) (33000,)
['appliedlearning', 'appliedlearning_health_sports', 'appliedlearning_history_civics', 'appliedlearning_literacy_lang
uage', 'appliedlearning_math_science', 'appliedlearning_music_arts', 'appliedlearning_specialneeds', 'appliedlearning
_warmth_care_hunger', 'health_sports', 'health_sports_appliedlearning', 'health_sports_history_civics', 'health_sport
s_literacy_language', 'health_sports_math_science', 'health_sports_music_arts', 'health_sports_specialneeds', 'health
_sports_warmth_care_hunger', 'history_civics', 'history_civics_appliedlearning', 'history_civics_health_sports', 'his
tory_civics_literacy_language', 'history_civics_math_science', 'history_civics_music_arts', 'history_civics_specialne
eds', 'literacy_language', 'literacy_language_appliedlearning', 'literacy_language_health_sports', 'literacy_language
_history_civics', 'literacy_language_math_science', 'literacy_language_music_arts', 'literacy_language_specialneeds',
'literacy_language_warmth_care_hunger', 'math_science', 'math_science_appliedlearning', 'math_science_health_sports',
'math_science_history_civics', 'math_science_literacy_language', 'math_science_music_arts', 'math_science_specialneed
s', 'math_science_warmth_care_hunger', 'music_arts', 'music_arts_appliedlearning', 'music_arts_health_sports', 'music
_arts_history_civics', 'music_arts_specialneeds', 'music_arts_warmth_care_hunger', 'specialneeds', 'specialneeds_heal
th_sports', 'specialneeds_music_arts', 'specialneeds_warmth_care_hunger', 'warmth_care_hunger']
====================================================================================================
```

clean_subcategories

In [195]:
```
vectorizer7 = CountVectorizer()
vectorizer7.fit(X_train['clean_subcategories'].values)


X_train_subcategory_ohe = vectorizer7.transform(X_train['clean_subcategories'].values)
X_cv_subcategory_ohe = vectorizer7.transform(X_cv['clean_subcategories'].values)
X_test_subcategory_ohe = vectorizer7.transform(X_test['clean_subcategories'].values)

print("After vectorizations")
print(X_train_subcategory_ohe.shape, y_train.shape)
print(X_cv_subcategory_ohe.shape, y_cv.shape)
print(X_test_subcategory_ohe.shape, y_test.shape)
print(vectorizer7.get_feature_names())
print("="*100)
```

```
After vectorizations
(44890, 379) (44890,)
(22110, 379) (22110,)
(33000, 379) (33000,)
['appliedsciences', 'appliedsciences_charactereducation', 'appliedsciences_civics_government', 'appliedsciences_colle
ge_careerprep', 'appliedsciences_communityservice', 'appliedsciences_earlydevelopment', 'appliedsciences_economics',
'appliedsciences_environmentalscience', 'appliedsciences_esl', 'appliedsciences_extracurricular', 'appliedsciences_fi
nancialliteracy', 'appliedsciences_foreignlanguages', 'appliedsciences_gym_fitness', 'appliedsciences_health_lifescie
nce', 'appliedsciences_health_wellness', 'appliedsciences_history_geography', 'appliedsciences_literacy', 'appliedsci
ences_literature_writing', 'appliedsciences_mathematics', 'appliedsciences_music', 'appliedsciences_nutritioneducatio
n', 'appliedsciences_other', 'appliedsciences_parentinvolvement', 'appliedsciences_performingarts', 'appliedsciences_
socialsciences', 'appliedsciences_specialneeds', 'appliedsciences_teamsports', 'appliedsciences_visualarts', 'charact
ereducation', 'charactereducation_civics_government', 'charactereducation_college_careerprep', 'charactereducation_co
mmunityservice', 'charactereducation_earlydevelopment', 'charactereducation_environmentalscience', 'charactereducatio
n_esl', 'charactereducation_extracurricular', 'charactereducation_financialliteracy', 'charactereducation_foreignlang
uages', 'charactereducation_gym_fitness', 'charactereducation_health_lifescience', 'charactereducation_health_wellnes
s', 'charactereducation_history_geography', 'charactereducation_literacy', 'charactereducation_literature_writing',
'charactereducation_mathematics', 'charactereducation_music', 'charactereducation_nutritioneducation', 'charactereduc
ation_other', 'charactereducation_parentinvolvement', 'charactereducation_performingarts', 'charactereducation_social
sciences', 'charactereducation_specialneeds', 'charactereducation_teamsports', 'charactereducation_visualarts', 'char
actereducation_warmth_care_hunger', 'civics_government', 'civics_government_college_careerprep', 'civics_government_c
ommunityservice', 'civics_government_economics', 'civics_government_environmentalscience', 'civics_government_esl',
'civics_government_extracurricular', 'civics_government_financialliteracy', 'civics_government_health_lifescience',
'civics_government_history_geography', 'civics_government_literacy', 'civics_government_literature_writing', 'civics_
government_mathematics', 'civics_government_parentinvolvement', 'civics_government_performingarts', 'civics_governmen
t_socialsciences', 'civics_government_specialneeds', 'civics_government_teamsports', 'civics_government_visualarts',
'college_careerprep', 'college_careerprep_communityservice', 'college_careerprep_earlydevelopment', 'college_careerpr
ep_economics', 'college_careerprep_environmentalscience', 'college_careerprep_esl', 'college_careerprep_extracurricul
ar', 'college_careerprep_financialliteracy', 'college_careerprep_foreignlanguages', 'college_careerprep_gym_fitness',
'college_careerprep_health_lifescience', 'college_careerprep_health_wellness', 'college_careerprep_history_geograph
y', 'college_careerprep_literacy', 'college_careerprep_literature_writing', 'college_careerprep_mathematics', 'colleg
e_careerprep_music', 'college_careerprep_nutritioneducation', 'college_careerprep_other', 'college_careerprep_parenti
nvolvement', 'college_careerprep_performingarts', 'college_careerprep_socialsciences', 'college_careerprep_specialnee
ds', 'college_careerprep_visualarts', 'college_careerprep_warmth_care_hunger', 'communityservice', 'communityservice_
earlydevelopment', 'communityservice_economics', 'communityservice_environmentalscience', 'communityservice_extracurr
icular', 'communityservice_financialliteracy', 'communityservice_gym_fitness', 'communityservice_health_lifescience',
'communityservice_health_wellness', 'communityservice_history_geography', 'communityservice_literacy', 'communityserv
ice_literature_writing', 'communityservice_mathematics', 'communityservice_nutritioneducation', 'communityservice_oth
er', 'communityservice_parentinvolvement', 'communityservice_performingarts', 'communityservice_socialsciences', 'com
munityservice_specialneeds', 'communityservice_visualarts', 'earlydevelopment', 'earlydevelopment_economics', 'earlyd
evelopment_environmentalscience', 'earlydevelopment_extracurricular', 'earlydevelopment_financialliteracy', 'earlydev
```

elopment_foreignlanguages', 'earlydevelopment_gym_fitness', 'earlydevelopment_health_lifescience', 'earlydevelopment_
health_wellness', 'earlydevelopment_history_geography', 'earlydevelopment_literacy', 'earlydevelopment_literature_wri
ting', 'earlydevelopment_mathematics', 'earlydevelopment_music', 'earlydevelopment_nutritioneducation', 'earlydevelop
ment_other', 'earlydevelopment_parentinvolvement', 'earlydevelopment_performingarts', 'earlydevelopment_socialscience
s', 'earlydevelopment_specialneeds', 'earlydevelopment_teamsports', 'earlydevelopment_visualarts', 'earlydevelopment_
warmth_care_hunger', 'economics', 'economics_environmentalscience', 'economics_financialliteracy', 'economics_health_
lifescience', 'economics_history_geography', 'economics_literacy', 'economics_mathematics', 'economics_nutritioneduca
tion', 'economics_other', 'economics_socialsciences', 'economics_specialneeds', 'economics_visualarts', 'environmenta
lscience', 'environmentalscience_extracurricular', 'environmentalscience_financialliteracy', 'environmentalscience_fo
reignlanguages', 'environmentalscience_gym_fitness', 'environmentalscience_health_lifescience', 'environmentalscience
_health_wellness', 'environmentalscience_history_geography', 'environmentalscience_literacy', 'environmentalscience_l
iterature_writing', 'environmentalscience_mathematics', 'environmentalscience_music', 'environmentalscience_nutrition
education', 'environmentalscience_other', 'environmentalscience_parentinvolvement', 'environmentalscience_performinga
rts', 'environmentalscience_socialsciences', 'environmentalscience_specialneeds', 'environmentalscience_teamsports',
'environmentalscience_visualarts', 'esl', 'esl_earlydevelopment', 'esl_economics', 'esl_environmentalscience', 'esl_e
xtracurricular', 'esl_financialliteracy', 'esl_foreignlanguages', 'esl_gym_fitness', 'esl_health_lifescience', 'esl_h
ealth_wellness', 'esl_history_geography', 'esl_literacy', 'esl_literature_writing', 'esl_mathematics', 'esl_music',
'esl_nutritioneducation', 'esl_other', 'esl_parentinvolvement', 'esl_performingarts', 'esl_socialsciences', 'esl_spec
ialneeds', 'esl_visualarts', 'extracurricular', 'extracurricular_financialliteracy', 'extracurricular_foreignlanguage
s', 'extracurricular_gym_fitness', 'extracurricular_health_lifescience', 'extracurricular_health_wellness', 'extracur
ricular_history_geography', 'extracurricular_literacy', 'extracurricular_literature_writing', 'extracurricular_mathem
atics', 'extracurricular_music', 'extracurricular_nutritioneducation', 'extracurricular_other', 'extracurricular_pare
ntinvolvement', 'extracurricular_performingarts', 'extracurricular_socialsciences', 'extracurricular_specialneeds',
'extracurricular_teamsports', 'extracurricular_visualarts', 'financialliteracy', 'financialliteracy_health_lifescienc
e', 'financialliteracy_health_wellness', 'financialliteracy_history_geography', 'financialliteracy_literacy', 'financ
ialliteracy_literature_writing', 'financialliteracy_mathematics', 'financialliteracy_other', 'financialliteracy_paren
tinvolvement', 'financialliteracy_socialsciences', 'financialliteracy_specialneeds', 'financialliteracy_visualarts',
'foreignlanguages', 'foreignlanguages_health_lifescience', 'foreignlanguages_health_wellness', 'foreignlanguages_hist
ory_geography', 'foreignlanguages_literacy', 'foreignlanguages_literature_writing', 'foreignlanguages_mathematics',
'foreignlanguages_music', 'foreignlanguages_other', 'foreignlanguages_performingarts', 'foreignlanguages_socialscienc
es', 'foreignlanguages_specialneeds', 'foreignlanguages_visualarts', 'gym_fitness', 'gym_fitness_health_lifescience',
'gym_fitness_health_wellness', 'gym_fitness_history_geography', 'gym_fitness_literacy', 'gym_fitness_literature_writi
ng', 'gym_fitness_mathematics', 'gym_fitness_music', 'gym_fitness_nutritioneducation', 'gym_fitness_other', 'gym_fitn
ess_performingarts', 'gym_fitness_specialneeds', 'gym_fitness_teamsports', 'gym_fitness_visualarts', 'gym_fitness_war
mth_care_hunger', 'health_lifescience', 'health_lifescience_health_wellness', 'health_lifescience_history_geography',
'health_lifescience_literacy', 'health_lifescience_literature_writing', 'health_lifescience_mathematics', 'health_lif
escience_music', 'health_lifescience_nutritioneducation', 'health_lifescience_other', 'health_lifescience_parentinvol
vement', 'health_lifescience_performingarts', 'health_lifescience_socialsciences', 'health_lifescience_specialneeds',
'health_lifescience_teamsports', 'health_lifescience_visualarts', 'health_lifescience_warmth_care_hunger', 'health_we
llness', 'health_wellness_history_geography', 'health_wellness_literacy', 'health_wellness_literature_writing', 'heal
th_wellness_mathematics', 'health_wellness_music', 'health_wellness_nutritioneducation', 'health_wellness_other', 'he
alth_wellness_parentinvolvement', 'health_wellness_performingarts', 'health_wellness_socialsciences', 'health_wellnes

```
s_specialneeds', 'health_wellness_teamsports', 'health_wellness_visualarts', 'health_wellness_warmth_care_hunger', 'h
istory_geography', 'history_geography_literacy', 'history_geography_literature_writing', 'history_geography_mathemati
cs', 'history_geography_music', 'history_geography_other', 'history_geography_parentinvolvement', 'history_geography_
performingarts', 'history_geography_socialsciences', 'history_geography_specialneeds', 'history_geography_teamsport
s', 'history_geography_visualarts', 'literacy', 'literacy_literature_writing', 'literacy_mathematics', 'literacy_musi
c', 'literacy_nutritioneducation', 'literacy_other', 'literacy_parentinvolvement', 'literacy_performingarts', 'litera
cy_socialsciences', 'literacy_specialneeds', 'literacy_teamsports', 'literacy_visualarts', 'literacy_warmth_care_hung
er', 'literature_writing', 'literature_writing_mathematics', 'literature_writing_music', 'literature_writing_other',
'literature_writing_parentinvolvement', 'literature_writing_performingarts', 'literature_writing_socialsciences', 'li
terature_writing_specialneeds', 'literature_writing_teamsports', 'literature_writing_visualarts', 'literature_writing
_warmth_care_hunger', 'mathematics', 'mathematics_music', 'mathematics_nutritioneducation', 'mathematics_other', 'mat
hematics_parentinvolvement', 'mathematics_performingarts', 'mathematics_socialsciences', 'mathematics_specialneeds',
'mathematics_teamsports', 'mathematics_visualarts', 'mathematics_warmth_care_hunger', 'music', 'music_other', 'music_
parentinvolvement', 'music_performingarts', 'music_socialsciences', 'music_specialneeds', 'music_teamsports', 'music_
visualarts', 'nutritioneducation', 'nutritioneducation_other', 'nutritioneducation_socialsciences', 'nutritioneducati
on_specialneeds', 'nutritioneducation_teamsports', 'nutritioneducation_visualarts', 'nutritioneducation_warmth_care_h
unger', 'other', 'other_parentinvolvement', 'other_socialsciences', 'other_specialneeds', 'other_teamsports', 'other_
visualarts', 'parentinvolvement', 'parentinvolvement_performingarts', 'parentinvolvement_socialsciences', 'parentinvo
lvement_specialneeds', 'parentinvolvement_teamsports', 'parentinvolvement_visualarts', 'parentinvolvement_warmth_care
_hunger', 'performingarts', 'performingarts_socialsciences', 'performingarts_specialneeds', 'performingarts_teamsport
s', 'performingarts_visualarts', 'socialsciences', 'socialsciences_specialneeds', 'socialsciences_teamsports', 'socia
lsciences_visualarts', 'specialneeds', 'specialneeds_teamsports', 'specialneeds_visualarts', 'specialneeds_warmth_car
e_hunger', 'teamsports', 'teamsports_visualarts', 'visualarts', 'visualarts_warmth_care_hunger', 'warmth_care_hunge
r']
==========================================================================================
```

5.4 Encoding Numerical Feature

price

In [119]:
```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
normalizer.fit(X_train['price'].values.reshape(1,-1))

X_train_price_norm = normalizer.transform(X_train['price'].values.reshape(1,-1))
X_cv_price_norm = normalizer.transform(X_cv['price'].values.reshape(1,-1))
X_test_price_norm = normalizer.transform(X_test['price'].values.reshape(1,-1))

X_train_price_norm=X_train_price_norm.reshape(-1,1)
X_cv_price_norm=X_cv_price_norm.reshape(-1,1)
X_test_price_norm=X_test_price_norm.reshape(-1,1)

print("After vectorizations")
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)
print("="*100)
```

```
After vectorizations
(44890, 1) (44890,)
(22110, 1) (22110,)
(33000, 1) (33000,)
====================================================================================================
```

teacher_number_of_previously_posted_projects

In [120]:
```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
normalizer.fit(X_train['teacher_number_of_previously_posted_projects'].values.reshape(1,-1))

X_train_teachernumber_norm = normalizer.transform(X_train['teacher_number_of_previously_posted_projects'].values.resha
pe(1,-1))
X_cv_teachernumber_norm = normalizer.transform(X_cv['teacher_number_of_previously_posted_projects'].values.reshape(1,-
1))
X_test_teachernumber_norm = normalizer.transform(X_test['teacher_number_of_previously_posted_projects'].values.reshape
(1,-1))

X_train_teachernumber_norm=X_train_teachernumber_norm.reshape(-1,1)
X_cv_teachernumber_norm=X_cv_teachernumber_norm.reshape(-1,1)
X_test_teachernumber_norm=X_test_teachernumber_norm.reshape(-1,1)

print("After vectorizations")
print(X_train_teachernumber_norm.shape, y_train.shape)
print(X_cv_teachernumber_norm.shape, y_cv.shape)
print(X_test_teachernumber_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(44890, 1) (44890,)
(22110, 1) (22110,)
(33000, 1) (33000,)
====================================================================================================
```

5.5 Merging all features

set 1

In [121]:
```
from scipy.sparse import hstack
X_tr = hstack((X_train_essay_bow, X_train_teacher_ohe, X_train_grade_ohe, X_train_state_ohe,X_train_category_ohe,X_tra
in_subcategory_ohe,X_train_price_norm,X_train_teachernumber_norm)).tocsr()
X_cr = hstack((X_cv_essay_bow, X_cv_teacher_ohe, X_cv_grade_ohe, X_cv_state_ohe,X_cv_category_ohe,X_cv_subcategory_ohe
,X_cv_price_norm,X_cv_teachernumber_norm)).tocsr()
X_te = hstack((X_test_essay_bow, X_test_teacher_ohe, X_test_grade_ohe, X_test_state_ohe,X_test_category_ohe,X_test_sub
category_ohe,X_test_price_norm,X_test_teachernumber_norm)).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
(44890, 5491) (44890,)
(22110, 5491) (22110,)
(33000, 5491) (33000,)
====================================================================================================
```

Applying Naive Bayes on Set 1

In [122]:
```
alpha=[0.00001,0.0005, 0.0001,0.005,0.001,0.05,0.01,0.1,0.5,1,5,10,50,100]
```

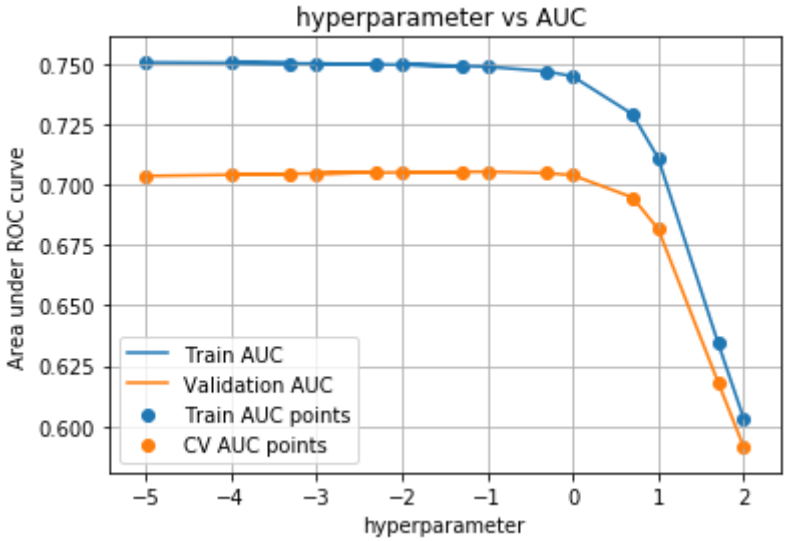
In [125]:
```
# how to take log of list : https://stackoverflow.com/questions/11656767/how-to-take-the-log-of-all-elements-of-a-list
from math import log
import numpy
log_alpha=[numpy.log10(y) for y in alpha]
log_alpha
```

Out[125]:
```
[-5.0,
 -3.3010299956639813,
 -4.0,
 -2.3010299956639813,
 -3.0,
 -1.3010299956639813,
 -2.0,
 -1.0,
 -0.3010299956639812,
 0.0,
 0.6989700043360189,
 1.0,
 1.6989700043360187,
 2.0]
```

Finding best alpha

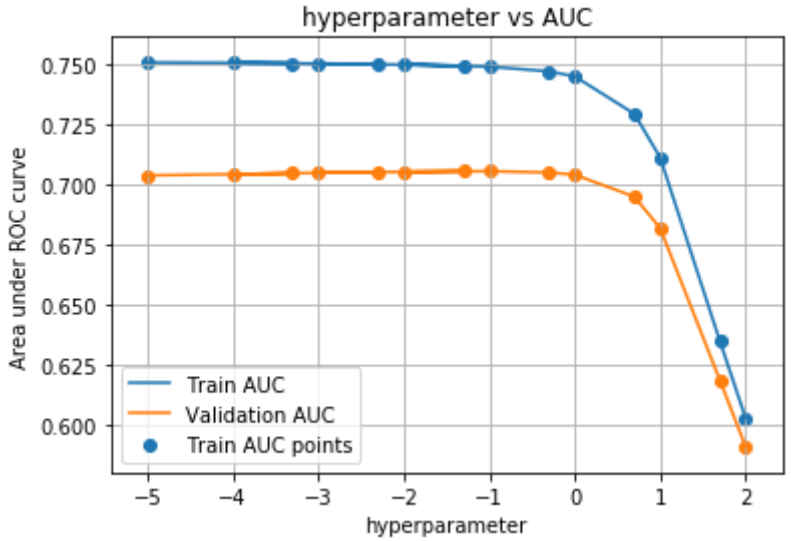
In [126]:
```
from sklearn.naive_bayes import MultinomialNB
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score

train_auc = []
cv_auc = []

for i in tqdm(alpha):
    neigh = MultinomialNB(alpha = i, class_prior = [0.5,0.5])
    neigh.fit(X_tr, y_train)

    y_train_pred = neigh.predict_proba( X_tr)[:, 1]
    y_cv_pred = neigh.predict_proba(X_cr)[:, 1]


    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))

plt.plot(log_alpha, train_auc, label='Train AUC')
plt.plot(log_alpha, cv_auc, label='Validation AUC')

plt.scatter(log_alpha, train_auc, label='Train AUC points')
plt.scatter(log_alpha, cv_auc, label='CV AUC points')

plt.legend()
plt.xlabel("hyperparameter")
plt.ylabel("Area under ROC curve")
plt.title("hyperparameter vs AUC ")
plt.grid()
plt.show()
```

```
100%|██████████| 14/14 [00:02<00:00,  5.87it/s]
```



In [127]:
```
from sklearn.naive_bayes import MultinomialNB
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score

train1_auc = []
cv1_auc = []

for i in tqdm(alpha):
    neigh = MultinomialNB(alpha = i,fit_prior= True, class_prior = None)
    neigh.fit(X_tr, y_train)

    y_train_pred = neigh.predict_proba( X_tr)[:, 1]
    y_cv_pred = neigh.predict_proba(X_cr)[:, 1]


    train1_auc.append(roc_auc_score(y_train,y_train_pred))
    cv1_auc.append(roc_auc_score(y_cv, y_cv_pred))

plt.plot(log_alpha, train1_auc, label='Train AUC')
plt.plot(log_alpha, cv1_auc, label='Validation AUC')

plt.scatter(log_alpha, train1_auc, label='Train AUC points')
plt.scatter(log_alpha, cv1_auc)

plt.legend()
plt.xlabel("hyperparameter")
plt.ylabel("Area under ROC curve")
plt.title("hyperparameter vs AUC")
plt.grid()
plt.show()
```

```
100%|██████████| 14/14 [00:02<00:00,  5.06it/s]
```



In [ ]:
```
 From above 2 cases (with class_prior[0.5,0.5] and class_prior=None) we can conclude that best_alpha (log alpha) is 1 as
cv_auc is maximum and also gap between train_auc and cv_auc is miminum
```

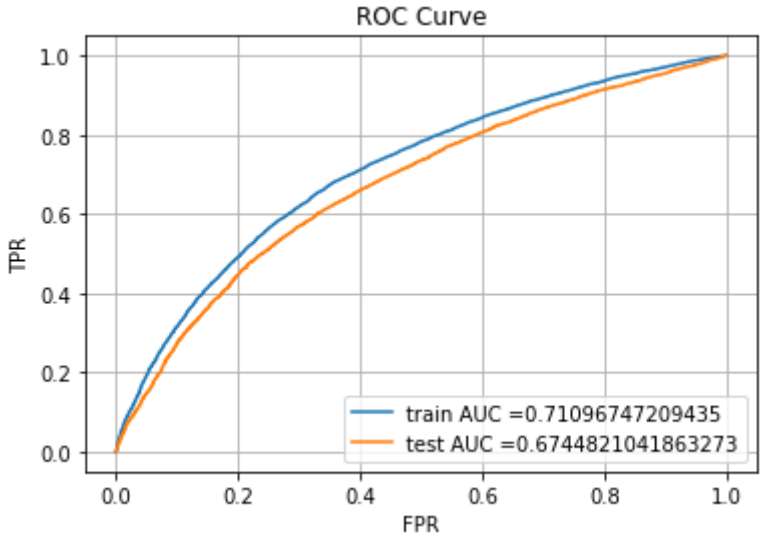
In [ ]:
```
best log_alpha is 1
so best alpha is 10
```

In [130]:
```
best_alpha=10
```

best_alpha=10

PLotting ROC Curve on both train and test data

In [131]:
```
import numpy as np
from sklearn import metrics


neigh = MultinomialNB(alpha = best_alpha, class_prior = [0.5,0.5])
neigh.fit(X_tr, y_train)

y_train_pred = neigh.predict_proba( X_tr)[:, 1]
y_test_pred = neigh.predict_proba(X_te)[:, 1]

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

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

#Plot curve :

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(metrics.auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(metrics.auc(test_fpr, test_tpr)))
plt.legend(loc='lower right')
plt.xlabel("FPR")
plt.ylabel("TPR")
plt.title("ROC Curve")
plt.grid()
plt.show()
```



Finding optimal threshold

In [133]:
```
#finding best threshold : https://stats.stackexchange.com/questions/123124/how-to-determine-the-optimal-threshold-for-a-classifier-and-generate-roc-curve
from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds = roc_curve(y_test, y_test_pred)
optimal_idx = np.argmax(tpr - fpr)
optimal_threshold = thresholds[optimal_idx]
print("Threshold value is:", np.round(optimal_threshold,3))
```

```
Threshold value is: 0.801
```

In [134]:
```
def predict(proba, threshould):
    predictions = []
    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
prediction = predict(y_test_pred,optimal_threshold)
```

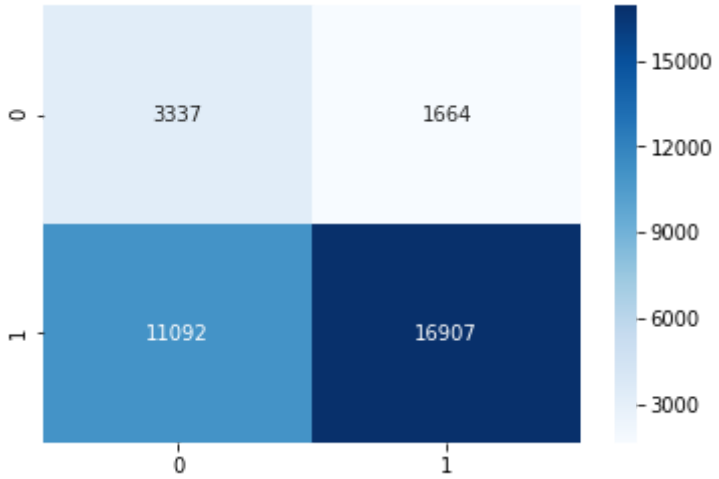
confusion matrix

In [135]:
```
from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(y_test,prediction)
print('Confusion matrix : \n',matrix)
```

```
Confusion matrix :
 [[ 3337  1664]
 [11092 16907]]
```

In [136]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(matrix, annot=True,fmt="d",cmap='Blues')
```

Out[136]: <matplotlib.axes._subplots.AxesSubplot at 0x237b0a45c50>



set 2

In [258]:
```
from scipy.sparse import hstack
X_tr = hstack((X_train_essay_tfidf, X_train_teacher_ohe, X_train_grade_ohe, X_train_state_ohe,X_train_category_ohe,X_train_subcategory_ohe,X_train_price_norm,X_train_teachernumber_norm)).tocsr()
X_cr = hstack((X_cv_essay_tfidf, X_cv_teacher_ohe, X_cv_grade_ohe, X_cv_state_ohe,X_cv_category_ohe,X_cv_subcategory_ohe,X_cv_price_norm,X_cv_teachernumber_norm)).tocsr()
X_te = hstack((X_test_essay_tfidf, X_test_teacher_ohe, X_test_grade_ohe, X_test_state_ohe,X_test_category_ohe,X_test_subcategory_ohe,X_test_price_norm,X_test_teachernumber_norm)).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
(44890, 5491) (44890,)
(22110, 5491) (22110,)
(33000, 5491) (33000,)
====================================================================================================
```

Applying Naive Bayes on Set 2

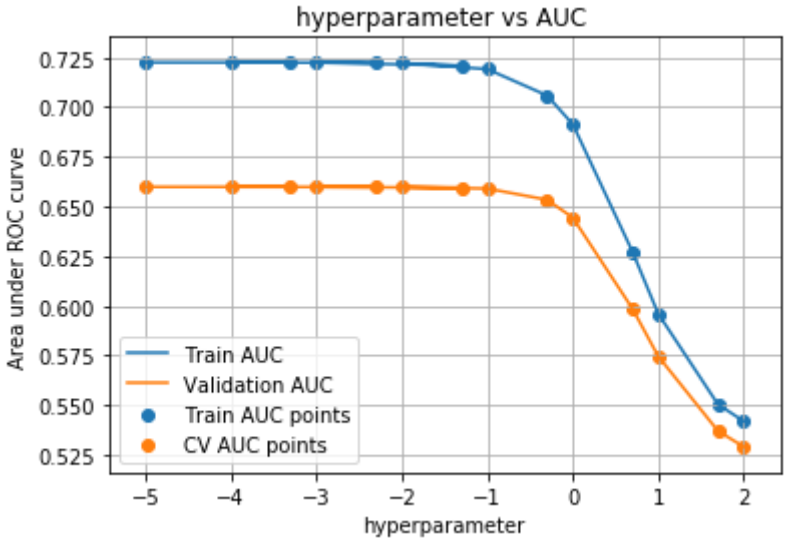
In [157]:
```
alpha=[0.00001,0.0005, 0.0001,0.005,0.001,0.05,0.01,0.1,0.5,1,5,10,50,100]
from math import log
import numpy
log_alpha=[numpy.log10(y) for y in alpha]
log_alpha
```

Out[157]:
```
[-5.0,
 -3.3010299956639813,
 -4.0,
 -2.3010299956639813,
 -3.0,
 -1.3010299956639813,
 -2.0,
 -1.0,
 -0.3010299956639812,
 0.0,
 0.6989700043360189,
 1.0,
 1.6989700043360187,
 2.0]
```

Finding best alpha

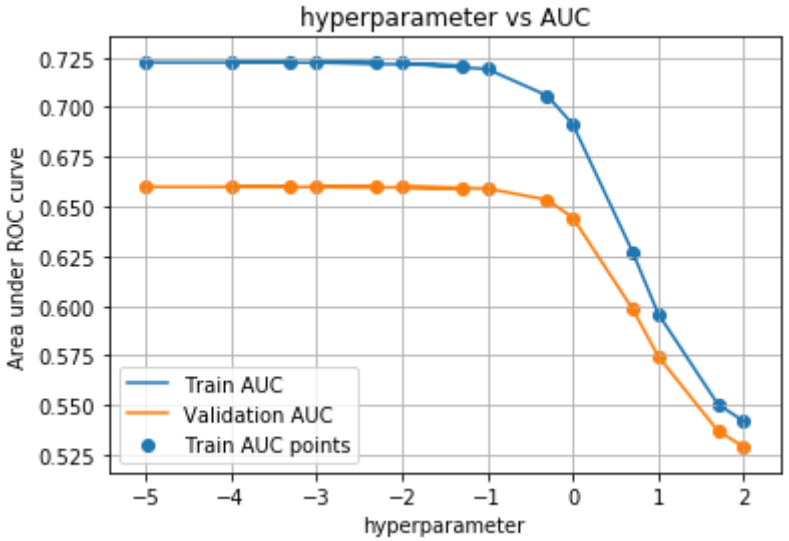
In [165]:
```
from sklearn.naive_bayes import MultinomialNB
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score

train_auc = []
cv_auc = []

for i in tqdm(alpha):
    neigh = MultinomialNB(alpha = i, class_prior = [0.5,0.5])
    neigh.fit(X_tr, y_train)

    y_train_pred = neigh.predict_proba( X_tr)[:, 1]
    y_cv_pred = neigh.predict_proba(X_cr)[:, 1]


    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))

plt.plot(log_alpha, train_auc, label='Train AUC')
plt.plot(log_alpha, cv_auc, label='Validation AUC')

plt.scatter(log_alpha, train_auc, label='Train AUC points')
plt.scatter(log_alpha, cv_auc, label='CV AUC points')

plt.legend()
plt.xlabel("hyperparameter")
plt.ylabel("Area under ROC curve")
plt.title("hyperparameter vs AUC ")
plt.grid()
plt.show()
```

```
100%|██████████| 14/14 [00:02<00:00,  6.16it/s]
```



In [159]:
```
from sklearn.naive_bayes import MultinomialNB
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score

train1_auc = []
cv1_auc = []

for i in tqdm(alpha):
    neigh = MultinomialNB(alpha = i,fit_prior= True, class_prior = None)
    neigh.fit(X_tr, y_train)

    y_train_pred = neigh.predict_proba( X_tr)[:, 1]
    y_cv_pred = neigh.predict_proba(X_cr)[:, 1]


    train1_auc.append(roc_auc_score(y_train,y_train_pred))
    cv1_auc.append(roc_auc_score(y_cv, y_cv_pred))

plt.plot(log_alpha, train1_auc, label='Train AUC')
plt.plot(log_alpha, cv1_auc, label='Validation AUC')

plt.scatter(log_alpha, train1_auc, label='Train AUC points')
plt.scatter(log_alpha, cv1_auc)

plt.legend()
plt.xlabel("hyperparameter")
plt.ylabel("Area under ROC curve")
plt.title("hyperparameter vs AUC")
plt.grid()
plt.show()
```

```
100%|██████████| 14/14 [00:02<00:00,  6.28it/s]
```

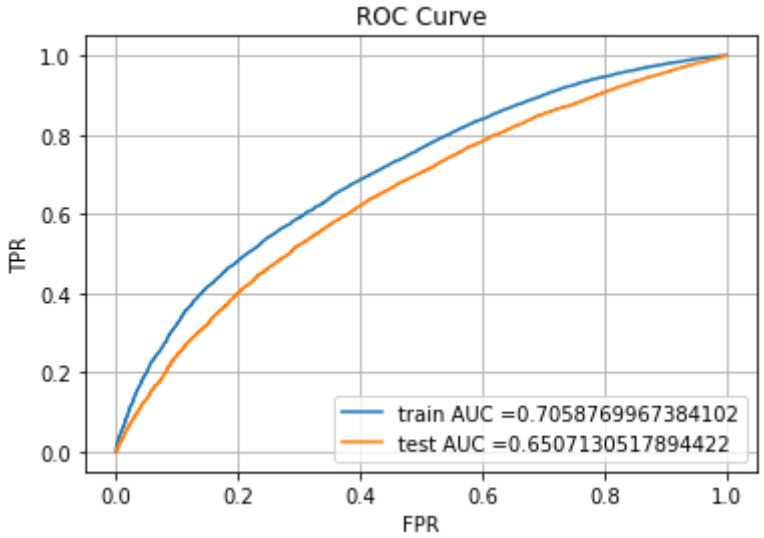


In [ ]:
```
From above 2 cases (with class_prior[0.5,0.5] and class_prior=None) we can conclude that best_alpha (log alpha)
is -0.3010299956639812 as
cv_auc is maximum and also gap between train_auc and cv_auc is miminum
```

In [168]:
```
best_alpha=0.5
```

Plotting ROC curve on both train and test data

In [269]:
```
import numpy as np
from sklearn import metrics


neigh = MultinomialNB(alpha = best_alpha, class_prior = [0.5,0.5])
neigh.fit(X_tr, y_train)

y_train_pred = neigh.predict_proba( X_tr)[:, 1]
y_test_pred = neigh.predict_proba(X_te)[:, 1]

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

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

#Plot curve :

plt.plot(train_fpr, train_tpr, label="train AUC ="+str(metrics.auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(metrics.auc(test_fpr, test_tpr)))
plt.legend(loc='lower right')
plt.xlabel("FPR")
plt.ylabel("TPR")
plt.title("ROC Curve")
plt.grid()
plt.show()
```



Finding optimal threshold

In [170]:
```
#https://stackoverflow.com/questions/28719067/roc-curve-and-cut-off-point-python
from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds = roc_curve(y_test, y_test_pred)
optimal_idx = np.argmax(tpr - fpr)
optimal_threshold = thresholds[optimal_idx]
print("Threshold value is:", np.round(optimal_threshold,3))
```

```
Threshold value is: 0.539
```

In [171]:
```
def predict(proba, threshould):
    predictions = []
    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
prediction = predict(y_test_pred,optimal_threshold)
```

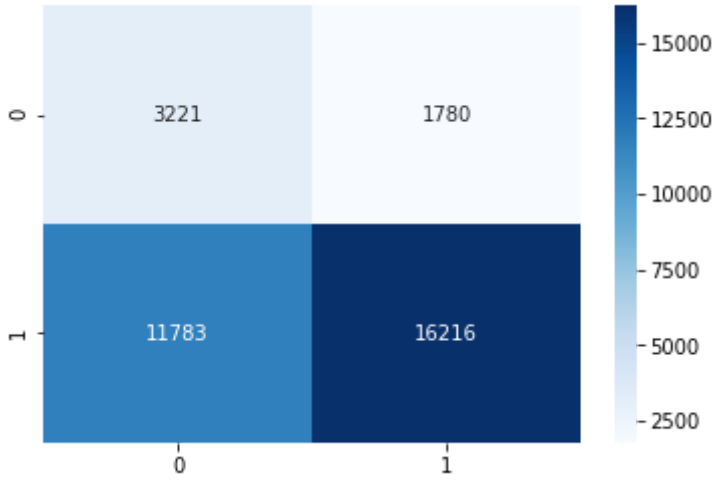
confusion matrix

In [172]:
```
from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(y_test,prediction)
print('Confusion matrix : \n',matrix)
```

```
Confusion matrix :
 [[ 3221  1780]
 [11783 16216]]
```

In [173]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(matrix, annot=True,fmt="d",cmap='Blues')
```

Out[173]: <matplotlib.axes._subplots.AxesSubplot at 0x237a2583198>



Find Top Features

In [276]:
```
all_feature_name=[]
```

In [277]:
```
all_feature_name.extend(vectorizer2.get_feature_names())
all_feature_name.extend(vectorizer3.get_feature_names())
all_feature_name.extend(vectorizer4.get_feature_names())
all_feature_name.extend(vectorizer5.get_feature_names())
all_feature_name.extend(vectorizer6.get_feature_names())
all_feature_name.extend(vectorizer7.get_feature_names())
all_feature_name.append('price')
all_feature_name.append('teacher_number_of_previously_posted_projects')
```

In [257]:
```
len(all_feature_name)
```

Out[257]: 5491

In [278]:
```
totalfeatures=len(all_feature_name)
```

set2 top 20 positive features

In [279]:
```
neigh = MultinomialNB(alpha =0.5,fit_prior= True, class_prior = None)
neigh.fit(X_tr,y_train)

#https://github.com/lvsreddy/Naive-Bayes-on-Donors-Choose-Dataset/blob/master/Naive%20Bayes%20on%20Donors%20choose.ipynb

bow_features_probs = []
for a in range(totalfeatures):
    bow_features_probs.append(neigh.feature_log_prob_[1,a] )
```

In [280]:
```
#https://www.geeksforgeeks.org/python-pandas-dataframe-sort_values-set-1/

final_bow_features1 = pd.DataFrame({'feature_prob_estimates' : bow_features_probs, 'feature_names': all_feature_name})

final_bow_features1.sort_values("feature_prob_estimates", axis = 0, ascending = False,
                 inplace = True, na_position ='last')
final_bow_features1.head(20)
```

Out[280]:

	feature_prob_estimates	feature_names
5002	-3.243266	mrs
5008	-3.502716	grades_prek_2
5003	-3.641740	ms
5005	-3.686371	grades_3_5
5083	-4.115813	literacy_language
5006	-4.478611	grades_6_8
5091	-4.499325	math_science
5013	-4.557125	ca
4363	-4.574568	students
5087	-4.593633	literacy_language_math_science
5007	-4.916853	grades_9_12
5001	-4.949692	mr
5068	-4.993317	health_sports
5408	-5.001064	literacy
5410	-5.154231	literacy_mathematics
5043	-5.296893	ny
5052	-5.336788	tx
5018	-5.480416	fl
5422	-5.500149	literature_writing_mathematics
5409	-5.563312	literacy_literature_writing

set2 top 20 negative features

In [281]:
```
neigh = MultinomialNB(alpha =0.5,fit_prior= True, class_prior = None)
neigh.fit(X_tr,y_train)
bow_features_probs = []
for a in range(totalfeatures):
    bow_features_probs.append(neigh.feature_log_prob_[0,a] )
```

In [283]:
```
final_bow_features2 = pd.DataFrame({'feature_prob_estimates' : bow_features_probs, 'feature_names': all_feature_name})

final_bow_features2.sort_values("feature_prob_estimates", axis = 0, ascending = False,
                 inplace = True, na_position ='last')
final_bow_features2.head(20)
```

Out[283]:

	feature_prob_estimates	feature_names
5002	-3.304794	mrs
5008	-3.516807	grades_prek_2
5003	-3.643076	ms
5005	-3.741890	grades_3_5
5083	-4.268036	literacy_language
5091	-4.339296	math_science
5006	-4.434812	grades_6_8
4363	-4.602013	students
5013	-4.659650	ca
5087	-4.779626	literacy_language_math_science
5007	-4.880112	grades_9_12
5001	-4.884349	mr
5068	-4.979305	health_sports
5052	-5.048576	tx
5408	-5.293004	literacy
5410	-5.343233	literacy_mathematics
5018	-5.375090	fl
5043	-5.461996	ny
5432	-5.508456	mathematics
3954	-5.566450	school

conclusion

In [272]:
```
#http://zetcode.com/python/prettytable/

from prettytable import PrettyTable

x = PrettyTable()

x.field_names = ["Vectorizer", "Model", "Hyperparameter: Alpha", "Test AUC"]

x.add_row(["BOW", "Multinomial Naive Bayes", 10, 0.67])
x.add_row(["TF-IDF", "Multinomial Naive Bayes", 0.5, 0.65])

print(x)
```

```
+------------+------------------------+----------------------+----------+
| Vectorizer |          Model         | Hyperparameter: Alpha | Test AUC |
+------------+------------------------+----------------------+----------+
|    BOW     | Multinomial Naive Bayes |          10          |   0.67   |
|   TF-IDF   | Multinomial Naive Bayes |         0.5          |   0.65   |
+------------+------------------------+----------------------+----------+
```

In [284]:
```
!jupyter nbconvert --to html NB_solve.ipynb
```

```
[NbConvertApp] Converting notebook NB_solve.ipynb to html
[NbConvertApp] Writing 686174 bytes to NB_solve.html
```