DonorsChoose

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

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

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

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

About the DonorsChoose Data Set

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

Feature	Description
`project_id`	A unique identifier for the proposed project. **Example**
`project_title`	Title of the project. **Examples:** • `Art Will Make You Happy!` • `First Grade Fun`
`project_grade_category`	Grade level of students for which the project is targete enumerated values: • `Grades PreK-2` • `Grades 3-5` • `Grades 6-8` • `Grades 9-12`
`project_subject_categories`	One or more (comma-separated) subject categories f following enumerated list of values: • `Applied Learning` • `Care & Hunger` • `Health & Sports` • `History & Civics` • `Literacy & Language` • `Math & Science` • `Music & The Arts` • `Special Needs` • `Warmth` **Examples:** • `Music & The Arts` • `Literacy & Language, Math & Science`
`school_state`	State where school is located (Two-letter U.S. postal ((https://en.wikipedia.org/wiki/List_of_U.S._state_abbr **Example:** `WY`
`project_subject_subcategories`	One or more (comma-separated) subject subcategori **Examples:** • `Literacy` • `Literature & Writing, Social Sciences`
`project_resource_summary`	An explanation of the resources needed for the projec • `My students need hands on literacy mater sensory needs!`

Feature	Description
project_essay_1	First application essay*
project_essay_2	Second application essay*
project_essay_3	Third application essay*
project_essay_4	Fourth application essay*
project_submitted_datetime	Datetime when project application was submitted. **Ex** 12:43:56.245
teacher_id	A unique identifier for the teacher of the proposed pro bdf8baa8fedef6bfeec7ae4ff1c15c56
teacher_prefix	Teacher's title. One of the following enumerated value • nan • Dr. • Mr. • Mrs. • Ms. • Teacher.
teacher_number_of_previously_posted_projects	Number of project applications previously submitted b **Example:** 2

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

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

Feature	Description
id	A project_id value from the train.csv file. **Example:** p036502
description	Desciption of the resource. **Example:** Tenor Saxophone Reeds, Box of 25
quantity	Quantity of the resource required. **Example:** 3
price	Price of the resource required. **Example:** 9.95

Note: Many projects require multiple resources. The id value corresponds to a project_id in train.csv, so you use it as a key to retrieve all resources needed for a project:

The data set contains the following label (the value you will attempt to predict):

Label	Description
project_is_approved	A binary flag indicating whether DonorsChoose approved the project. A value of 0 indicates the project was not approved, and a value of 1 indicates the project was approved.

Notes on the Essay Data

Prior to May 17, 2016, the prompts for the essays were as follows:

- __project_essay_1:__ "Introduce us to your classroom"
- __project_essay_2:__ "Tell us more about your students"
- __project_essay_3:__ "Describe how your students will use the materials you're requesting"
- __project_essay_3:__ "Close by sharing why your project will make a difference"

Starting on May 17, 2016, the number of essays was reduced from 4 to 2, and the prompts for the first 2 essays were changed to the following:

- __project_essay_1:__ "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
- __project_essay_2:__ "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

For all projects with project_submitted_datetime of 2016-05-17 and later, the values of project_essay_3 and project_essay_4 will be NaN.

In [1]:

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

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

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer

import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

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

from tqdm import tqdm
import os

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

1.1 Reading Data

In [2]:

```
project_data = pd.read_csv('train_data.csv')
resource_data = pd.read_csv('resources.csv')
```

In [3]:

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

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

In [4]:

```
print("Number of data points in train data", resource_data.shape)
print(resource_data.columns.values)
resource_data.head(2)
```

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

Out[4]:

	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 [5]:

```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

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

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

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

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

PREPROCESSING OF PROJECT GRADE CATEGORY

In [6]:

```
grade_categories=list(project_data['project_grade_category'].values)
clean_grades=[]
for i in grade_categories:
    temp=""
    for j in i.split(','):
        j=j.replace(' ','_')
        j=j.replace('-','_')
        temp+=j
        clean_grades.append(temp)
project_data['clean_grades']=clean_grades
project_data.drop(['project_grade_category'],axis=1,inplace=True)
```

1.3 preprocessing of project_subject_subcategories

In [7]:

```
sub_catogories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/a/47
301924/4084039

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

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

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

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

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

1.3 Text preprocessing

In [8]:

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

In [9]:

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

Out[9]:

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

In [10]:

```
#### 1.4.2.3 Using Pretrained Models: TFIDF weighted W2V
```

In [11]:

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

My students are English learners that are working on English as their seco
nd or third languages. We are a melting pot of refugees, immigrants, and n
ative-born Americans bringing the gift of language to our school. \r\n\r\n
We have over 24 languages represented in our English Learner program with
students at every level of mastery. We also have over 40 countries repres
ented with the families within our school. Each student brings a wealth o
f knowledge and experiences to us that open our eyes to new cultures, beli
efs, and respect.\"The limits of your language are the limits of your worl
d.\"-Ludwig Wittgenstein Our English learner's have a strong support syst
em at home that begs for more resources. Many times our parents are learn
ing to read and speak English along side of their children. Sometimes thi
s creates barriers for parents to be able to help their child learn phonet
ics, letter recognition, and other reading skills.\r\n\r\nBy providing the
se dvd's and players, students are able to continue their mastery of the E
nglish language even if no one at home is able to assist. All families wi
th students within the Level 1 proficiency status, will be a offered to be
a part of this program. These educational videos will be specially chosen
by the English Learner Teacher and will be sent home regularly to watch.
The videos are to help the child develop early reading skills.\r\n\r\nPare
nts that do not have access to a dvd player will have the opportunity to c
heck out a dvd player to use for the year. The plan is to use these video
s and educational dvd's for the years to come for other EL students.\r\nna
nnan

==

The 51 fifth grade students that will cycle through my classroom this year
all love learning, at least most of the time. At our school, 97.3% of the
students receive free or reduced price lunch. Of the 560 students, 97.3% a
re minority students. \r\nThe school has a vibrant community that loves to
get together and celebrate. Around Halloween there is a whole school parad
e to show off the beautiful costumes that students wear. On Cinco de Mayo
we put on a big festival with crafts made by the students, dances, and gam
es. At the end of the year the school hosts a carnival to celebrate the ha
rd work put in during the school year, with a dunk tank being the most pop
ular activity.My students will use these five brightly colored Hokki stool
s in place of regular, stationary, 4-legged chairs. As I will only have a
total of ten in the classroom and not enough for each student to have an i
ndividual one, they will be used in a variety of ways. During independent
reading time they will be used as special chairs students will each use on
occasion. I will utilize them in place of chairs at my small group tables
during math and reading times. The rest of the day they will be used by th
e students who need the highest amount of movement in their life in order
to stay focused on school.\r\n\r\nWhenever asked what the classroom is mis
sing, my students always say more Hokki Stools. They can't get their fill
of the 5 stools we already have. When the students are sitting in group wi
th me on the Hokki Stools, they are always moving, but at the same time do
ing their work. Anytime the students get to pick where they can sit, the H
okki Stools are the first to be taken. There are always students who head
over to the kidney table to get one of the stools who are disappointed as
there are not enough of them. \r\n\r\nWe ask a lot of students to sit for
7 hours a day. The Hokki stools will be a compromise that allow my student
s to do desk work and move at the same time. These stools will help studen
ts to meet their 60 minutes a day of movement by allowing them to activate
their core muscles for balance while they sit. For many of my students, th
ese chairs will take away the barrier that exists in schools for a child w
ho can't sit still.nannan

==

How do you remember your days of school? Was it in a sterile environment w
ith plain walls, rows of desks, and a teacher in front of the room? A typi
cal day in our room is nothing like that. I work hard to create a warm inv
iting themed room for my students look forward to coming to each day.\r\n
\r\nMy class is made up of 28 wonderfully unique boys and girls of mixed r

aces in Arkansas.\r\nThey attend a Title I school, which means there is a high enough percentage of free and reduced-price lunch to qualify. Our school is an \"open classroom\" concept, which is very unique as there are no walls separating the classrooms. These 9 and 10 year-old students are very eager learners; they are like sponges, absorbing all the information and experiences and keep on wanting more.With these resources such as the comfy red throw pillows and the whimsical nautical hanging decor and the blue fish nets, I will be able to help create the mood in our classroom setting to be one of a themed nautical environment. Creating a classroom environment is very important in the success in each and every child's education. The nautical photo props will be used with each child as they step foot into our classroom for the first time on Meet the Teacher evening. I'll take pictures of each child with them, have them developed, and then hung in our classroom ready for their first day of 4th grade. This kind gesture will set the tone before even the first day of school! The nautical thank you cards will be used throughout the year by the students as they create thank you cards to their team groups.\r\n\r\nYour generous donations will help me to help make our classroom a fun, inviting, learning environment from day one.\r\n\r\nIt costs lost of money out of my own pocket on resources to get our classroom ready. Please consider helping with this project to make our new school year a very successful one. Thank you!nannan
==
My kindergarten students have varied disabilities ranging from speech and language delays, cognitive delays, gross/fine motor delays, to autism. They are eager beavers and always strive to work their hardest working past their limitations. \r\n\r\nThe materials we have are the ones I seek out for my students. I teach in a Title I school where most of the students receive free or reduced price lunch. Despite their disabilities and limitations, my students love coming to school and come eager to learn and explore. Have you ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting? This is how my kids feel all the time. The want to be able to move as they learn or so they say.Wobble chairs are the answer and I love then because they develop their core, which enhances gross motor and in Turn fine motor skills. \r\nThey also want to learn through games, my kids don't want to sit and do worksheets. They want to learn to count by jumping and playing. Physical engagement is the key to our success. The number toss and color and shape mats can make that happen. My students will forget they are doing work and just have the fun a 6 year old deserves.nannan
==
The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires. -William A. Ward\r\n\r\nMy school has 803 students which is makeup is 97.6% African-American, making up the largest segment of the student body. A typical school in Dallas is made up of 23.2% African-American students. Most of the students are on free or reduced lunch. We aren't receiving doctors, lawyers, or engineers children from rich backgrounds or neighborhoods. As an educator I am inspiring minds of young children and we focus not only on academics but one smart, effective, efficient, and disciplined students with good character.In our classroom we can utilize the Bluetooth for swift transitions during class. I use a speaker which doesn't amplify the sound enough to receive the message. Due to the volume of my speaker my students can't hear videos or books clearly and it isn't making the lessons as meaningful. But with the blue tooth speaker my students will be able to hear and I can stop, pause and replay it at any time.\r\nThe cart will allow me to have more room for storage of things that are needed for the day and has an extra part to it I can use. The table top chart has all of the letter, words and pictures for students to learn about different letters and it is more accessible.nannan
==

In [13]:

```
# https://stackoverflow.com/a/47091490/4084039
import re

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

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

In [14]:

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

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

In [15]:

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

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

In [16]:

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

```
My kindergarten students have varied disabilities ranging from speech and
language delays cognitive delays gross fine motor delays to autism They ar
e eager beavers and always strive to work their hardest working past their
limitations The materials we have are the ones I seek out for my students
I teach in a Title I school where most of the students receive free or red
uced price lunch Despite their disabilities and limitations my students lo
ve coming to school and come eager to learn and explore Have you ever felt
like you had ants in your pants and you needed to groove and move as you w
ere in a meeting This is how my kids feel all the time The want to be able
to move as they learn or so they say Wobble chairs are the answer and I lo
ve then because they develop their core which enhances gross motor and in
Turn fine motor skills They also want to learn through games my kids do no
t want to sit and do worksheets They want to learn to count by jumping and
playing Physical engagement is the key to our success The number toss and
color and shape mats can make that happen My students will forget they are
doing work and just have the fun a 6 year old deserves nannan
```

In [17]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've",\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', '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', 'each', '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', "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', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [18]:

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

```
100%|████████████████████████████████████████████████████████████████████████████████| 109248/109248 [02:19<00:00, 780.60it/s]
```

In [19]:

```
# after preprocesing
preprocessed_essays[20000]
```

Out[19]:

```
'kindergarten students varied disabilities ranging speech language delays
cognitive delays gross fine motor delays autism eager beavers always striv
e work hardest working past limitations materials ones seek students teach
title school students receive free reduced price lunch despite disabilitie
s limitations students love coming school come eager learn explore ever fe
lt like ants pants needed groove move meeting kids feel time want able mov
e learn say wobble chairs answer love develop core enhances gross motor tu
rn fine motor skills also want learn games kids not want sit worksheets wa
nt learn count jumping playing physical engagement key success number toss
color shape mats make happen students forget work fun 6 year old deserves
nannan'
```

1.4 Preprocessing of `project_title`

In [20]:

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

```
100%|████████████████████████████████████████████████████████████████
██| 109248/109248 [00:05<00:00, 21544.09it/s]
```

1.5 Preparing data for models

In [21]:

```
project_data.columns
```

Out[21]:

```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', '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_approve
d',
       'clean_categories', 'clean_grades', 'clean_subcategories', 'essa
y'],
      dtype='object')
```

we are going to consider

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

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

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

1.5.1 Vectorizing Categorical data

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

In [21]:

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

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

In [22]:

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

```
['Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvemen
t', 'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'Nutrition
Education', 'Warmth', 'Care_Hunger', 'SocialSciences', 'PerformingArts',
'CharacterEducation', 'TeamSports', 'Other', 'College_CareerPrep', 'Musi
c', 'History_Geography', 'Health_LifeScience', 'EarlyDevelopment', 'ESL',
'Gym_Fitness', 'EnvironmentalScience', 'VisualArts', 'Health_Wellness', 'A
ppliedSciences', 'SpecialNeeds', 'Literature_Writing', 'Mathematics', 'Lit
eracy']
Shape of matrix after one hot encodig  (109248, 30)
```

In [23]:

```
# you can do the similar thing with state, teacher_prefix and project_grade_category al
so
```

In [24]:

```
#vectorizing school state
vectorizer=CountVectorizer(vocabulary=project_data['school_state'].unique(),lowercase=T
rue,binary=True)
school_state_one_hot=vectorizer.fit_transform(project_data['school_state'].values)
print(vectorizer.get_feature_names())
print("shape of matrix after one hot encoding",school_state_one_hot.shape)
```

```
['IN', 'FL', 'AZ', 'KY', 'TX', 'CT', 'GA', 'SC', 'NC', 'CA', 'NY', 'OK',
'MA', 'NV', 'OH', 'PA', 'AL', 'LA', 'VA', 'AR', 'WA', 'WV', 'ID', 'TN', 'M
S', 'CO', 'UT', 'IL', 'MI', 'HI', 'IA', 'RI', 'NJ', 'MO', 'DE', 'MN', 'M
E', 'WY', 'ND', 'OR', 'AK', 'MD', 'WI', 'SD', 'NE', 'NM', 'DC', 'KS', 'M
T', 'NH', 'VT']
shape of matrix after one hot encoding (109248, 51)
```

In [25]:

```
#vectorizing project grade
vectorizer=CountVectorizer(vocabulary=project_data['clean_grades'].unique(),lowercase=F
alse,binary=True)
project_grade_one_hot=vectorizer.fit_transform(project_data['clean_grades'].values)
print(vectorizer.get_feature_names())
print("shape of matrix after one hot encoding ",project_grade_one_hot.shape)
```

```
['Grades_PreK_2', 'Grades_6_8', 'Grades_3_5', 'Grades_9_12']
shape of matrix after one hot encoding  (109248, 4)
```

In [26]:

```
#vectorizing teacher prefix
x=project_data['teacher_prefix'].fillna('')
vectorizer = CountVectorizer()

teacher_prefix_one_hot = vectorizer.fit_transform(x.values)
print(vectorizer.get_feature_names())
print("Shape of matrix after one hot encodig ",teacher_prefix_one_hot.shape)
```

```
['dr', 'mr', 'mrs', 'ms', 'teacher']
Shape of matrix after one hot encodig  (109248, 5)
```

1.5.2 Vectorizing Text data

1.5.2.1 Bag of words

In [27]:

```
# We are considering only the words which appeared in at least 10 documents(rows or pro
jects).
vectorizer = CountVectorizer(min_df=10)
text_bow = vectorizer.fit_transform(preprocessed_essays)
print("Shape of matrix after one hot encodig ",text_bow.shape)
```

```
Shape of matrix after one hot encodig  (109248, 16512)
```

In [28]:

```
# you can vectorize the title also
# before you vectorize the title make sure you preprocess it
vectorizer=CountVectorizer(min_df=10)
title_bow=vectorizer.fit_transform(preprocessed_title)
print('shape of matrix after vectorizing',title_bow.shape)
```

```
shape of matrix after vectorizing (109248, 3222)
```

1.5.2.2 TFIDF vectorizer

In [29]:

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

```
Shape of matrix after one hot encodig  (109248, 16512)
```

In [30]:

```
vectorizer=TfidfVectorizer(min_df=10)
title_tfidf=vectorizer.fit_transform(preprocessed_title)
print("shape of matrix after vectorizing",title_tfidf.shape)
```

```
shape of matrix after vectorizing (109248, 3222)
```

1.5.2.3 Using Pretrained Models: Avg W2V

In [0]:

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

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

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

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

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

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

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

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


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

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


'''
```

Out[0]:

```
'\n# Reading glove vectors in python: https://stackoverflow.com/a/3823034
9/4084039\ndef loadGloveModel(gloveFile):\n    print ("Loading Glove Mode
l")\n    f = open(gloveFile,\'r\', encoding="utf8")\n    model = {}\n    f
or line in tqdm(f):\n        splitLine = line.split()\n        word = spli
tLine[0]\n        embedding = np.array([float(val) for val in splitLine
[1:]])\n        model[word] = embedding\n    print ("Done.",len(model)," w
ords loaded!")\n    return model\nmodel = loadGloveModel(\'glove.42B.300d.
txt\')\n\n# ==========================\nOutput:\n    \nLoading Glove Mod
el\n1917495it [06:32, 4879.69it/s]\nDone. 1917495  words loaded!\n\n# ====
========================\n\nwords = []\nfor i in preproced_texts:\n    wor
ds.extend(i.split(\' \'))\n\nfor i in preproced_titles:\n    words.extend
(i.split(\' \'))\nprint("all the words in the coupus", len(words))\nwords
= set(words)\nprint("the unique words in the coupus", len(words))\n\ninter
_words = set(model.keys()).intersection(words)\nprint("The number of words
that are present in both glove vectors and our coupus",       len(inter_wo
rds),"(",np.round(len(inter_words)/len(words)*100,3),"%)")\n\nwords_courpu
s = {}\nwords_glove = set(model.keys())\nfor i in words:\n    if i in word
s_glove:\n        words_courpus[i] = model[i]\nprint("word 2 vec length",
len(words_courpus))\n\n\n# stronging variables into pickle files python: h
ttp://www.jessicayung.com/how-to-use-pickle-to-save-and-load-variables-in-
python/\n\nimport pickle\nwith open(\'glove_vectors\', \'wb\') as f:\n
pickle.dump(words_courpus, f)\n\n\n'
```

In [31]:

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

In [32]:

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

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

```
100%|████████████████████████████████████████████████████████████████
███| 109248/109248 [01:04<00:00, 1704.32it/s]

109248
300
```

In [33]:

```
title_avg_w2v_vectors = []; # the avg-w2v for each sentence/review is stored in this li
st
for sentence in tqdm(preprocessed_title): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    title_avg_w2v_vectors.append(vector)

print(len(title_avg_w2v_vectors))
print(len(title_avg_w2v_vectors[0]))
```

```
100%|████████████████████████████████████████████████████████████████
██| 109248/109248 [00:02<00:00, 41473.29it/s]

109248
300
```

1.5.2.3 Using Pretrained Models: TFIDF weighted W2V

In [34]:

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

In [35]:

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

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

```
100%|████████████████████████████████████████████████████████████████
████| 109248/109248 [04:52<00:00, 372.92it/s]

109248
300
```

In [36]:

```
# Similarly you can vectorize for title also
title_tfidf_w2v_vectors = []; # the avg-w2v for each sentence/review is stored in this
 list
for sentence in tqdm(preprocessed_title): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf value((sen
tence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split())) # ge
tting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    title_tfidf_w2v_vectors.append(vector)

print(len(title_tfidf_w2v_vectors))
print(len(title_tfidf_w2v_vectors[0]))
```

```
100%|████████████████████████████████████████████████████████████████
██| 109248/109248 [00:04<00:00, 23429.60it/s]

109248
300
```

1.5.3 Vectorizing Numerical features

In [22]:

```
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'}).reset_i
ndex()
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

In [23]:

```
data1=project_data.drop(['id','teacher_id','project_essay_1','project_essay_2','project
_essay_3','project_essay_4'],axis=1)
data1.head(2)
data=data1[0:50000]
data[0:2]
```

Out[23]:

	Unnamed: 0	teacher_prefix	school_state	project_submitted_datetime	project_title
0	160221	Mrs.	IN	2016-12-05 13:43:57	Educational Support for English Learners at Home
1	140945	Mr.	FL	2016-10-25 09:22:10	Wanted: Projector for Hungry Learners

In [25]:

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

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

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

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

```
Mean : 298.1193425966608, Standard deviation : 367.49634838483496
```

In [26]:

```
price_standardized
```

Out[26]:

```
array([[-0.3905327 ],
       [ 0.00239637],
       [ 0.59519138],
       ...,
       [-0.15825829],
       [-0.61243967],
       [-0.51216657]])
```

In [27]:

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

# Now standardize the data with above maen and variance.
projects_standardized = projects_scalar.transform(project_data['teacher_number_of_previ
ously_posted_projects'].values.reshape(-1,1))
projects_standardized
```

```
C:\Users\HP\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ut
ils\validation.py:475: DataConversionWarning:

Data with input dtype int64 was converted to float64 by StandardScaler.


Mean : 298.1193425966608, Standard deviation : 367.49634838483496

C:\Users\HP\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ut
ils\validation.py:475: DataConversionWarning:

Data with input dtype int64 was converted to float64 by StandardScaler.
```

Out[27]:

```
array([[-0.40152481],
       [-0.14951799],
       [-0.36552384],
       ...,
       [-0.29352189],
       [-0.40152481],
       [-0.40152481]])
```

In [28]:

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

# Now standardize the data with above maen and variance.
quantity_standardized = projects_scalar.transform(project_data['quantity'].values.resha
pe(-1,1))
quantity_standardized
```

```
C:\Users\HP\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ut
ils\validation.py:475: DataConversionWarning:

Data with input dtype int64 was converted to float64 by StandardScaler.


Mean : 298.1193425966608, Standard deviation : 367.49634838483496

C:\Users\HP\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ut
ils\validation.py:475: DataConversionWarning:

Data with input dtype int64 was converted to float64 by StandardScaler.
```

Out[28]:

```
array([[ 0.23047132],
       [-0.60977424],
       [ 0.19227834],
       ...,
       [-0.4951953 ],
       [-0.03687954],
       [-0.45700232]])
```

1.5.4 Merging all the above features

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

In [35]:

```
print(categories_one_hot.shape)
print(sub_categories_one_hot.shape)
print(teacher_prefix_one_hot.shape)
print(school_state_one_hot.shape)
print(project_grade_one_hot.shape)
print(title_bow.shape)
print(text_bow.shape)
print(price_standardized.shape)
print(projects_standardized.shape)
```

```
(109248, 9)
(109248, 30)
(109248, 5)
(109248, 51)
(109248, 4)
(109248, 3222)
(109248, 16512)
(109248, 1)
(109248, 1)
```

In [36]:

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

Out[36]:

```
(109248, 16552)
```

In [29]:

```
y1=project_data['project_is_approved']
print(y1.shape)
y=y1[0:50000]
```

```
(109248,)
```

In [43]:

```
# please write all the code with proper documentation, and proper titles for each subse
ction
# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the reader
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label
```

Computing Sentiment Scores

In [28]:

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

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

sid = SentimentIntensityAnalyzer()

for_sentiment = 'a person is a person no matter how small dr seuss i teach the smallest
students with the biggest enthusiasm \
for learning my students learn in many different ways using all of our senses and multi
ple intelligences i use a wide range\
of techniques to help all my students succeed students in my class come from a variety
 of different backgrounds which makes\
for wonderful sharing of experiences and cultures including native americans our school
is a caring community of successful \
learners which can be seen through collaborative student project based learning in and
 out of the classroom kindergarteners \
in my class love to work with hands on materials and have many different opportunities
 to practice a skill before it is\
mastered having the social skills to work cooperatively with friends is a crucial aspec
t of the kindergarten curriculum\
montana is the perfect place to learn about agriculture and nutrition my students love
 to role play in our pretend kitchen\
in the early childhood classroom i have had several kids ask me can we try cooking with
real food i will take their idea \
and create common core cooking lessons where we learn important math and writing concep
ts while cooking delicious healthy \
food for snack time my students will have a grounded appreciation for the work that wen
t into making the food and knowledge \
of where the ingredients came from as well as how it is healthy for their bodies this p
roject would expand our learning of \
nutrition and agricultural cooking recipes by having us peel our own apples to make hom
emade applesauce make our own bread \
and mix up healthy plants from our classroom garden in the spring we will also create o
ur own cookbooks to be printed and \
shared with families students will gain math and literature skills as well as a life lo
ng enjoyment for healthy cooking \
nannan'
ss = sid.polarity_scores(for_sentiment)

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

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

```
C:\Users\HP\AppData\Local\Continuum\anaconda3\lib\site-packages\nltk\twitt
er\__init__.py:20: UserWarning:

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


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

Assignment 9: RF and GBDT

Response Coding: Example

> The response tabel is built only on train dataset. For a category which is not there in train data and present in test data, we will encode them with default values Ex: in our test data if have State: D then we encode it as [0.5, 0.05]

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

   - Set 1: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(BOW) + preprocessed_eassay (BOW)
   - Set 2: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF)+ preprocessed_eassay (TFIDF)
   - Set 3: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(AVG W2V)+ preprocessed_eassay (AVG W2V). Here for this set take **30K** datapoints only.
   - Set 4: categorical(instead of one hot encoding, try response coding (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/): use probability values), numerical features + project_title(TFIDF W2V)+ preprocessed_eassay (TFIDF W2V). Here for this set take **30K** datapoints only.

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

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

3. **Representation of results**

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

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

or

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

     seaborn heat maps (https://seaborn.pydata.org/generated/seaborn.heatmap.html) with rows as **n_estimators**, columns as **max_depth**, and values inside the cell representing **AUC Score**
   - You can choose either of the plotting techniques: 3d plot or heat map
   - Once after you found the best hyper parameter, you need to train your model with it, and find the AUC on test data and plot the ROC curve on both train and test.

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

4. **Conclusion**

   - You need to summarize the results at the end of the notebook, summarize it in the table format. To print out a table please refer to this prettytable library link (http://zetcode.com/python/prettytable/)

Note: Data Leakage

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

2. Random Forest and GBDT

2.1 Splitting data into Train and cross validation(or test): Stratified Sampling

In [0]:

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

In [30]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(data,y,test_size=0.30,stratify=y)
X_train,X_cv,y_train,y_cv=train_test_split(X_train,y_train,test_size=0.30,stratify=y_tr
ain)
print(X_train.shape,y_train.shape)
print(X_cv.shape,y_cv.shape)
print(X_test.shape,y_test.shape)
```

```
(24500, 14) (24500,)
(10500, 14) (10500,)
(15000, 14) (15000,)
```

2.2 Make Data Model Ready: encoding numerical, categorical features

In [0]:

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

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

RESPONSE CODING OF CATEGORICAL FEATURES

In [31]:

```
X_train.groupby('teacher_prefix').size()
```

Out[31]:

```
teacher_prefix
Mr.         2375
Mrs.       12813
Ms.         8797
Teacher      514
dtype: int64
```

In [32]:

```
#teacher_prefix_0=X_train['teacher_prefix'].where(X_train['project_is_approved']==0).value_counts()
teacher_prefix_1=X_train['teacher_prefix'].where(X_train['project_is_approved']==1).value_counts()
print(teacher_prefix_1)
```

```
Mrs.       10928
Ms.         7379
Mr.         1994
Teacher      418
Name: teacher_prefix, dtype: int64
```

In [33]:

```
#response coding of teacher prefix

pr_accepted={}
pr_rejected={}
for x in X_train['teacher_prefix'].unique():
    print('*'*50)
    print(x)
    total=X_train['teacher_prefix'][X_train['teacher_prefix']==x].count()
    print('Total projects ={}'.format(total))
    for y in X_train['project_is_approved'].unique():
        n=X_train['project_is_approved'][(X_train['teacher_prefix']==x)&(X_train['proje
ct_is_approved']==y)].count()
        if y:
            print('Approved projects ={}'.format(n))
            print('Approved probability={}'.format(n/total))
            pr_accepted.update({x:n/total})
        else:
            print('Rejected projects ={}'.format(n))
            print('Rejected probability={}'.format(n/total))
            pr_rejected.update({x:n/total})
```

```
**************************************************
Ms.
Total projects =8797
Rejected projects =1418
Rejected probability=0.1611913152210981
Approved projects =7379
Approved probability=0.8388086847789019
**************************************************
Mrs.
Total projects =12813
Rejected projects =1885
Rejected probability=0.14711621009911807
Approved projects =10928
Approved probability=0.8528837899008819
**************************************************
Mr.
Total projects =2375
Rejected projects =381
Rejected probability=0.16042105263157894
Approved projects =1994
Approved probability=0.8395789473684211
**************************************************
Teacher
Total projects =514
Rejected projects =96
Rejected probability=0.1867704280155642
Approved projects =418
Approved probability=0.8132295719844358
**************************************************
nan
Total projects =0
Rejected projects =0
Rejected probability=nan
Approved projects =0
Approved probability=nan
```

In [34]:

```
print(pr_rejected)
print(pr_accepted)
```

```
{'Ms.': 0.1611913152210981, 'Mrs.': 0.14711621009911807, 'Mr.': 0.16042105
263157894, 'Teacher': 0.1867704280155642, nan: nan}
{'Ms.': 0.8388086847789019, 'Mrs.': 0.8528837899008819, 'Mr.': 0.839578947
3684211, 'Teacher': 0.8132295719844358, nan: nan}
```

In [35]:

```
pr_accepted['Ms.']
```

Out[35]:

```
0.8388086847789019
```

In [40]:

```
p1=[]
p2=[]
for i in (X_train['teacher_prefix'].values):
    if i in pr_accepted:
        p1.append(pr_accepted[i])
        p2.append(pr_rejected[i])
    else:
        p1.append(0.5)
        p2.append(0.5)
print(len(p1))
print(len(p2))
```

```
24500
24500
```

In [42]:

```
p3=[]
p4=[]
for i in (X_cv['teacher_prefix'].values):
    if i in pr_accepted:
        p3.append(pr_accepted[i])
        p4.append(pr_rejected[i])
    else:
        p3.append(0.5)
        p4.append(0.5)
print(len(p3))
print(len(p4))
```

```
10500
10500
```

In [43]:

```
X_test['teacher_prefix'].unique()
```

Out[43]:

```
array(['Mrs.', 'Ms.', 'Mr.', 'Teacher'], dtype=object)
```

In [44]:

```
p5=[]
p6=[]
for i in (X_test['teacher_prefix'].values):
        p5.append(pr_accepted[i])
        p6.append(pr_rejected[i])
print(len(p5))
print(len(p6))
```

```
15000
15000
```

In [45]:

```
teacher_prefix_accepted=pd.DataFrame(p1)
print(teacher_prefix_accepted[0:5])
teacher_prefix_rejected=pd.DataFrame(p2)
print(teacher_prefix_rejected[0:5])
```

```
          0
0  0.838809
1  0.852884
2  0.838809
3  0.852884
4  0.838809
          0
0  0.161191
1  0.147116
2  0.161191
3  0.147116
4  0.161191
```

In [46]:

```
teacher_prefix_accepted=pd.DataFrame(p3)
print(teacher_prefix_accepted.shape)
teacher_prefix_rejected=pd.DataFrame(p4)
print(teacher_prefix_rejected.shape)
```

```
(10500, 1)
(10500, 1)
```

In [47]:

```
teacher_prefix_accepted=pd.DataFrame(p5)
print(teacher_prefix_accepted.shape)
teacher_prefix_rejected=pd.DataFrame(p6)
print(teacher_prefix_rejected.shape)
```

```
(15000, 1)
(15000, 1)
```

In [48]:

```
x1=teacher_prefix_accepted[0].fillna(value=0.5)
x2=teacher_prefix_rejected[0].fillna(value=0.5)
X_train['teacher_prefix_acc_prob']=x1
X_train['teacher_prefix_rej_prob']=x2
X_train['teacher_prefix_acc_prob'].fillna(0.5,inplace=True)
X_train['teacher_prefix_rej_prob'].fillna(0.5,inplace=True)
X_train[0:5]
```

Out[48]:

	Unnamed: 0	teacher_prefix	school_state	project_submitted_datetime	project
36740	32556	Ms.	LA	2017-02-03 13:25:06	Ms. Britta Busy Bee
5495	21790	Mrs.	PA	2016-08-16 12:22:01	Firefly Reading Writing Project
4877	178594	Ms.	UT	2016-12-15 15:05:19	Meting Science Art into a Treasure Keepsak
36415	141495	Mrs.	CA	2016-09-01 00:24:34	Chromeb for a Cor
32628	137177	Ms.	MS	2016-08-24 12:29:25	\"BOOST Learning Through Technolo

In [49]:

```
X_cv['teacher_prefix_acc_prob']=teacher_prefix_accepted
X_cv['teacher_prefix_rej_prob']=teacher_prefix_rejected
X_cv['teacher_prefix_acc_prob'].fillna(0.5,inplace=True)
X_cv['teacher_prefix_rej_prob'].fillna(0.5,inplace=True)
X_cv.shape
```

Out[49]:

```
(10500, 16)
```

In [50]:

```
X_test['teacher_prefix_acc_prob']=teacher_prefix_accepted
X_test['teacher_prefix_rej_prob']=teacher_prefix_rejected
X_test['teacher_prefix_acc_prob'].fillna(0.5,inplace=True)
X_test['teacher_prefix_rej_prob'].fillna(0.5,inplace=True)
X_test.shape
```

Out[50]:

```
(15000, 16)
```

In [51]:

```
#response encoding for school state
state_accepted={}
state_rejected={}
for x in X_train['school_state'].unique():
    print('*'*50)
    print(x)
    states_total=X_train['school_state'][X_train['school_state']==x].count()
    print('Total projects ={}'.format(states_total))
    for y in X_train['project_is_approved'].unique():
        n_state=X_train['project_is_approved'][(X_train['school_state']==x)&(X_train['p
roject_is_approved']==y)].count()
        if y:
            print('Approved projects ={}'.format(n_state))
            print('Approved probability={}'.format(n_state/states_total))
            state_accepted.update({x:n_state/states_total})
        else:
            print('Rejected projects ={}'.format(n_state))
            print('Rejected probability={}'.format(n_state/states_total))
            state_rejected.update({x:n_state/states_total})
```

```
**************************************************
LA
Total projects =546
Rejected projects =96
Rejected probability=0.17582417582417584
Approved projects =450
Approved probability=0.8241758241758241
**************************************************
PA
Total projects =709
Rejected projects =110
Rejected probability=0.15514809590973203
Approved projects =599
Approved probability=0.844851904090268
**************************************************
UT
Total projects =388
Rejected projects =65
Rejected probability=0.16752577319587628
Approved projects =323
Approved probability=0.8324742268041238
**************************************************
CA
Total projects =3429
Rejected projects =496
Rejected probability=0.1446485855934675
Approved projects =2933
Approved probability=0.8553514144065325
**************************************************
MS
Total projects =302
Rejected projects =49
Rejected probability=0.16225165562913907
Approved projects =253
Approved probability=0.8377483443708609
**************************************************
OK
Total projects =530
Rejected projects =80
Rejected probability=0.1509433962264151
Approved projects =450
Approved probability=0.8490566037735849
**************************************************
WA
Total projects =546
Rejected projects =62
Rejected probability=0.11355311355311355
Approved projects =484
Approved probability=0.8864468864468864
**************************************************
VA
Total projects =432
Rejected projects =70
Rejected probability=0.16203703703703703
Approved projects =362
Approved probability=0.8379629629629629
**************************************************
NY
Total projects =1620
Rejected projects =230
Rejected probability=0.1419753086419753
```

```
Approved projects =1390
Approved probability=0.8580246913580247
**************************************************
AZ
Total projects =487
Rejected projects =79
Rejected probability=0.162217659137577
Approved projects =408
Approved probability=0.837782340862423
**************************************************
MD
Total projects =306
Rejected projects =49
Rejected probability=0.16013071895424835
Approved projects =257
Approved probability=0.8398692810457516
**************************************************
NC
Total projects =1141
Rejected projects =171
Rejected probability=0.14986853637160386
Approved projects =970
Approved probability=0.8501314636283961
**************************************************
AL
Total projects =396
Rejected projects =53
Rejected probability=0.13383838383838384
Approved projects =343
Approved probability=0.8661616161616161
**************************************************
HI
Total projects =114
Rejected projects =13
Rejected probability=0.11403508771929824
Approved projects =101
Approved probability=0.8859649122807017
**************************************************
TX
Total projects =1633
Rejected projects =339
Rejected probability=0.20759338640538885
Approved projects =1294
Approved probability=0.7924066135946112
**************************************************
DC
Total projects =120
Rejected projects =22
Rejected probability=0.18333333333333332
Approved projects =98
Approved probability=0.8166666666666667
**************************************************
MI
Total projects =719
Rejected projects =112
Rejected probability=0.15577190542420027
Approved projects =607
Approved probability=0.8442280945757997
**************************************************
MA
Total projects =520
```

```
Rejected projects =79
Rejected probability=0.1519230769230769
Approved projects =441
Approved probability=0.8480769230769231
**************************************************
KY
Total projects =294
Rejected projects =45
Rejected probability=0.15306122448979592
Approved projects =249
Approved probability=0.8469387755102041
**************************************************
MN
Total projects =283
Rejected projects =41
Rejected probability=0.14487632508833923
Approved projects =242
Approved probability=0.8551236749116607
**************************************************
GA
Total projects =885
Rejected projects =122
Rejected probability=0.13785310734463277
Approved projects =763
Approved probability=0.8621468926553673
**************************************************
IN
Total projects =564
Rejected projects =89
Rejected probability=0.15780141843971632
Approved projects =475
Approved probability=0.8421985815602837
**************************************************
SC
Total projects =882
Rejected projects =120
Rejected probability=0.1360544217687075
Approved projects =762
Approved probability=0.8639455782312925
**************************************************
FL
Total projects =1383
Rejected projects =246
Rejected probability=0.17787418655097614
Approved projects =1137
Approved probability=0.8221258134490239
**************************************************
NV
Total projects =344
Rejected projects =46
Rejected probability=0.13372093023255813
Approved projects =298
Approved probability=0.8662790697674418
**************************************************
IL
Total projects =988
Rejected projects =153
Rejected probability=0.1548582995951417
Approved projects =835
Approved probability=0.8451417004048583
**************************************************
```

```
MO
Total projects =599
Rejected projects =82
Rejected probability=0.13689482470784642
Approved projects =517
Approved probability=0.8631051752921536
**************************************************
NM
Total projects =118
Rejected projects =19
Rejected probability=0.16101694915254236
Approved projects =99
Approved probability=0.8389830508474576
**************************************************
WI
Total projects =417
Rejected projects =69
Rejected probability=0.16546762589928057
Approved projects =348
Approved probability=0.8345323741007195
**************************************************
IA
Total projects =154
Rejected projects =18
Rejected probability=0.11688311688311688
Approved projects =136
Approved probability=0.8831168831168831
**************************************************
NJ
Total projects =492
Rejected projects =90
Rejected probability=0.18292682926829268
Approved projects =402
Approved probability=0.8170731707317073
**************************************************
TN
Total projects =377
Rejected projects =50
Rejected probability=0.13262599469496023
Approved projects =327
Approved probability=0.8673740053050398
**************************************************
CO
Total projects =259
Rejected projects =54
Rejected probability=0.2084942084942085
Approved projects =205
Approved probability=0.7915057915057915
**************************************************
CT
Total projects =392
Rejected projects =45
Rejected probability=0.11479591836734694
Approved projects =347
Approved probability=0.8852040816326531
**************************************************
ID
Total projects =146
Rejected projects =30
Rejected probability=0.2054794520547945
Approved projects =116
```

```
Approved probability=0.7945205479452054
**************************************************
OH
Total projects =584
Rejected projects =72
Rejected probability=0.1232876712328767
Approved projects =512
Approved probability=0.8767123287671232
**************************************************
NE
Total projects =67
Rejected projects =12
Rejected probability=0.1791044776119403
Approved projects =55
Approved probability=0.8208955223880597
**************************************************
NH
Total projects =68
Rejected projects =7
Rejected probability=0.10294117647058823
Approved projects =61
Approved probability=0.8970588235294118
**************************************************
OR
Total projects =289
Rejected projects =47
Rejected probability=0.16262975778546712
Approved projects =242
Approved probability=0.8373702422145328
**************************************************
AR
Total projects =243
Rejected projects =48
Rejected probability=0.19753086419753085
Approved projects =195
Approved probability=0.8024691358024691
**************************************************
DE
Total projects =67
Rejected projects =5
Rejected probability=0.07462686567164178
Approved projects =62
Approved probability=0.9253731343283582
**************************************************
KS
Total projects =125
Rejected projects =13
Rejected probability=0.104
Approved projects =112
Approved probability=0.896
**************************************************
SD
Total projects =63
Rejected projects =10
Rejected probability=0.15873015873015872
Approved projects =53
Approved probability=0.8412698412698413
**************************************************
RI
Total projects =69
Rejected projects =9
```

```
Rejected probability=0.13043478260869565
Approved projects =60
Approved probability=0.8695652173913043
**************************************************
AK
Total projects =81
Rejected projects =13
Rejected probability=0.16049382716049382
Approved projects =68
Approved probability=0.8395061728395061
**************************************************
ME
Total projects =94
Rejected projects =12
Rejected probability=0.1276595744680851
Approved projects =82
Approved probability=0.8723404255319149
**************************************************
WV
Total projects =113
Rejected projects =14
Rejected probability=0.12389380530973451
Approved projects =99
Approved probability=0.8761061946902655
**************************************************
MT
Total projects =51
Rejected projects =11
Rejected probability=0.21568627450980393
Approved projects =40
Approved probability=0.7843137254901961
**************************************************
WY
Total projects =23
Rejected projects =5
Rejected probability=0.21739130434782608
Approved projects =18
Approved probability=0.782608695652174
**************************************************
VT
Total projects =16
Rejected projects =4
Rejected probability=0.25
Approved projects =12
Approved probability=0.75
**************************************************
ND
Total projects =32
Rejected projects =4
Rejected probability=0.125
Approved projects =28
Approved probability=0.875
```

In [52]:

```
print(state_accepted)
```

```
{'LA': 0.8241758241758241, 'PA': 0.844851904090268, 'UT': 0.83247422680412
38, 'CA': 0.8553514144065325, 'MS': 0.8377483443708609, 'OK': 0.8490566037
735849, 'WA': 0.8864468864468864, 'VA': 0.8379629629629629, 'NY': 0.858024
6913580247, 'AZ': 0.837782340862423, 'MD': 0.8398692810457516, 'NC': 0.850
1314636283961, 'AL': 0.8661616161616161, 'HI': 0.8859649122807017, 'TX':
0.7924066135946112, 'DC': 0.8166666666666667, 'MI': 0.8442280945757997, 'M
A': 0.8480769230769231, 'KY': 0.8469387755102041, 'MN': 0.855123674911660
7, 'GA': 0.8621468926553673, 'IN': 0.8421985815602837, 'SC': 0.86394557823
12925, 'FL': 0.8221258134490239, 'NV': 0.8662790697674418, 'IL': 0.8451417
004048583, 'MO': 0.8631051752921536, 'NM': 0.8389830508474576, 'WI': 0.834
5323741007195, 'IA': 0.8831168831168831, 'NJ': 0.8170731707317073, 'TN':
0.8673740053050398, 'CO': 0.7915057915057915, 'CT': 0.8852040816326531, 'I
D': 0.7945205479452054, 'OH': 0.8767123287671232, 'NE': 0.820895522388059
7, 'NH': 0.8970588235294118, 'OR': 0.8373702422145328, 'AR': 0.80246913580
24691, 'DE': 0.9253731343283582, 'KS': 0.896, 'SD': 0.8412698412698413, 'R
I': 0.8695652173913043, 'AK': 0.8395061728395061, 'ME': 0.872340425531914
9, 'WV': 0.8761061946902655, 'MT': 0.7843137254901961, 'WY': 0.78260869565
2174, 'VT': 0.75, 'ND': 0.875}
```

In [53]:

```
s1=[]
s2=[]
for i in (X_train['school_state']):
    s1.append(state_accepted[i])
    s2.append(state_rejected[i])
print(len(s1))
print(len(s2))
```

```
24500
24500
```

In [54]:

```
s3=[]
s4=[]
for i in (X_cv['school_state']):
    s3.append(state_accepted[i])
    s4.append(state_rejected[i])
print(len(s3))
print(len(s4))
```

```
10500
10500
```

In [55]:

```
s5=[]
s6=[]
for i in (X_test['school_state']):
    s5.append(state_accepted[i])
    s6.append(state_rejected[i])
print(len(s5))
print(len(s6))
```

```
15000
15000
```

In [56]:

```
train_school_state_accepted=pd.DataFrame(s1)
train_school_state_rejected=pd.DataFrame(s2)
cv_school_state_accepted=pd.DataFrame(s3)
cv_school_state_rejected=pd.DataFrame(s4)
test_school_state_accepted=pd.DataFrame(s5)
test_school_state_rejected=pd.DataFrame(s6)


print(train_school_state_accepted.shape)
print(train_school_state_rejected.shape)
print(cv_school_state_accepted.shape)
print(cv_school_state_rejected.shape)
print(test_school_state_accepted.shape)
print(test_school_state_rejected.shape)
```

```
(24500, 1)
(24500, 1)
(10500, 1)
(10500, 1)
(15000, 1)
(15000, 1)
```

In [57]:

```
#state1=school_state_accepted[0].fillna(value=0.5)
#state2=school_state_rejected[0].fillna(value=0.5)
X_train['school_state_acc_prob']=train_school_state_accepted
X_train['school_state_rej_prob']=train_school_state_rejected
X_cv['school_state_acc_prob']=cv_school_state_accepted
X_cv['school_state_rej_prob']=cv_school_state_rejected
X_test['school_state_acc_prob']=test_school_state_accepted
X_test['school_state_rej_prob']=test_school_state_rejected
X_train['school_state_acc_prob'].fillna(0.5,inplace=True)
X_train['school_state_rej_prob'].fillna(0.5,inplace=True)
X_cv['school_state_acc_prob'].fillna(0.5,inplace=True)
X_cv['school_state_rej_prob'].fillna(0.5,inplace=True)
X_test['school_state_acc_prob'].fillna(0.5,inplace=True)
X_test['school_state_rej_prob'].fillna(0.5,inplace=True)
print(X_train.shape)
print(X_cv.shape)
print(X_test.shape)
```

```
(24500, 18)
(10500, 18)
(15000, 18)
```

In [58]:

```
#response coding for clean _categories
clean_cat_accepted={}
clean_cat_rejected={}
for x in X_train['clean_categories'].unique():
    print('*'*50)
    print(x)
    clean_cat_total=X_train['clean_categories'][X_train['clean_categories']==x].count()
    print('Total projects ={}'.format(clean_cat_total))
    for y in X_train['project_is_approved'].unique():
        n_clean_cat=X_train['project_is_approved'][(X_train['clean_categories']==x)&(X_
train['project_is_approved']==y)].count()
        if y:
            print('Approved projects ={}'.format(n_clean_cat))
            print('Approved probability={}'.format(n_clean_cat/clean_cat_total))
            clean_cat_accepted.update({x:n_clean_cat/clean_cat_total})
        else:
            print('Rejected projects ={}'.format(n_clean_cat))
            print('Rejected probability={}'.format(n_clean_cat/clean_cat_total))
            clean_cat_rejected.update({x:n_clean_cat/clean_cat_total})
```

```
**************************************************
Literacy_Language
Total projects =5406
Rejected projects =737
Rejected probability=0.13633000369959306
Approved projects =4669
Approved probability=0.863669996300407
**************************************************
Math_Science Music_Arts
Total projects =376
Rejected projects =61
Rejected probability=0.1622340425531915
Approved projects =315
Approved probability=0.8377659574468085
**************************************************
Math_Science Literacy_Language
Total projects =489
Rejected projects =60
Rejected probability=0.12269938650306748
Approved projects =429
Approved probability=0.8773006134969326
**************************************************
Literacy_Language Math_Science
Total projects =3264
Rejected projects =441
Rejected probability=0.13511029411764705
Approved projects =2823
Approved probability=0.8648897058823529
**************************************************
AppliedLearning
Total projects =841
Rejected projects =168
Rejected probability=0.19976218787158145
Approved projects =673
Approved probability=0.8002378121284186
**************************************************
Literacy_Language SpecialNeeds
Total projects =906
Rejected projects =133
Rejected probability=0.1467991169977925
Approved projects =773
Approved probability=0.8532008830022075
**************************************************
Literacy_Language AppliedLearning
Total projects =144
Rejected projects =21
Rejected probability=0.14583333333333334
Approved projects =123
Approved probability=0.8541666666666666
**************************************************
Health_Sports
Total projects =2330
Rejected projects =364
Rejected probability=0.15622317596566523
Approved projects =1966
Approved probability=0.8437768240343347
**************************************************
Literacy_Language History_Civics
Total projects =169
Rejected projects =26
Rejected probability=0.15384615384615385
```

```
Approved projects =143
Approved probability=0.8461538461538461
**************************************************
Literacy_Language Music_Arts
Total projects =400
Rejected projects =66
Rejected probability=0.165
Approved projects =334
Approved probability=0.835
**************************************************
History_Civics Literacy_Language
Total projects =326
Rejected projects =24
Rejected probability=0.0736196319018405
Approved projects =302
Approved probability=0.9263803680981595
**************************************************
Math_Science
Total projects =3776
Rejected projects =685
Rejected probability=0.18140889830508475
Approved projects =3091
Approved probability=0.8185911016949152
**************************************************
History_Civics
Total projects =390
Rejected projects =65
Rejected probability=0.16666666666666666
Approved projects =325
Approved probability=0.8333333333333334
**************************************************
Math_Science SpecialNeeds
Total projects =447
Rejected projects =73
Rejected probability=0.16331096196868009
Approved projects =374
Approved probability=0.8366890380313199
**************************************************
Music_Arts
Total projects =1171
Rejected projects =188
Rejected probability=0.1605465414175918
Approved projects =983
Approved probability=0.8394534585824082
**************************************************
Math_Science AppliedLearning
Total projects =256
Rejected projects =40
Rejected probability=0.15625
Approved projects =216
Approved probability=0.84375
**************************************************
Health_Sports SpecialNeeds
Total projects =293
Rejected projects =41
Rejected probability=0.13993174061433447
Approved projects =252
Approved probability=0.8600682593856656
**************************************************
AppliedLearning Music_Arts
Total projects =194
```

```
Rejected projects =35
Rejected probability=0.18041237113402062
Approved projects =159
Approved probability=0.8195876288659794
**************************************************
AppliedLearning Literacy_Language
Total projects =487
Rejected projects =80
Rejected probability=0.16427104722792607
Approved projects =407
Approved probability=0.8357289527720739
**************************************************
Health_Sports Literacy_Language
Total projects =157
Rejected projects =28
Rejected probability=0.17834394904458598
Approved projects =129
Approved probability=0.821656050955414
**************************************************
AppliedLearning SpecialNeeds
Total projects =327
Rejected projects =55
Rejected probability=0.16819571865443425
Approved projects =272
Approved probability=0.8318042813455657
**************************************************
Health_Sports Math_Science
Total projects =59
Rejected projects =12
Rejected probability=0.2033898305084746
Approved projects =47
Approved probability=0.7966101694915254
**************************************************
Warmth Care_Hunger
Total projects =281
Rejected projects =17
Rejected probability=0.060498220640569395
Approved projects =264
Approved probability=0.9395017793594306
**************************************************
AppliedLearning Math_Science
Total projects =220
Rejected projects =40
Rejected probability=0.18181818181818182
Approved projects =180
Approved probability=0.8181818181818182
**************************************************
SpecialNeeds
Total projects =952
Rejected projects =174
Rejected probability=0.18277310924369747
Approved projects =778
Approved probability=0.8172268907563025
**************************************************
Health_Sports Music_Arts
Total projects =33
Rejected projects =7
Rejected probability=0.21212121212121213
Approved projects =26
Approved probability=0.7878787878787878
**************************************************
```

```
History_Civics Math_Science
Total projects =86
Rejected projects =11
Rejected probability=0.12790697674418605
Approved projects =75
Approved probability=0.872093023255814
**************************************************
Math_Science Health_Sports
Total projects =78
Rejected projects =16
Rejected probability=0.20512820512820512
Approved projects =62
Approved probability=0.7948717948717948
**************************************************
Health_Sports Warmth Care_Hunger
Total projects =7
Rejected projects =1
Rejected probability=0.14285714285714285
Approved projects =6
Approved probability=0.8571428571428571
**************************************************
SpecialNeeds Music_Arts
Total projects =72
Rejected projects =12
Rejected probability=0.16666666666666666
Approved projects =60
Approved probability=0.8333333333333334
**************************************************
Math_Science History_Civics
Total projects =137
Rejected projects =26
Rejected probability=0.1897810218978102
Approved projects =111
Approved probability=0.8102189781021898
**************************************************
History_Civics SpecialNeeds
Total projects =45
Rejected projects =8
Rejected probability=0.17777777777777778
Approved projects =37
Approved probability=0.8222222222222222
**************************************************
Literacy_Language Health_Sports
Total projects =11
Rejected projects =2
Rejected probability=0.18181818181818182
Approved projects =9
Approved probability=0.8181818181818182
**************************************************
AppliedLearning Health_Sports
Total projects =136
Rejected projects =24
Rejected probability=0.17647058823529413
Approved projects =112
Approved probability=0.8235294117647058
**************************************************
History_Civics Music_Arts
Total projects =65
Rejected projects =12
Rejected probability=0.18461538461538463
Approved projects =53
```

```
Approved probability=0.8153846153846154
**************************************************
AppliedLearning History_Civics
Total projects =37
Rejected projects =8
Rejected probability=0.21621621621621623
Approved projects =29
Approved probability=0.7837837837837838
**************************************************
History_Civics Health_Sports
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
Health_Sports AppliedLearning
Total projects =45
Rejected projects =8
Rejected probability=0.17777777777777778
Approved projects =37
Approved probability=0.8222222222222222
**************************************************
Music_Arts SpecialNeeds
Total projects =34
Rejected projects =2
Rejected probability=0.058823529411764705
Approved projects =32
Approved probability=0.9411764705882353
**************************************************
History_Civics AppliedLearning
Total projects =12
Rejected projects =1
Rejected probability=0.08333333333333333
Approved projects =11
Approved probability=0.9166666666666666
**************************************************
Health_Sports History_Civics
Total projects =9
Rejected projects =1
Rejected probability=0.1111111111111111
Approved projects =8
Approved probability=0.8888888888888888
**************************************************
SpecialNeeds Warmth Care_Hunger
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
Music_Arts History_Civics
Total projects =3
Rejected projects =2
Rejected probability=0.6666666666666666
Approved projects =1
Approved probability=0.3333333333333333
**************************************************
Math_Science Warmth Care_Hunger
Total projects =3
Rejected projects =0
```

```
Rejected probability=0.0
Approved projects =3
Approved probability=1.0
**************************************************
Music_Arts Health_Sports
Total projects =3
Rejected projects =0
Rejected probability=0.0
Approved projects =3
Approved probability=1.0
**************************************************
Music_Arts AppliedLearning
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
AppliedLearning Warmth Care_Hunger
Total projects =6
Rejected projects =2
Rejected probability=0.3333333333333333
Approved projects =4
Approved probability=0.6666666666666666
**************************************************
SpecialNeeds Health_Sports
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
```

In [59]:

```
clean_cat_accepted.update({'Music_Arts Warmth Care_Hunger':0.5})
clean_cat_rejected.update({'Music_Arts Warmth Care_Hunger':0.5})
```

In [60]:

```
cat1=[]
cat2=[]
cat3=[]
cat4=[]
cat5=[]
cat6=[]
for i in (X_train['clean_categories']):
    if i in clean_cat_accepted:
        cat1.append(clean_cat_accepted[i])
        cat2.append(clean_cat_rejected[i])
    else:
        cat1.append(0.5)
        cat2.append(0.5)
for i in (X_cv['clean_categories']):
    if i in clean_cat_accepted:
        cat3.append(clean_cat_accepted[i])
        cat4.append(clean_cat_rejected[i])
    else:
        cat3.append(0.5)
        cat4.append(0.5)
for i in (X_test['clean_categories']):
    if i in clean_cat_accepted:
        cat5.append(clean_cat_accepted[i])
        cat6.append(clean_cat_rejected[i])
    else:
        cat5.append(0.5)
        cat6.append(0.5)


print(len(cat1))
print(len(cat2))
print(len(cat3))
print(len(cat4))
print(len(cat5))
print(len(cat6))
```

```
24500
24500
10500
10500
15000
15000
```

In [61]:

```
train_clean_category_accepted=pd.DataFrame(cat1)
train_clean_category_rejected=pd.DataFrame(cat2)
cv_clean_category_accepted=pd.DataFrame(cat3)
cv_clean_category_rejected=pd.DataFrame(cat4)
test_clean_category_accepted=pd.DataFrame(cat5)
test_clean_category_rejected=pd.DataFrame(cat6)
X_train['clean_cat_acc_prob']=train_clean_category_accepted
X_train['clean_cat_rej_prob']=train_clean_category_rejected
X_cv['clean_cat_acc_prob']=cv_clean_category_accepted
X_cv['clean_cat_rej_prob']=cv_clean_category_rejected
X_test['clean_cat_acc_prob']=test_clean_category_accepted
X_test['clean_cat_rej_prob']=test_clean_category_rejected
X_train['clean_cat_acc_prob'].fillna(0.5,inplace=True)
X_train['clean_cat_rej_prob'].fillna(0.5,inplace=True)
X_cv['clean_cat_acc_prob'].fillna(0.5,inplace=True)
X_cv['clean_cat_rej_prob'].fillna(0.5,inplace=True)
X_test['clean_cat_acc_prob'].fillna(0.5,inplace=True)
X_test['clean_cat_rej_prob'].fillna(0.5,inplace=True)
print(X_train.shape)
print(X_cv.shape)
print(X_test.shape)
```

```
(24500, 20)
(10500, 20)
(15000, 20)
```

In [62]:

```
#response coding for clean sub categories
clean_subcat_accepted={}
clean_subcat_rejected={}
for x in X_train['clean_subcategories'].unique():
    print('*'*50)
    print(x)
    clean_subcat_total=X_train['clean_subcategories'][X_train['clean_subcategories']==x
].count()
    print('Total projects ={}'.format(clean_subcat_total))
    for y in X_train['project_is_approved'].unique():
        n_clean_subcat=X_train['project_is_approved'][(X_train['clean_subcategories']==
x)&(X_train['project_is_approved']==y)].count()
        if y:
            print('Approved projects ={}'.format(n_clean_subcat))
            print('Approved probability={}'.format(n_clean_subcat/clean_subcat_total))
            clean_subcat_accepted.update({x:n_clean_subcat/clean_subcat_total})
        else:
            print('Rejected projects ={}'.format(n_clean_subcat))
            print('Rejected probability={}'.format(n_clean_subcat/clean_subcat_total))
            clean_subcat_rejected.update({x:n_clean_subcat/clean_subcat_total})
```

```
**************************************************
Literature_Writing
Total projects =998
Rejected projects =159
Rejected probability=0.1593186372745491
Approved projects =839
Approved probability=0.8406813627254509
**************************************************
Literacy Literature_Writing
Total projects =1283
Rejected projects =174
Rejected probability=0.13561964146531566
Approved projects =1109
Approved probability=0.8643803585346843
**************************************************
AppliedSciences VisualArts
Total projects =151
Rejected projects =25
Rejected probability=0.16556291390728478
Approved projects =126
Approved probability=0.8344370860927153
**************************************************
AppliedSciences Literature_Writing
Total projects =81
Rejected projects =5
Rejected probability=0.06172839506172839
Approved projects =76
Approved probability=0.9382716049382716
**************************************************
Literacy Mathematics
Total projects =1819
Rejected projects =244
Rejected probability=0.13413963716327654
Approved projects =1575
Approved probability=0.8658603628367235
**************************************************
College_CareerPrep
Total projects =84
Rejected projects =16
Rejected probability=0.19047619047619047
Approved projects =68
Approved probability=0.8095238095238095
**************************************************
Literature_Writing SpecialNeeds
Total projects =299
Rejected projects =59
Rejected probability=0.19732441471571907
Approved projects =240
Approved probability=0.802675585284281
**************************************************
Literacy Other
Total projects =48
Rejected projects =6
Rejected probability=0.125
Approved projects =42
Approved probability=0.875
**************************************************
ESL Literacy
Total projects =500
Rejected projects =70
Rejected probability=0.14
```

```
Approved projects =430
Approved probability=0.86
**************************************************
Literature_Writing Mathematics
Total projects =1362
Rejected projects =184
Rejected probability=0.13509544787077826
Approved projects =1178
Approved probability=0.8649045521292217
**************************************************
Gym_Fitness Health_Wellness
Total projects =502
Rejected projects =61
Rejected probability=0.12151394422310757
Approved projects =441
Approved probability=0.8784860557768924
**************************************************
ESL
Total projects =91
Rejected projects =11
Rejected probability=0.12087912087912088
Approved projects =80
Approved probability=0.8791208791208791
**************************************************
Literacy SocialSciences
Total projects =79
Rejected projects =13
Rejected probability=0.16455696202531644
Approved projects =66
Approved probability=0.8354430379746836
**************************************************
Literacy VisualArts
Total projects =123
Rejected projects =17
Rejected probability=0.13821138211382114
Approved projects =106
Approved probability=0.8617886178861789
**************************************************
History_Geography Literacy
Total projects =122
Rejected projects =4
Rejected probability=0.03278688524590164
Approved projects =118
Approved probability=0.9672131147540983
**************************************************
Mathematics
Total projects =1220
Rejected projects =216
Rejected probability=0.17704918032786884
Approved projects =1004
Approved probability=0.8229508196721311
**************************************************
History_Geography
Total projects =108
Rejected projects =25
Rejected probability=0.23148148148148148
Approved projects =83
Approved probability=0.7685185185185185
**************************************************
Mathematics SpecialNeeds
Total projects =286
```

```
Rejected projects =51
Rejected probability=0.17832167832167833
Approved projects =235
Approved probability=0.8216783216783217
**************************************************
VisualArts
Total projects =477
Rejected projects =95
Rejected probability=0.19916142557651992
Approved projects =382
Approved probability=0.80083857442348
**************************************************
Literacy SpecialNeeds
Total projects =558
Rejected projects =67
Rejected probability=0.12007168458781362
Approved projects =491
Approved probability=0.8799283154121864
**************************************************
Mathematics Other
Total projects =26
Rejected projects =7
Rejected probability=0.2692307692307692
Approved projects =19
Approved probability=0.7307692307692307
**************************************************
CharacterEducation CommunityService
Total projects =26
Rejected projects =6
Rejected probability=0.23076923076923078
Approved projects =20
Approved probability=0.7692307692307693
**************************************************
Health_LifeScience Mathematics
Total projects =124
Rejected projects =28
Rejected probability=0.22580645161290322
Approved projects =96
Approved probability=0.7741935483870968
**************************************************
Literacy PerformingArts
Total projects =29
Rejected projects =5
Rejected probability=0.1724137931034483
Approved projects =24
Approved probability=0.8275862068965517
**************************************************
Health_Wellness
Total projects =848
Rejected projects =124
Rejected probability=0.14622641509433962
Approved projects =724
Approved probability=0.8537735849056604
**************************************************
Music
Total projects =345
Rejected projects =42
Rejected probability=0.12173913043478261
Approved projects =303
Approved probability=0.8782608695652174
**************************************************
```

```
Literacy
Total projects =2195
Rejected projects =263
Rejected probability=0.11981776765375854
Approved projects =1932
Approved probability=0.8801822323462415
**************************************************
AppliedSciences Mathematics
Total projects =741
Rejected projects =131
Rejected probability=0.1767881241565452
Approved projects =610
Approved probability=0.8232118758434548
**************************************************
AppliedSciences SpecialNeeds
Total projects =84
Rejected projects =12
Rejected probability=0.14285714285714285
Approved projects =72
Approved probability=0.8571428571428571
**************************************************
Economics FinancialLiteracy
Total projects =21
Rejected projects =6
Rejected probability=0.2857142857142857
Approved projects =15
Approved probability=0.7142857142857143
**************************************************
Literature_Writing SocialSciences
Total projects =72
Rejected projects =10
Rejected probability=0.1388888888888889
Approved projects =62
Approved probability=0.8611111111111112
**************************************************
Health_Wellness SpecialNeeds
Total projects =259
Rejected projects =37
Rejected probability=0.14285714285714285
Approved projects =222
Approved probability=0.8571428571428571
**************************************************
CharacterEducation VisualArts
Total projects =23
Rejected projects =3
Rejected probability=0.13043478260869565
Approved projects =20
Approved probability=0.8695652173913043
**************************************************
ESL Mathematics
Total projects =62
Rejected projects =8
Rejected probability=0.12903225806451613
Approved projects =54
Approved probability=0.8709677419354839
**************************************************
ESL EnvironmentalScience
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
```

```
Approved probability=0.8333333333333334
**************************************************
AppliedSciences Literacy
Total projects =138
Rejected projects =20
Rejected probability=0.14492753623188406
Approved projects =118
Approved probability=0.855072463768116
**************************************************
AppliedSciences
Total projects =526
Rejected projects =103
Rejected probability=0.1958174904942966
Approved projects =423
Approved probability=0.8041825095057035
**************************************************
EnvironmentalScience
Total projects =255
Rejected projects =47
Rejected probability=0.1843137254901961
Approved projects =208
Approved probability=0.8156862745098039
**************************************************
EarlyDevelopment Literacy
Total projects =165
Rejected projects =25
Rejected probability=0.15151515151515152
Approved projects =140
Approved probability=0.8484848484848485
**************************************************
Health_Wellness Literature_Writing
Total projects =60
Rejected projects =8
Rejected probability=0.13333333333333333
Approved projects =52
Approved probability=0.8666666666666667
**************************************************
Economics History_Geography
Total projects =7
Rejected projects =1
Rejected probability=0.14285714285714285
Approved projects =6
Approved probability=0.8571428571428571
**************************************************
ESL Literature_Writing
Total projects =184
Rejected projects =24
Rejected probability=0.13043478260869565
Approved projects =160
Approved probability=0.8695652173913043
**************************************************
EarlyDevelopment SpecialNeeds
Total projects =169
Rejected projects =24
Rejected probability=0.14201183431952663
Approved projects =145
Approved probability=0.8579881656804734
**************************************************
Mathematics Music
Total projects =13
Rejected projects =1
```

```
Rejected probability=0.07692307692307693
Approved projects =12
Approved probability=0.9230769230769231
**************************************************
Health_Wellness Mathematics
Total projects =52
Rejected projects =11
Rejected probability=0.21153846153846154
Approved projects =41
Approved probability=0.7884615384615384
**************************************************
Warmth Care_Hunger
Total projects =281
Rejected projects =17
Rejected probability=0.060498220640569395
Approved projects =264
Approved probability=0.9395017793594306
**************************************************
CharacterEducation Literacy
Total projects =64
Rejected projects =10
Rejected probability=0.15625
Approved projects =54
Approved probability=0.84375
**************************************************
EarlyDevelopment Literature_Writing
Total projects =52
Rejected projects =13
Rejected probability=0.25
Approved projects =39
Approved probability=0.75
**************************************************
EarlyDevelopment Mathematics
Total projects =71
Rejected projects =14
Rejected probability=0.19718309859154928
Approved projects =57
Approved probability=0.8028169014084507
**************************************************
Music PerformingArts
Total projects =224
Rejected projects =30
Rejected probability=0.13392857142857142
Approved projects =194
Approved probability=0.8660714285714286
**************************************************
EarlyDevelopment
Total projects =198
Rejected projects =38
Rejected probability=0.1919191919191919
Approved projects =160
Approved probability=0.8080808080808081
**************************************************
SpecialNeeds
Total projects =952
Rejected projects =174
Rejected probability=0.18277310924369747
Approved projects =778
Approved probability=0.8172268907563025
**************************************************
College_CareerPrep SpecialNeeds
```

```
Total projects =28
Rejected projects =6
Rejected probability=0.21428571428571427
Approved projects =22
Approved probability=0.7857142857142857
**************************************************
CharacterEducation Other
Total projects =24
Rejected projects =4
Rejected probability=0.16666666666666666
Approved projects =20
Approved probability=0.8333333333333334
**************************************************
Literature_Writing PerformingArts
Total projects =22
Rejected projects =4
Rejected probability=0.18181818181818182
Approved projects =18
Approved probability=0.8181818181818182
**************************************************
Gym_Fitness Music
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
Mathematics VisualArts
Total projects =121
Rejected projects =24
Rejected probability=0.19834710743801653
Approved projects =97
Approved probability=0.8016528925619835
**************************************************
NutritionEducation
Total projects =65
Rejected projects =10
Rejected probability=0.15384615384615385
Approved projects =55
Approved probability=0.8461538461538461
**************************************************
Gym_Fitness
Total projects =276
Rejected projects =45
Rejected probability=0.16304347826086957
Approved projects =231
Approved probability=0.8369565217391305
**************************************************
College_CareerPrep VisualArts
Total projects =34
Rejected projects =6
Rejected probability=0.17647058823529413
Approved projects =28
Approved probability=0.8235294117647058
**************************************************
EarlyDevelopment VisualArts
Total projects =32
Rejected projects =7
Rejected probability=0.21875
Approved projects =25
Approved probability=0.78125
```

```
**************************************************
Other SpecialNeeds
Total projects =80
Rejected projects =17
Rejected probability=0.2125
Approved projects =63
Approved probability=0.7875
**************************************************
FinancialLiteracy Mathematics
Total projects =33
Rejected projects =6
Rejected probability=0.18181818181818182
Approved projects =27
Approved probability=0.8181818181818182
**************************************************
EnvironmentalScience Literacy
Total projects =87
Rejected projects =12
Rejected probability=0.13793103448275862
Approved projects =75
Approved probability=0.8620689655172413
**************************************************
Gym_Fitness TeamSports
Total projects =135
Rejected projects =33
Rejected probability=0.24444444444444444
Approved projects =102
Approved probability=0.7555555555555555
**************************************************
Literacy ParentInvolvement
Total projects =37
Rejected projects =5
Rejected probability=0.13513513513513514
Approved projects =32
Approved probability=0.8648648648648649
**************************************************
Other
Total projects =204
Rejected projects =38
Rejected probability=0.18627450980392157
Approved projects =166
Approved probability=0.8137254901960784
**************************************************
Health_Wellness VisualArts
Total projects =7
Rejected projects =1
Rejected probability=0.14285714285714285
Approved projects =6
Approved probability=0.8571428571428571
**************************************************
Health_Wellness Literacy
Total projects =90
Rejected projects =16
Rejected probability=0.17777777777777778
Approved projects =74
Approved probability=0.8222222222222222
**************************************************
Health_LifeScience SpecialNeeds
Total projects =36
Rejected projects =6
Rejected probability=0.16666666666666666
```

```
Approved projects =30
Approved probability=0.8333333333333334
**************************************************
TeamSports
Total projects =241
Rejected projects =43
Rejected probability=0.17842323651452283
Approved projects =198
Approved probability=0.8215767634854771
**************************************************
Literature_Writing VisualArts
Total projects =158
Rejected projects =34
Rejected probability=0.21518987341772153
Approved projects =124
Approved probability=0.7848101265822784
**************************************************
Health_LifeScience Health_Wellness
Total projects =35
Rejected projects =8
Rejected probability=0.22857142857142856
Approved projects =27
Approved probability=0.7714285714285715
**************************************************
EarlyDevelopment Other
Total projects =32
Rejected projects =4
Rejected probability=0.125
Approved projects =28
Approved probability=0.875
**************************************************
SocialSciences
Total projects =43
Rejected projects =7
Rejected probability=0.16279069767441862
Approved projects =36
Approved probability=0.8372093023255814
**************************************************
Health_Wellness Warmth Care_Hunger
Total projects =7
Rejected projects =1
Rejected probability=0.14285714285714285
Approved projects =6
Approved probability=0.8571428571428571
**************************************************
AppliedSciences EnvironmentalScience
Total projects =217
Rejected projects =39
Rejected probability=0.17972350230414746
Approved projects =178
Approved probability=0.8202764976958525
**************************************************
CharacterEducation EarlyDevelopment
Total projects =39
Rejected projects =8
Rejected probability=0.20512820512820512
Approved projects =31
Approved probability=0.7948717948717948
**************************************************
EnvironmentalScience Literature_Writing
Total projects =58
```

```
Rejected projects =8
Rejected probability=0.13793103448275862
Approved projects =50
Approved probability=0.8620689655172413
**************************************************
FinancialLiteracy
Total projects =32
Rejected projects =5
Rejected probability=0.15625
Approved projects =27
Approved probability=0.84375
**************************************************
Economics Mathematics
Total projects =10
Rejected projects =0
Rejected probability=0.0
Approved projects =10
Approved probability=1.0
**************************************************
SpecialNeeds VisualArts
Total projects =72
Rejected projects =12
Rejected probability=0.16666666666666666
Approved projects =60
Approved probability=0.8333333333333334
**************************************************
Health_LifeScience
Total projects =183
Rejected projects =26
Rejected probability=0.14207650273224043
Approved projects =157
Approved probability=0.8579234972677595
**************************************************
EnvironmentalScience History_Geography
Total projects =35
Rejected projects =5
Rejected probability=0.14285714285714285
Approved projects =30
Approved probability=0.8571428571428571
**************************************************
College_CareerPrep Mathematics
Total projects =64
Rejected projects =11
Rejected probability=0.171875
Approved projects =53
Approved probability=0.828125
**************************************************
AppliedSciences Music
Total projects =13
Rejected projects =1
Rejected probability=0.07692307692307693
Approved projects =12
Approved probability=0.9230769230769231
**************************************************
Civics_Government SocialSciences
Total projects =28
Rejected projects =1
Rejected probability=0.03571428571428571
Approved projects =27
Approved probability=0.9642857142857143
**************************************************
```

```
ForeignLanguages
Total projects =72
Rejected projects =19
Rejected probability=0.2638888888888889
Approved projects =53
Approved probability=0.7361111111111112
**************************************************
EnvironmentalScience Mathematics
Total projects =176
Rejected projects =31
Rejected probability=0.17613636363636365
Approved projects =145
Approved probability=0.8238636363636364
**************************************************
FinancialLiteracy SpecialNeeds
Total projects =10
Rejected projects =2
Rejected probability=0.2
Approved projects =8
Approved probability=0.8
**************************************************
History_Geography Literature_Writing
Total projects =145
Rejected projects =15
Rejected probability=0.10344827586206896
Approved projects =130
Approved probability=0.896551724137931
**************************************************
ESL Health_Wellness
Total projects =5
Rejected projects =0
Rejected probability=0.0
Approved projects =5
Approved probability=1.0
**************************************************
Health_Wellness NutritionEducation
Total projects =162
Rejected projects =30
Rejected probability=0.18518518518518517
Approved projects =132
Approved probability=0.8148148148148148
**************************************************
AppliedSciences Extracurricular
Total projects =28
Rejected projects =1
Rejected probability=0.03571428571428571
Approved projects =27
Approved probability=0.9642857142857143
**************************************************
EnvironmentalScience Health_LifeScience
Total projects =192
Rejected projects =41
Rejected probability=0.21354166666666666
Approved projects =151
Approved probability=0.7864583333333334
**************************************************
ForeignLanguages Literacy
Total projects =53
Rejected projects =9
Rejected probability=0.16981132075471697
Approved projects =44
```

```
Approved probability=0.8301886792452831
**************************************************
Health_Wellness TeamSports
Total projects =84
Rejected projects =14
Rejected probability=0.16666666666666666
Approved projects =70
Approved probability=0.8333333333333334
**************************************************
EarlyDevelopment Health_Wellness
Total projects =66
Rejected projects =10
Rejected probability=0.15151515151515152
Approved projects =56
Approved probability=0.8484848484848485
**************************************************
History_Geography SocialSciences
Total projects =67
Rejected projects =10
Rejected probability=0.14925373134328357
Approved projects =57
Approved probability=0.8507462686567164
**************************************************
ForeignLanguages Literature_Writing
Total projects =21
Rejected projects =6
Rejected probability=0.2857142857142857
Approved projects =15
Approved probability=0.7142857142857143
**************************************************
ParentInvolvement TeamSports
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
AppliedSciences College_CareerPrep
Total projects =88
Rejected projects =11
Rejected probability=0.125
Approved projects =77
Approved probability=0.875
**************************************************
Literature_Writing Music
Total projects =17
Rejected projects =0
Rejected probability=0.0
Approved projects =17
Approved probability=1.0
**************************************************
NutritionEducation SpecialNeeds
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
PerformingArts
Total projects =96
Rejected projects =13
```

```
Rejected probability=0.13541666666666666
Approved projects =83
Approved probability=0.8645833333333334
**************************************************
College_CareerPrep Literature_Writing
Total projects =64
Rejected projects =8
Rejected probability=0.125
Approved projects =56
Approved probability=0.875
**************************************************
History_Geography VisualArts
Total projects =38
Rejected projects =7
Rejected probability=0.18421052631578946
Approved projects =31
Approved probability=0.8157894736842105
**************************************************
Civics_Government Literature_Writing
Total projects =18
Rejected projects =2
Rejected probability=0.1111111111111111
Approved projects =16
Approved probability=0.8888888888888888
**************************************************
AppliedSciences Health_Wellness
Total projects =9
Rejected projects =1
Rejected probability=0.1111111111111111
Approved projects =8
Approved probability=0.8888888888888888
**************************************************
Civics_Government FinancialLiteracy
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
CharacterEducation
Total projects =64
Rejected projects =16
Rejected probability=0.25
Approved projects =48
Approved probability=0.75
**************************************************
College_CareerPrep FinancialLiteracy
Total projects =4
Rejected projects =2
Rejected probability=0.5
Approved projects =2
Approved probability=0.5
**************************************************
College_CareerPrep Literacy
Total projects =62
Rejected projects =7
Rejected probability=0.11290322580645161
Approved projects =55
Approved probability=0.8870967741935484
**************************************************
Civics_Government History_Geography
```

```
Total projects =45
Rejected projects =7
Rejected probability=0.15555555555555556
Approved projects =38
Approved probability=0.8444444444444444
**************************************************
ForeignLanguages History_Geography
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
Civics_Government Literacy
Total projects =30
Rejected projects =1
Rejected probability=0.03333333333333333
Approved projects =29
Approved probability=0.9666666666666667
**************************************************
AppliedSciences Health_LifeScience
Total projects =142
Rejected projects =23
Rejected probability=0.1619718309859155
Approved projects =119
Approved probability=0.8380281690140845
**************************************************
ESL EarlyDevelopment
Total projects =15
Rejected projects =1
Rejected probability=0.06666666666666667
Approved projects =14
Approved probability=0.9333333333333333
**************************************************
Mathematics ParentInvolvement
Total projects =21
Rejected projects =4
Rejected probability=0.19047619047619047
Approved projects =17
Approved probability=0.8095238095238095
**************************************************
EnvironmentalScience SpecialNeeds
Total projects =41
Rejected projects =4
Rejected probability=0.0975609756097561
Approved projects =37
Approved probability=0.9024390243902439
**************************************************
CommunityService
Total projects =16
Rejected projects =3
Rejected probability=0.1875
Approved projects =13
Approved probability=0.8125
**************************************************
FinancialLiteracy Health_Wellness
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
```

```
**************************************************
AppliedSciences PerformingArts
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
EarlyDevelopment PerformingArts
Total projects =6
Rejected projects =0
Rejected probability=0.0
Approved projects =6
Approved probability=1.0
**************************************************
CharacterEducation ForeignLanguages
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
Gym_Fitness Other
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
History_Geography SpecialNeeds
Total projects =22
Rejected projects =4
Rejected probability=0.18181818181818182
Approved projects =18
Approved probability=0.8181818181818182
**************************************************
Extracurricular VisualArts
Total projects =20
Rejected projects =4
Rejected probability=0.2
Approved projects =16
Approved probability=0.8
**************************************************
Health_Wellness Other
Total projects =39
Rejected projects =7
Rejected probability=0.1794871794871795
Approved projects =32
Approved probability=0.8205128205128205
**************************************************
Health_LifeScience Literacy
Total projects =63
Rejected projects =7
Rejected probability=0.1111111111111111
Approved projects =56
Approved probability=0.8888888888888888
**************************************************
ParentInvolvement PerformingArts
Total projects =2
Rejected projects =0
Rejected probability=0.0
```

```
Approved projects =2
Approved probability=1.0
**************************************************
Health_LifeScience SocialSciences
Total projects =14
Rejected projects =4
Rejected probability=0.2857142857142857
Approved projects =10
Approved probability=0.7142857142857143
**************************************************
ESL VisualArts
Total projects =10
Rejected projects =2
Rejected probability=0.2
Approved projects =8
Approved probability=0.8
**************************************************
EarlyDevelopment ParentInvolvement
Total projects =10
Rejected projects =3
Rejected probability=0.3
Approved projects =7
Approved probability=0.7
**************************************************
Extracurricular Mathematics
Total projects =11
Rejected projects =1
Rejected probability=0.09090909090909091
Approved projects =10
Approved probability=0.9090909090909091
**************************************************
College_CareerPrep Other
Total projects =23
Rejected projects =6
Rejected probability=0.2608695652173913
Approved projects =17
Approved probability=0.7391304347826086
**************************************************
Music SpecialNeeds
Total projects =29
Rejected projects =1
Rejected probability=0.034482758620689655
Approved projects =28
Approved probability=0.9655172413793104
**************************************************
College_CareerPrep CommunityService
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
AppliedSciences ESL
Total projects =16
Rejected projects =1
Rejected probability=0.0625
Approved projects =15
Approved probability=0.9375
**************************************************
CharacterEducation Mathematics
Total projects =21
```

```
Rejected projects =4
Rejected probability=0.19047619047619047
Approved projects =17
Approved probability=0.8095238095238095
**************************************************
History_Geography Mathematics
Total projects =30
Rejected projects =3
Rejected probability=0.1
Approved projects =27
Approved probability=0.9
**************************************************
Extracurricular Other
Total projects =9
Rejected projects =2
Rejected probability=0.2222222222222222
Approved projects =7
Approved probability=0.7777777777777778
**************************************************
College_CareerPrep EnvironmentalScience
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
Gym_Fitness Mathematics
Total projects =7
Rejected projects =1
Rejected probability=0.14285714285714285
Approved projects =6
Approved probability=0.8571428571428571
**************************************************
Civics_Government
Total projects =16
Rejected projects =2
Rejected probability=0.125
Approved projects =14
Approved probability=0.875
**************************************************
Health_LifeScience VisualArts
Total projects =24
Rejected projects =2
Rejected probability=0.08333333333333333
Approved projects =22
Approved probability=0.9166666666666666
**************************************************
CommunityService Health_Wellness
Total projects =4
Rejected projects =1
Rejected probability=0.25
Approved projects =3
Approved probability=0.75
**************************************************
Other VisualArts
Total projects =18
Rejected projects =4
Rejected probability=0.2222222222222222
Approved projects =14
Approved probability=0.7777777777777778
**************************************************
```

```
EnvironmentalScience ParentInvolvement
Total projects =6
Rejected projects =0
Rejected probability=0.0
Approved projects =6
Approved probability=1.0
**************************************************
ParentInvolvement VisualArts
Total projects =10
Rejected projects =1
Rejected probability=0.1
Approved projects =9
Approved probability=0.9
**************************************************
CharacterEducation Literature_Writing
Total projects =40
Rejected projects =6
Rejected probability=0.15
Approved projects =34
Approved probability=0.85
**************************************************
Gym_Fitness SpecialNeeds
Total projects =30
Rejected projects =4
Rejected probability=0.13333333333333333
Approved projects =26
Approved probability=0.8666666666666667
**************************************************
ForeignLanguages Mathematics
Total projects =8
Rejected projects =1
Rejected probability=0.125
Approved projects =7
Approved probability=0.875
**************************************************
ESL SpecialNeeds
Total projects =45
Rejected projects =7
Rejected probability=0.15555555555555556
Approved projects =38
Approved probability=0.8444444444444444
**************************************************
AppliedSciences History_Geography
Total projects =23
Rejected projects =5
Rejected probability=0.21739130434782608
Approved projects =18
Approved probability=0.782608695652174
**************************************************
EnvironmentalScience VisualArts
Total projects =43
Rejected projects =7
Rejected probability=0.16279069767441862
Approved projects =36
Approved probability=0.8372093023255814
**************************************************
CommunityService VisualArts
Total projects =10
Rejected projects =2
Rejected probability=0.2
Approved projects =8
```

```
Approved probability=0.8
**************************************************
EarlyDevelopment Gym_Fitness
Total projects =7
Rejected projects =1
Rejected probability=0.14285714285714285
Approved projects =6
Approved probability=0.8571428571428571
**************************************************
Health_LifeScience History_Geography
Total projects =15
Rejected projects =2
Rejected probability=0.13333333333333333
Approved projects =13
Approved probability=0.8666666666666667
**************************************************
EnvironmentalScience Music
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
History_Geography PerformingArts
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
Extracurricular PerformingArts
Total projects =8
Rejected projects =0
Rejected probability=0.0
Approved projects =8
Approved probability=1.0
**************************************************
Extracurricular Health_Wellness
Total projects =7
Rejected projects =2
Rejected probability=0.2857142857142857
Approved projects =5
Approved probability=0.7142857142857143
**************************************************
CharacterEducation ParentInvolvement
Total projects =12
Rejected projects =2
Rejected probability=0.16666666666666666
Approved projects =10
Approved probability=0.8333333333333334
**************************************************
EnvironmentalScience Health_Wellness
Total projects =10
Rejected projects =2
Rejected probability=0.2
Approved projects =8
Approved probability=0.8
**************************************************
CharacterEducation Extracurricular
Total projects =12
Rejected projects =4
```

```
Rejected probability=0.3333333333333333
Approved projects =8
Approved probability=0.6666666666666666
**************************************************
AppliedSciences Civics_Government
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
Health_LifeScience Literature_Writing
Total projects =41
Rejected projects =5
Rejected probability=0.12195121951219512
Approved projects =36
Approved probability=0.8780487804878049
**************************************************
Economics
Total projects =10
Rejected projects =0
Rejected probability=0.0
Approved projects =10
Approved probability=1.0
**************************************************
AppliedSciences Other
Total projects =21
Rejected projects =3
Rejected probability=0.14285714285714285
Approved projects =18
Approved probability=0.8571428571428571
**************************************************
AppliedSciences CharacterEducation
Total projects =12
Rejected projects =2
Rejected probability=0.16666666666666666
Approved projects =10
Approved probability=0.8333333333333334
**************************************************
CharacterEducation College_CareerPrep
Total projects =21
Rejected projects =3
Rejected probability=0.14285714285714285
Approved projects =18
Approved probability=0.8571428571428571
**************************************************
Civics_Government College_CareerPrep
Total projects =3
Rejected projects =0
Rejected probability=0.0
Approved projects =3
Approved probability=1.0
**************************************************
AppliedSciences SocialSciences
Total projects =13
Rejected projects =2
Rejected probability=0.15384615384615385
Approved projects =11
Approved probability=0.8461538461538461
**************************************************
Literacy Music
```

```
Total projects =30
Rejected projects =2
Rejected probability=0.06666666666666667
Approved projects =28
Approved probability=0.9333333333333333
**************************************************
CharacterEducation Health_Wellness
Total projects =22
Rejected projects =2
Rejected probability=0.09090909090909091
Approved projects =20
Approved probability=0.9090909090909091
**************************************************
AppliedSciences EarlyDevelopment
Total projects =27
Rejected projects =4
Rejected probability=0.14814814814814814
Approved projects =23
Approved probability=0.8518518518518519
**************************************************
SocialSciences VisualArts
Total projects =11
Rejected projects =2
Rejected probability=0.18181818181818182
Approved projects =9
Approved probability=0.8181818181818182
**************************************************
ParentInvolvement
Total projects =7
Rejected projects =3
Rejected probability=0.42857142857142855
Approved projects =4
Approved probability=0.5714285714285714
**************************************************
Health_Wellness Music
Total projects =11
Rejected projects =1
Rejected probability=0.09090909090909091
Approved projects =10
Approved probability=0.9090909090909091
**************************************************
CommunityService EnvironmentalScience
Total projects =10
Rejected projects =2
Rejected probability=0.2
Approved projects =8
Approved probability=0.8
**************************************************
Extracurricular Music
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
CharacterEducation EnvironmentalScience
Total projects =6
Rejected projects =0
Rejected probability=0.0
Approved projects =6
Approved probability=1.0
```

```
**************************************************
Gym_Fitness Literacy
Total projects =4
Rejected projects =2
Rejected probability=0.5
Approved projects =2
Approved probability=0.5
**************************************************
Extracurricular
Total projects =22
Rejected projects =3
Rejected probability=0.13636363636363635
Approved projects =19
Approved probability=0.8636363636363636
**************************************************
ForeignLanguages Health_Wellness
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
Health_LifeScience NutritionEducation
Total projects =7
Rejected projects =2
Rejected probability=0.2857142857142857
Approved projects =5
Approved probability=0.7142857142857143
**************************************************
ForeignLanguages SpecialNeeds
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
CharacterEducation Civics_Government
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Literature_Writing ParentInvolvement
Total projects =11
Rejected projects =2
Rejected probability=0.18181818181818182
Approved projects =9
Approved probability=0.8181818181818182
**************************************************
CharacterEducation SpecialNeeds
Total projects =36
Rejected projects =7
Rejected probability=0.19444444444444445
Approved projects =29
Approved probability=0.8055555555555556
**************************************************
History_Geography Other
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
```

```
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
Literature_Writing Other
Total projects =29
Rejected projects =5
Rejected probability=0.1724137931034483
Approved projects =24
Approved probability=0.8275862068965517
**************************************************
CommunityService Health_LifeScience
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
Civics_Government Health_Wellness
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
History_Geography Music
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
EarlyDevelopment NutritionEducation
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
ESL Extracurricular
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
College_CareerPrep PerformingArts
Total projects =9
Rejected projects =4
Rejected probability=0.4444444444444444
Approved projects =5
Approved probability=0.5555555555555556
**************************************************
ForeignLanguages PerformingArts
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
College_CareerPrep ParentInvolvement
Total projects =10
```

```
Rejected projects =3
Rejected probability=0.3
Approved projects =7
Approved probability=0.7
**************************************************
ForeignLanguages SocialSciences
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
Civics_Government EnvironmentalScience
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
College_CareerPrep Health_LifeScience
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
Gym_Fitness NutritionEducation
Total projects =15
Rejected projects =3
Rejected probability=0.2
Approved projects =12
Approved probability=0.8
**************************************************
PerformingArts VisualArts
Total projects =16
Rejected projects =5
Rejected probability=0.3125
Approved projects =11
Approved probability=0.6875
**************************************************
ESL SocialSciences
Total projects =4
Rejected projects =2
Rejected probability=0.5
Approved projects =2
Approved probability=0.5
**************************************************
FinancialLiteracy Literature_Writing
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
Music VisualArts
Total projects =13
Rejected projects =3
Rejected probability=0.23076923076923078
Approved projects =10
Approved probability=0.7692307692307693
**************************************************
```

```
Extracurricular ParentInvolvement
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
Mathematics SocialSciences
Total projects =15
Rejected projects =4
Rejected probability=0.26666666666666666
Approved projects =11
Approved probability=0.7333333333333333
**************************************************
College_CareerPrep ForeignLanguages
Total projects =5
Rejected projects =2
Rejected probability=0.4
Approved projects =3
Approved probability=0.6
**************************************************
College_CareerPrep Extracurricular
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
Health_Wellness History_Geography
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
CommunityService FinancialLiteracy
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
EnvironmentalScience Extracurricular
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
CommunityService SpecialNeeds
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
Civics_Government Health_LifeScience
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
```

```
Approved probability=0.8
**************************************************
CommunityService Literature_Writing
Total projects =5
Rejected projects =2
Rejected probability=0.4
Approved projects =3
Approved probability=0.6
**************************************************
Health_Wellness SocialSciences
Total projects =4
Rejected projects =1
Rejected probability=0.25
Approved projects =3
Approved probability=0.75
**************************************************
ESL ForeignLanguages
Total projects =9
Rejected projects =2
Rejected probability=0.2222222222222222
Approved projects =7
Approved probability=0.7777777777777778
**************************************************
AppliedSciences ParentInvolvement
Total projects =14
Rejected projects =4
Rejected probability=0.2857142857142857
Approved projects =10
Approved probability=0.7142857142857143
**************************************************
Civics_Government Economics
Total projects =7
Rejected projects =0
Rejected probability=0.0
Approved projects =7
Approved probability=1.0
**************************************************
CharacterEducation PerformingArts
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
EnvironmentalScience NutritionEducation
Total projects =9
Rejected projects =3
Rejected probability=0.3333333333333333
Approved projects =6
Approved probability=0.6666666666666666
**************************************************
Other PerformingArts
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
CommunityService Literacy
Total projects =6
Rejected projects =3
```

```
Rejected probability=0.5
Approved projects =3
Approved probability=0.5
**************************************************
EnvironmentalScience SocialSciences
Total projects =14
Rejected projects =2
Rejected probability=0.14285714285714285
Approved projects =12
Approved probability=0.8571428571428571
**************************************************
EarlyDevelopment Health_LifeScience
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
Mathematics PerformingArts
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
AppliedSciences Gym_Fitness
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
ForeignLanguages VisualArts
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
Extracurricular NutritionEducation
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
SocialSciences SpecialNeeds
Total projects =8
Rejected projects =0
Rejected probability=0.0
Approved projects =8
Approved probability=1.0
**************************************************
College_CareerPrep History_Geography
Total projects =4
Rejected projects =1
Rejected probability=0.25
Approved projects =3
Approved probability=0.75
**************************************************
NutritionEducation Other
```

```
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
CommunityService ESL
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Civics_Government SpecialNeeds
Total projects =4
Rejected projects =1
Rejected probability=0.25
Approved projects =3
Approved probability=0.75
**************************************************
Extracurricular SpecialNeeds
Total projects =3
Rejected projects =0
Rejected probability=0.0
Approved projects =3
Approved probability=1.0
**************************************************
Extracurricular Literature_Writing
Total projects =6
Rejected projects =0
Rejected probability=0.0
Approved projects =6
Approved probability=1.0
**************************************************
EarlyDevelopment EnvironmentalScience
Total projects =7
Rejected projects =3
Rejected probability=0.42857142857142855
Approved projects =4
Approved probability=0.5714285714285714
**************************************************
Civics_Government CommunityService
Total projects =5
Rejected projects =0
Rejected probability=0.0
Approved projects =5
Approved probability=1.0
**************************************************
CommunityService ParentInvolvement
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
NutritionEducation SocialSciences
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
```

```
**************************************************
Literacy TeamSports
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
EarlyDevelopment Music
Total projects =5
Rejected projects =0
Rejected probability=0.0
Approved projects =5
Approved probability=1.0
**************************************************
Extracurricular TeamSports
Total projects =3
Rejected projects =0
Rejected probability=0.0
Approved projects =3
Approved probability=1.0
**************************************************
Health_LifeScience ParentInvolvement
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
CommunityService Economics
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
Other TeamSports
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
ForeignLanguages Health_LifeScience
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
College_CareerPrep EarlyDevelopment
Total projects =6
Rejected projects =0
Rejected probability=0.0
Approved projects =6
Approved probability=1.0
**************************************************
CommunityService Extracurricular
Total projects =4
Rejected projects =2
Rejected probability=0.5
```

```
Approved projects =2
Approved probability=0.5
**************************************************
Economics Literacy
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
CharacterEducation Health_LifeScience
Total projects =4
Rejected projects =1
Rejected probability=0.25
Approved projects =3
Approved probability=0.75
**************************************************
EarlyDevelopment ForeignLanguages
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
Civics_Government VisualArts
Total projects =6
Rejected projects =2
Rejected probability=0.3333333333333333
Approved projects =4
Approved probability=0.6666666666666666
**************************************************
Gym_Fitness PerformingArts
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
SpecialNeeds Warmth Care_Hunger
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
AppliedSciences CommunityService
Total projects =4
Rejected projects =1
Rejected probability=0.25
Approved projects =3
Approved probability=0.75
**************************************************
Music SocialSciences
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
CommunityService Mathematics
Total projects =5
```

```
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
CharacterEducation Gym_Fitness
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
Extracurricular Literacy
Total projects =10
Rejected projects =3
Rejected probability=0.3
Approved projects =7
Approved probability=0.7
**************************************************
CharacterEducation SocialSciences
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
College_CareerPrep NutritionEducation
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
FinancialLiteracy Literacy
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
FinancialLiteracy ForeignLanguages
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
College_CareerPrep Health_Wellness
Total projects =5
Rejected projects =2
Rejected probability=0.4
Approved projects =3
Approved probability=0.6
**************************************************
CommunityService Other
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
```

```
EarlyDevelopment History_Geography
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Gym_Fitness VisualArts
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
AppliedSciences Economics
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
College_CareerPrep SocialSciences
Total projects =7
Rejected projects =2
Rejected probability=0.2857142857142857
Approved projects =5
Approved probability=0.7142857142857143
**************************************************
AppliedSciences Warmth Care_Hunger
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
ESL Music
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Economics EnvironmentalScience
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
ParentInvolvement SpecialNeeds
Total projects =5
Rejected projects =0
Rejected probability=0.0
Approved projects =5
Approved probability=1.0
**************************************************
NutritionEducation TeamSports
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
```

```
Approved probability=0.5
**************************************************
Health_LifeScience Music
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
CharacterEducation History_Geography
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
PerformingArts SpecialNeeds
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
ESL Health_LifeScience
Total projects =6
Rejected projects =3
Rejected probability=0.5
Approved projects =3
Approved probability=0.5
**************************************************
PerformingArts TeamSports
Total projects =3
Rejected projects =0
Rejected probability=0.0
Approved projects =3
Approved probability=1.0
**************************************************
Mathematics Warmth Care_Hunger
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
ESL PerformingArts
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
Music ParentInvolvement
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
TeamSports VisualArts
Total projects =2
Rejected projects =1
```

```
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
Mathematics NutritionEducation
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
ForeignLanguages Other
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
CharacterEducation Economics
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
PerformingArts SocialSciences
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
ESL History_Geography
Total projects =7
Rejected projects =0
Rejected probability=0.0
Approved projects =7
Approved probability=1.0
**************************************************
AppliedSciences ForeignLanguages
Total projects =3
Rejected projects =2
Rejected probability=0.6666666666666666
Approved projects =1
Approved probability=0.3333333333333333
**************************************************
FinancialLiteracy History_Geography
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
EnvironmentalScience Other
Total projects =4
Rejected projects =2
Rejected probability=0.5
Approved projects =2
Approved probability=0.5
**************************************************
CharacterEducation TeamSports
```

```
Total projects =5
Rejected projects =2
Rejected probability=0.4
Approved projects =3
Approved probability=0.6
**************************************************
ForeignLanguages Music
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
CommunityService PerformingArts
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
EnvironmentalScience Warmth Care_Hunger
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Other Warmth Care_Hunger
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
SpecialNeeds TeamSports
Total projects =6
Rejected projects =1
Rejected probability=0.16666666666666666
Approved projects =5
Approved probability=0.8333333333333334
**************************************************
Economics SpecialNeeds
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
History_Geography ParentInvolvement
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
ESL ParentInvolvement
Total projects =2
Rejected projects =2
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
```

```
**************************************************
CharacterEducation Music
Total projects =5
Rejected projects =1
Rejected probability=0.2
Approved projects =4
Approved probability=0.8
**************************************************
NutritionEducation VisualArts
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
CharacterEducation Warmth Care_Hunger
Total projects =2
Rejected projects =2
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
CharacterEducation FinancialLiteracy
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Music Other
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
**************************************************
Extracurricular History_Geography
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Civics_Government Mathematics
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
College_CareerPrep ESL
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
AppliedSciences TeamSports
Total projects =2
Rejected projects =0
Rejected probability=0.0
```

```
Approved projects =2
Approved probability=1.0
**************************************************
EarlyDevelopment SocialSciences
Total projects =4
Rejected projects =0
Rejected probability=0.0
Approved projects =4
Approved probability=1.0
**************************************************
ParentInvolvement SocialSciences
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
Literature_Writing TeamSports
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
College_CareerPrep Economics
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
CommunityService History_Geography
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
EnvironmentalScience PerformingArts
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Economics VisualArts
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Gym_Fitness Literature_Writing
Total projects =3
Rejected projects =2
Rejected probability=0.6666666666666666
Approved projects =1
Approved probability=0.3333333333333333
**************************************************
EarlyDevelopment Extracurricular
Total projects =3
```

```
Rejected projects =2
Rejected probability=0.6666666666666666
Approved projects =1
Approved probability=0.3333333333333333
**************************************************
CharacterEducation ESL
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
EnvironmentalScience ForeignLanguages
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
Gym_Fitness History_Geography
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
College_CareerPrep Music
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
Health_LifeScience Other
Total projects =3
Rejected projects =1
Rejected probability=0.3333333333333333
Approved projects =2
Approved probability=0.6666666666666666
**************************************************
Literacy NutritionEducation
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
Other SocialSciences
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
FinancialLiteracy SocialSciences
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
```

```
College_CareerPrep TeamSports
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
ParentInvolvement Warmth Care_Hunger
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
Health_LifeScience TeamSports
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
EarlyDevelopment Warmth Care_Hunger
Total projects =2
Rejected projects =0
Rejected probability=0.0
Approved projects =2
Approved probability=1.0
**************************************************
Gym_Fitness ParentInvolvement
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
EarlyDevelopment TeamSports
Total projects =2
Rejected projects =1
Rejected probability=0.5
Approved projects =1
Approved probability=0.5
**************************************************
EnvironmentalScience FinancialLiteracy
Total projects =1
Rejected projects =0
Rejected probability=0.0
Approved projects =1
Approved probability=1.0
**************************************************
CommunityService SocialSciences
Total projects =1
Rejected projects =1
Rejected probability=1.0
Approved projects =0
Approved probability=0.0
```

In [63]:

```
subcat1=[]
subcat2=[]
subcat3=[]
subcat4=[]
subcat5=[]
subcat6=[]
for i in (X_train['clean_subcategories']):
    subcat1.append(clean_subcat_accepted[i])
    subcat2.append(clean_subcat_rejected[i])
for i in (X_cv['clean_subcategories']):
    if i in clean_subcat_accepted:
        subcat3.append(clean_subcat_accepted[i])
        subcat4.append(clean_subcat_rejected[i])
    else:
        subcat3.append(0.5)
        subcat4.append(0.5)
for i in (X_test['clean_subcategories']):
    if i in clean_subcat_accepted:
        subcat5.append(clean_subcat_accepted[i])
        subcat6.append(clean_subcat_rejected[i])
    else:
        subcat5.append(0.5)
        subcat6.append(0.5)
print(len(subcat1))
print(len(subcat2))
print(len(subcat3))
print(len(subcat4))
print(len(subcat5))
print(len(subcat6))
```

```
24500
24500
10500
10500
15000
15000
```

In [64]:

```
train_clean_subcategory_accepted=pd.DataFrame(subcat1)
train_clean_subcategory_rejected=pd.DataFrame(subcat2)
cv_clean_subcategory_accepted=pd.DataFrame(subcat3)
cv_clean_subcategory_rejected=pd.DataFrame(subcat4)
test_clean_subcategory_accepted=pd.DataFrame(subcat5)
test_clean_subcategory_rejected=pd.DataFrame(subcat6)
X_train['clean_subcat_acc_prob']=train_clean_subcategory_accepted
X_train['clean_subcat_rej_prob']=train_clean_subcategory_rejected
X_cv['clean_subcat_acc_prob']=cv_clean_subcategory_accepted
X_cv['clean_subcat_rej_prob']=cv_clean_subcategory_rejected
X_test['clean_subcat_acc_prob']=test_clean_subcategory_accepted
X_test['clean_subcat_rej_prob']=test_clean_subcategory_rejected
X_train['clean_subcat_acc_prob'].fillna(0.5,inplace=True)
X_train['clean_subcat_rej_prob'].fillna(0.5,inplace=True)
X_cv['clean_subcat_acc_prob'].fillna(0.5,inplace=True)
X_cv['clean_subcat_rej_prob'].fillna(0.5,inplace=True)
X_test['clean_subcat_acc_prob'].fillna(0.5,inplace=True)
X_test['clean_subcat_rej_prob'].fillna(0.5,inplace=True)

print(X_train.shape)
print(X_cv.shape)
print(X_train.shape)
```

```
(24500, 22)
(10500, 22)
(24500, 22)
```

In [65]:

```
#response coding for clean grades
clean_grades_accepted={}
clean_grades_rejected={}
for x in X_train['clean_grades'].unique():
    print('*'*50)
    print(x)
    clean_grades_total=X_train['clean_grades'][X_train['clean_grades']==x].count()
    print('Total projects ={}'.format(clean_grades_total))
    for y in X_train['project_is_approved'].unique():
        n_clean_grades=X_train['project_is_approved'][(X_train['clean_grades']==x)&(X_t
rain['project_is_approved']==y)].count()
        if y:
            print('Approved projects ={}'.format(n_clean_grades))
            print('Approved probability={}'.format(n_clean_grades/clean_grades_total))
            clean_grades_accepted.update({x:n_clean_grades/clean_grades_total})
        else:
            print('Rejected projects ={}'.format(n_clean_grades))
            print('Rejected probability={}'.format(n_clean_grades/clean_grades_total))
            clean_grades_rejected.update({x:n_clean_grades/clean_grades_total})
```

```
**************************************************
Grades_3_5
Total projects =8328
Rejected projects =1215
Rejected probability=0.14589337175792508
Approved projects =7113
Approved probability=0.854106628242075
**************************************************
Grades_6_8
Total projects =3759
Rejected projects =609
Rejected probability=0.16201117318435754
Approved projects =3150
Approved probability=0.8379888268156425
**************************************************
Grades_PreK_2
Total projects =9934
Rejected projects =1562
Rejected probability=0.1572377692772297
Approved projects =8372
Approved probability=0.8427622307227702
**************************************************
Grades_9_12
Total projects =2479
Rejected projects =394
Rejected probability=0.15893505445744252
Approved projects =2085
Approved probability=0.8410649455425575
```

In [66]:

```
grades1=[]
grades2=[]
grades3=[]
grades4=[]
grades5=[]
grades6=[]
for i in (X_train['clean_grades']):
    if i in clean_grades_accepted:
        grades1.append(clean_grades_accepted[i])
        grades2.append(clean_grades_rejected[i])
    else:
        grades1.append(0.5)
        grades2.append(0.5)
for i in (X_cv['clean_grades']):
    if i in clean_grades_accepted:
        grades3.append(clean_grades_accepted[i])
        grades4.append(clean_grades_rejected[i])
    else:
        grades3.append(0.5)
        grades4.append(0.5)
for i in (X_test['clean_grades']):
    if i in clean_grades_accepted:
        grades5.append(clean_grades_accepted[i])
        grades6.append(clean_grades_rejected[i])
    else:
        grades5.append(0.5)
        grades6.append(0.5)


print(len(grades1))
print(len(grades2))
print(len(grades3))
print(len(grades4))
print(len(grades5))
print(len(grades6))
```

```
24500
24500
10500
10500
15000
15000
```

In [67]:

```
train_clean_grades_accepted=pd.DataFrame(grades1)
train_clean_grades_rejected=pd.DataFrame(grades2)
cv_clean_grades_accepted=pd.DataFrame(grades3)
cv_clean_grades_rejected=pd.DataFrame(grades4)
test_clean_grades_accepted=pd.DataFrame(grades5)
test_clean_grades_rejected=pd.DataFrame(grades6)
X_train['clean_grades_acc_prob']=train_clean_grades_accepted
X_train['clean_grades_rej_prob']=train_clean_grades_rejected
X_cv['clean_grades_acc_prob']=cv_clean_grades_accepted
X_cv['clean_grades_rej_prob']=cv_clean_grades_rejected
X_test['clean_grades_acc_prob']=test_clean_grades_accepted
X_test['clean_grades_rej_prob']=test_clean_grades_rejected
X_train['clean_grades_acc_prob'].fillna(0.5,inplace=True)
X_train['clean_grades_rej_prob'].fillna(0.5,inplace=True)
X_cv['clean_grades_acc_prob'].fillna(0.5,inplace=True)
X_cv['clean_grades_rej_prob'].fillna(0.5,inplace=True)
X_test['clean_grades_acc_prob'].fillna(0.5,inplace=True)
X_test['clean_grades_rej_prob'].fillna(0.5,inplace=True)

print(X_train.shape)
print(X_cv.shape)
print(X_test.shape)
```

```
(24500, 24)
(10500, 24)
(15000, 24)
```

In [68]:

```
#encoding numerical categories---price
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))

print("after vectorization")
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)
```

```
after vectorization
(1, 24500) (24500,)
(1, 10500) (10500,)
(1, 15000) (15000,)
```

In [69]:

```
price_train_norm=X_train_price_norm.reshape(24500,1)
price_cv_norm=X_cv_price_norm.reshape(10500,1)
price_test_norm=X_test_price_norm.reshape(15000,1)
print(price_train_norm.shape)
print(price_cv_norm.shape)
print(price_test_norm.shape)
```

```
(24500, 1)
(10500, 1)
(15000, 1)
```

In [70]:

```
#encoding numerical category quantity
normalizer=Normalizer()
normalizer.fit(X_train['quantity'].values.reshape(1,-1))

X_train_quantity_norm=normalizer.transform(X_train['quantity'].values.reshape(1,-1))
X_cv_quantity_norm=normalizer.transform(X_cv['quantity'].values.reshape(1,-1))
X_test_quantity_norm=normalizer.transform(X_test['quantity'].values.reshape(1,-1))

print('after vectorization')
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)
```

```
after vectorization
(1, 24500) (24500,)
(1, 10500) (10500,)
(1, 15000) (15000,)
```

In [71]:

```
train_quantity_norm=X_train_quantity_norm.reshape(24500,1)
cv_quantity_norm=X_cv_quantity_norm.reshape(10500,1)
test_quantity_norm=X_test_quantity_norm.reshape(15000,1)
print(train_quantity_norm.shape)
print(cv_quantity_norm.shape)
print(test_quantity_norm.shape)
```

```
(24500, 1)
(10500, 1)
(15000, 1)
```

In [72]:

```
#encoding previous projects posted by teachers
normalizer=Normalizer()
normalizer.fit(X_train['teacher_number_of_previously_posted_projects'].values.reshape(1
,-1))

X_train_projects_norm=normalizer.transform(X_train['teacher_number_of_previously_posted
_projects'].values.reshape(1,-1))
X_cv_projects_norm=normalizer.transform(X_cv['teacher_number_of_previously_posted_proje
cts'].values.reshape(1,-1))
X_test_projects_norm=normalizer.transform(X_test['teacher_number_of_previously_posted_p
rojects'].values.reshape(1,-1))

print("after vectorization")
print(X_train_projects_norm.shape,y_train.shape)
print(X_cv_projects_norm.shape,y_cv.shape)
print(X_test_projects_norm.shape,y_test.shape)
```

```
after vectorization
(1, 24500) (24500,)
(1, 10500) (10500,)
(1, 15000) (15000,)
```

In [73]:

```
projects_train_norm=X_train_projects_norm.reshape(24500,1)
projects_cv_norm=X_cv_projects_norm.reshape(10500,1)
projects_test_norm=X_test_projects_norm.reshape(15000,1)
print(projects_train_norm.shape)
print(projects_cv_norm.shape)
print(projects_test_norm.shape)
```

```
(24500, 1)
(10500, 1)
(15000, 1)
```

2.3 Make Data Model Ready: encoding eassay, and project_title

In [80]:

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

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

In [120]:

```
essaybowvectorizer=CountVectorizer(min_df=10,ngram_range=(1,1))
essaybowvectorizer.fit(X_train['essay'].values)
X_train_essay_bow=essaybowvectorizer.transform(X_train['essay'].values)
#print(X_train_essay_bow.shape)
X_cv_essay_bow=essaybowvectorizer.transform(X_cv["essay"].values)
X_test_essay_bow=essaybowvectorizer.transform(X_test['essay'].values)

print('AFTER VECTORIZATION')
print('='*50)
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)
```

```
AFTER VECTORIZATION
==================================================
(24500, 9448) (24500,)
(10500, 9448) (10500,)
(15000, 9448) (15000,)
```

In [121]:

```
#encoding project title
titlebowvectorizer=CountVectorizer(min_df=10,ngram_range=(1,1))
titlebowvectorizer.fit(X_train['project_title'].values)
X_train_title_bow=titlebowvectorizer.transform(X_train['project_title'].values)
X_cv_title_bow=titlebowvectorizer.transform(X_cv['project_title'].values)
X_test_title_bow=titlebowvectorizer.transform(X_test['project_title'].values)
print("after vectorization")
print(X_train_title_bow.shape,y_train.shape)
print(X_cv_title_bow.shape,y_cv.shape)
print(X_test_title_bow.shape,y_test.shape)
```

```
after vectorization
(24500, 1323) (24500,)
(10500, 1323) (10500,)
(15000, 1323) (15000,)
```

In [80]:

```
X_train.columns
```

Out[80]:

```
Index(['Unnamed: 0', 'teacher_prefix', 'school_state',
       'project_submitted_datetime', 'project_title',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approve
d',
       'clean_categories', 'clean_grades', 'clean_subcategories', 'essay',
       'price', 'quantity', 'teacher_prefix_acc_prob',
       'teacher_prefix_rej_prob', 'school_state_acc_prob',
       'school_state_rej_prob', 'clean_cat_acc_prob', 'clean_cat_rej_pro
b',
       'clean_subcat_acc_prob', 'clean_subcat_rej_prob',
       'clean_grades_acc_prob', 'clean_grades_rej_prob'],
      dtype='object')
```

In [122]:

```
final_train=X_train.drop(['Unnamed: 0', 'teacher_prefix', 'school_state','project_submi
tted_datetime', 'project_title','project_resource_summary','teacher_number_of_previousl
y_posted_projects', 'project_is_approved','clean_categories', 'clean_grades', 'clean_su
bcategories', 'essay','price', 'quantity'],axis=1)
final_cv=X_cv.drop(['Unnamed: 0', 'teacher_prefix', 'school_state','project_submitted_d
atetime', 'project_title','project_resource_summary','teacher_number_of_previously_post
ed_projects', 'project_is_approved','clean_categories', 'clean_grades', 'clean_subcateg
ories', 'essay','price', 'quantity'],axis=1)
final_test=X_test.drop(['Unnamed: 0', 'teacher_prefix', 'school_state','project_submitt
ed_datetime', 'project_title','project_resource_summary','teacher_number_of_previously_
posted_projects', 'project_is_approved','clean_categories', 'clean_grades', 'clean_subc
ategories', 'essay','price', 'quantity'],axis=1)
print(final_train.shape)
print(final_cv.shape)
print(final_test.shape)
```

```
(24500, 10)
(10500, 10)
(15000, 10)
```

In [123]:

```
#final data matrix
final_train_bow=hstack((final_train,price_train_norm,train_quantity_norm,projects_train
_norm,X_train_essay_bow,X_train_title_bow)).tocsr()
final_cv_bow=hstack((final_cv,price_cv_norm,cv_quantity_norm,projects_cv_norm,X_cv_essa
y_bow,X_cv_title_bow)).tocsr()
final_test_bow=hstack((final_test,price_test_norm,test_quantity_norm,projects_test_norm
,X_test_essay_bow,X_test_title_bow)).tocsr()
print(final_train_bow.shape,y_train.shape)
print(final_cv_bow.shape,y_cv.shape)
print(final_test_bow.shape,y_test.shape)
```

```
(24500, 10784) (24500,)
(10500, 10784) (10500,)
(15000, 10784) (15000,)
```

2.4 Applying Random Forest

Apply Random Forest on different kind of featurization as mentioned in the instructions
For Every model that you work on make sure you do the step 2 and step 3 of instrucations

2.4.1 Applying Random Forests on BOW, SET 1

In [0]:

```
# Please write all the code with proper documentation
```

In [74]:

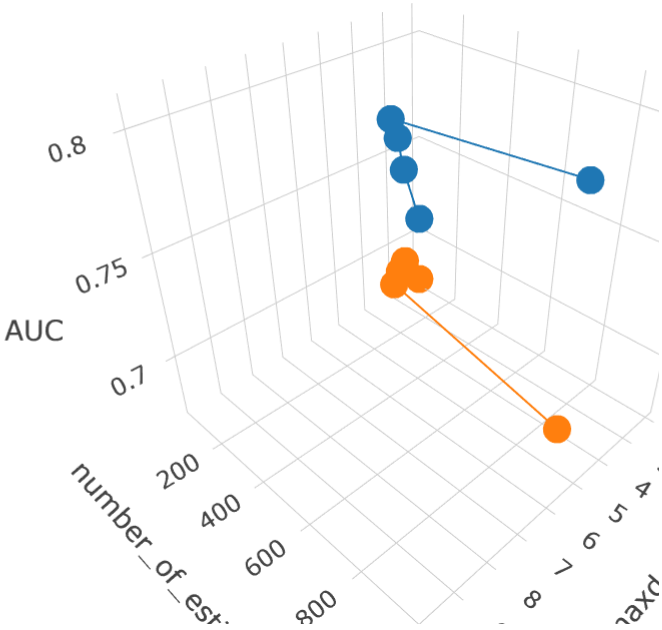
```
def enable_plotly_in_cell():
  import IPython
  from plotly.offline import init_notebook_mode
  display(IPython.core.display.HTML('''<script src="/static/components/requirejs/requir
e.js"></script>'''))
  init_notebook_mode(connected=False)
```

In [124]:

```
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import RandomForestClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8,10]
number_of_estimators=[50,100,200,300,1000]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        RandomForest=RandomForestClassifier(n_estimators=i,max_depth=j,class_weight='ba
lanced')
        calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
        calib_cv.fit(final_train_bow,y_train)

        y_tr_pred=calib_cv.predict_proba(final_train_bow)[:,1]
        y_cv_pred=calib_cv.predict_proba(final_cv_bow)[:,1]



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

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()

layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

```
  0%|
| 0/5 [00:00<?, ?it/s]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [00:06<00:41,  6.97s/it]
 29%|
| 2/7 [00:14<00:35,  7.06s/it]
 43%|
| 3/7 [00:21<00:28,  7.16s/it]
 57%|
| 4/7 [00:29<00:21,  7.31s/it]
 71%|
| 5/7 [00:37<00:15,  7.56s/it]
 86%|
   | 6/7 [00:46<00:08,  8.15s/it]
 20%|
| 1/5 [00:57<03:49, 57.33s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [00:13<01:23, 13.92s/it]
 29%|
| 2/7 [00:27<01:09, 13.89s/it]
 43%|
| 3/7 [00:41<00:55, 13.83s/it]
 57%|
| 4/7 [00:56<00:42, 14.11s/it]
 71%|
| 5/7 [01:11<00:28, 14.43s/it]
 86%|
   | 6/7 [01:27<00:15, 15.08s/it]
 40%|
| 2/5 [02:44<03:36, 72.25s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [00:25<02:31, 25.27s/it]
 29%|
| 2/7 [00:51<02:08, 25.67s/it]
 43%|
| 3/7 [01:19<01:44, 26.11s/it]
 57%|
| 4/7 [01:47<01:20, 26.69s/it]
 71%|
| 5/7 [02:16<00:55, 27.56s/it]
 86%|
   | 6/7 [02:49<00:29, 29.07s/it]
 60%|
| 3/5 [06:10<03:44, 112.41s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [00:40<04:02, 40.41s/it]
 29%|
| 2/7 [01:18<03:19, 39.80s/it]
 43%|
| 3/7 [01:57<02:38, 39.62s/it]
 57%|
| 4/7 [02:40<02:01, 40.42s/it]
 71%|
```

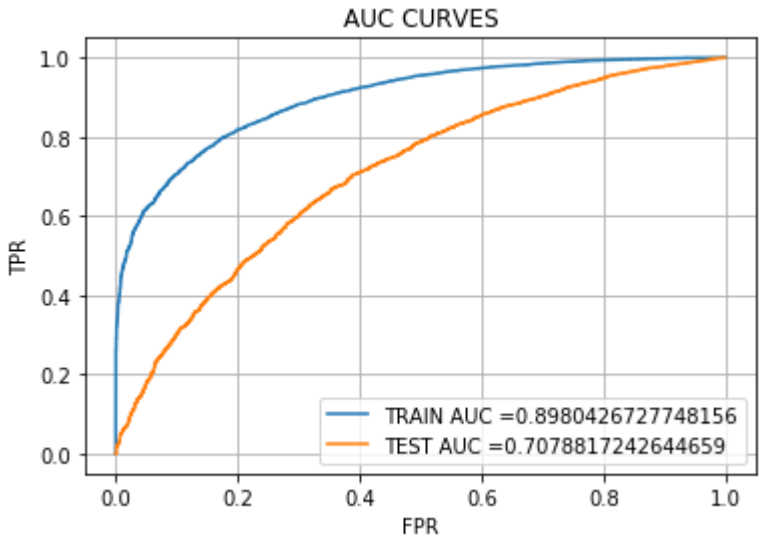
```
| 5/7 [03:24<01:23, 41.59s/it]
 86%|
| 6/7 [04:13<00:43, 43.86s/it]
 80%|
| 4/5 [11:19<02:51, 171.27s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [02:02<12:15, 122.54s/it]
 29%|
| 2/7 [04:09<10:18, 123.78s/it]
 43%|
| 3/7 [06:20<08:24, 126.18s/it]
 57%|
| 4/7 [08:38<06:29, 129.70s/it]
 71%|
| 5/7 [11:03<04:28, 134.06s/it]
 86%|
| 6/7 [13:45<02:22, 142.59s/it]
100%|
| 5/5 [28:09<00:00, 422.99s/it]
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators and we choose our best max depth and number of estimators to be 3 and 100 respectively.

In [83]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
RandomForest=RandomForestClassifier(max_depth=3,n_estimators=100,class_weight='balance
d')
calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
calib_cv.fit(final_train_bow,y_train)

y_train_pred=calib_cv.predict_proba(final_train_bow)[:,1]
y_test_pred=calib_cv.predict_proba(final_test_bow)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(y_train,y_tr_pred)
test_fpr,test_tpr,te_threshold=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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=3 and number of estimators=100 we got train auc =89.80% and test auc of 70.78%.

In [96]:

```
def find_best_threshold(threshold, fpr, tpr):
    t = threshold[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshold", np.rou
nd(t,3))
    return t

def predict_with_best_t(proba, threshold):
    predictions = []
    for i in proba:
        if i>=threshold:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
```

In [85]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(y_train,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(y_test,predict_with_best_t(y_test_pred,best_t)))
```

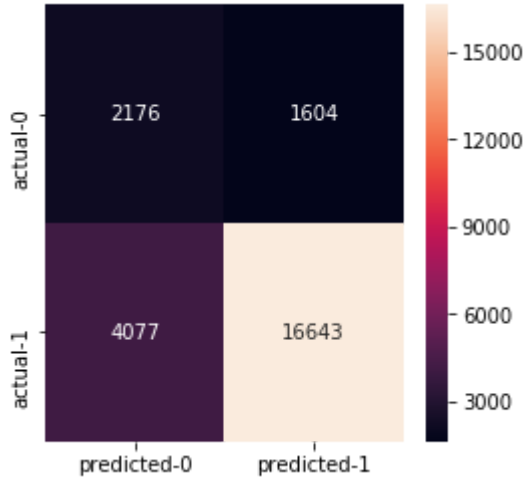
```
====================================================================================
==========================
the maximum value of tpr*(1-fpr) 0.6582921697207411 for threshold 0.799
TRAIN CONFUSION MATRIX
[[ 2176  1604]
 [ 4077 16643]]
test confusion matrix
[[1186 1128]
 [2835 9851]]
```

In [86]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[2176,1604],[4077,16643]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[86]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee1c2ce7f0>
```



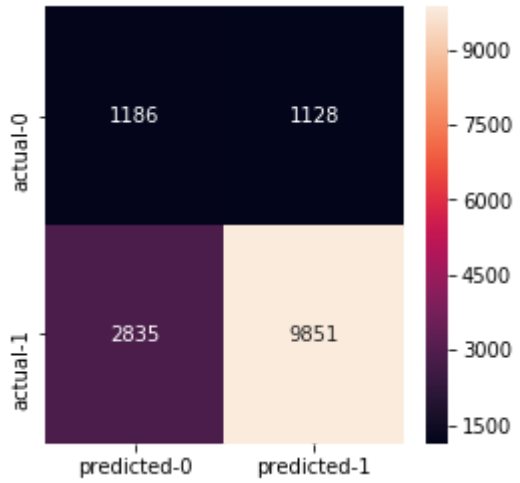
OBSERVATIONS: For train data we got decent values for tpr and tnr.

In [87]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[1186,1128],[2835,9851]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[87]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee0d103518>
```



OBSERVATIONS: For test data we got good tpr value

2.4.2 Applying Random Forests on TFIDF, SET 2

In [0]:

```
# Please write all the code with proper documentation
```

In [125]:

```
#tfidf encoding  of text
from sklearn.feature_extraction.text import TfidfVectorizer
essaytfidfvectorizer = TfidfVectorizer(min_df=10)
essaytfidfvectorizer.fit(X_train['essay'].values)
train_tfidf=essaytfidfvectorizer.transform(X_train['essay'].values)
cv_tfidf=essaytfidfvectorizer.transform(X_cv['essay'].values)
test_tfidf=essaytfidfvectorizer.transform(X_test['essay'].values)
print("Shape of matrix after one hot encodig ",train_tfidf.shape)
print("Shape of matrix after one hot encodig ",cv_tfidf.shape)
print("Shape of matrix after one hot encodig ",test_tfidf.shape)
```

```
Shape of matrix after one hot encodig  (24500, 9448)
Shape of matrix after one hot encodig  (10500, 9448)
Shape of matrix after one hot encodig  (15000, 9448)
```

In [126]:

```
#tfidf encoding of title
titletfidfvectorizer=TfidfVectorizer(min_df=10)
titletfidfvectorizer.fit(X_train['project_title'].values)
title_train_tfidf=titletfidfvectorizer.transform(X_train['project_title'].values)
title_cv_tfidf=titletfidfvectorizer.transform(X_cv['project_title'].values)
title_test_tfidf=titletfidfvectorizer.transform(X_test['project_title'].values)

print("Shape of matrix after one hot encodig ",title_train_tfidf.shape)
print("Shape of matrix after one hot encodig ",title_cv_tfidf.shape)
print("Shape of matrix after one hot encodig ",title_test_tfidf.shape)
```

```
Shape of matrix after one hot encodig  (24500, 1323)
Shape of matrix after one hot encodig  (10500, 1323)
Shape of matrix after one hot encodig  (15000, 1323)
```

In [127]:

```
#creating final data matrix
final_train_tfidf=hstack((final_train,price_train_norm,train_quantity_norm,projects_tra
in_norm,train_tfidf,title_train_tfidf)).tocsr()
final_cv_tfidf=hstack((final_cv,price_cv_norm,cv_quantity_norm,projects_cv_norm,cv_tfid
f,title_cv_tfidf)).tocsr()
final_test_tfidf=hstack((final_test,price_test_norm,test_quantity_norm,projects_test_no
rm,test_tfidf,title_test_tfidf)).tocsr()
print(final_train_tfidf.shape,y_train.shape)
print(final_cv_tfidf.shape,y_cv.shape)
print(final_test_tfidf.shape,y_test.shape)
```

```
(24500, 10784) (24500,)
(10500, 10784) (10500,)
(15000, 10784) (15000,)
```

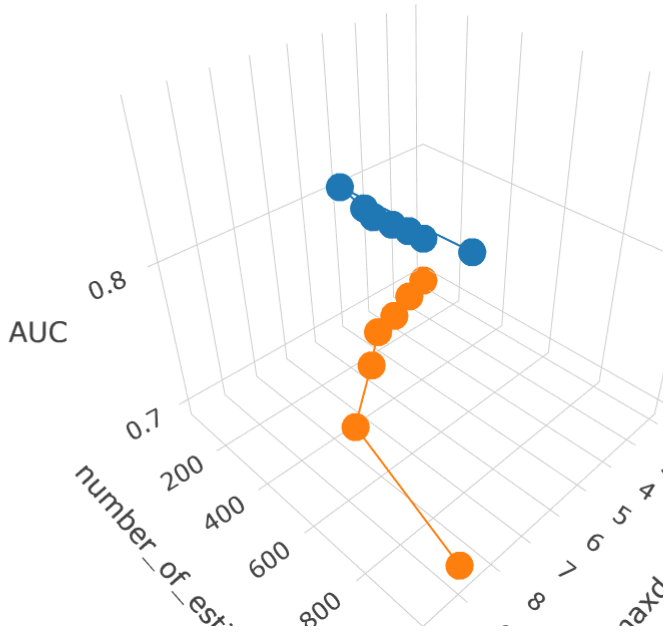
In [128]:

```
#plotting error plots
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import RandomForestClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8,9,10]
number_of_estimators=[50,100,150,200,300,500,1000]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        RandomForest=RandomForestClassifier(n_estimators=i,max_depth=j,class_weight='ba
lanced')
        calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
        calib_cv.fit(final_train_tfidf,y_train)

        y_tr_pred=calib_cv.predict_proba(final_train_tfidf)[:,1]
        y_cv_pred=calib_cv.predict_proba(final_cv_tfidf)[:,1]

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

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()

layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

```
  0%|
| 0/7 [00:00<?, ?it/s]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:07<00:53,  7.63s/it]
 25%|
| 2/8 [00:15<00:46,  7.77s/it]
 38%|
| 3/8 [00:24<00:40,  8.02s/it]
 50%|
| 4/8 [00:33<00:33,  8.26s/it]
 62%|
| 5/8 [00:42<00:25,  8.58s/it]
 75%|
| 6/8 [00:53<00:18,  9.22s/it]
 88%|
    | 7/8 [01:04<00:09,  9.97s/it]
 14%|
| 1/7 [01:18<07:48, 78.03s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:14<01:41, 14.56s/it]
 25%|
| 2/8 [00:30<01:29, 14.88s/it]
 38%|
| 3/8 [00:46<01:16, 15.28s/it]
 50%|
| 4/8 [01:03<01:02, 15.75s/it]
 62%|
| 5/8 [01:21<00:49, 16.41s/it]
 75%|
| 6/8 [01:41<00:35, 17.68s/it]
 88%|
    | 7/8 [02:05<00:19, 19.43s/it]
 29%|
| 2/7 [03:47<08:16, 99.35s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:20<02:26, 20.87s/it]
 25%|
| 2/8 [00:43<02:08, 21.35s/it]
 38%|
| 3/8 [01:07<01:50, 22.12s/it]
 50%|
| 4/8 [01:33<01:33, 23.31s/it]
 62%|
| 5/8 [01:59<01:12, 24.28s/it]
 75%|
| 6/8 [02:31<00:52, 26.38s/it]
 88%|
    | 7/8 [03:04<00:28, 28.51s/it]
 43%|
| 3/7 [07:28<09:03, 135.86s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:27<03:14, 27.77s/it]
 25%|
```

```
| 2/8 [00:57<02:49, 28.23s/it]
 38%|
| 3/8 [01:28<02:26, 29.21s/it]
 50%|
| 4/8 [02:01<02:01, 30.47s/it]
 62%|
| 5/8 [02:38<01:36, 32.15s/it]
 75%|
| 6/8 [03:18<01:09, 34.76s/it]
 88%|
          | 7/8 [04:04<00:38, 38.04s/it]
 57%|
| 4/7 [12:20<09:08, 182.89s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:40<04:46, 40.92s/it]
 25%|
| 2/8 [01:24<04:10, 41.68s/it]
 38%|
| 3/8 [02:10<03:35, 43.00s/it]
 50%|
| 4/8 [02:59<02:59, 44.83s/it]
 62%|
| 5/8 [03:51<02:20, 46.99s/it]
 75%|
| 6/8 [04:52<01:42, 51.14s/it]
 88%|
          | 7/8 [05:57<00:55, 55.42s/it]
 71%|
| 5/7 [19:28<08:32, 256.40s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [01:06<07:48, 66.95s/it]
 25%|
| 2/8 [02:20<06:53, 68.87s/it]
 38%|
| 3/8 [03:36<05:55, 71.14s/it]
 50%|
| 4/8 [04:58<04:57, 74.45s/it]
 62%|
| 5/8 [06:26<03:54, 78.31s/it]
 75%|
| 6/8 [08:08<02:50, 85.37s/it]
 88%|
          | 7/8 [09:58<01:32, 93.00s/it]
 86%|
           | 6/7 [31:22<06:33, 393.60s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [02:12<15:27, 132.56s/it]
 25%|
| 2/8 [04:32<13:29, 134.91s/it]
 38%|
| 3/8 [07:04<11:39, 139.86s/it]
 50%|
| 4/8 [09:44<09:43, 145.99s/it]
 62%|
| 5/8 [12:36<07:41, 153.70s/it]
```

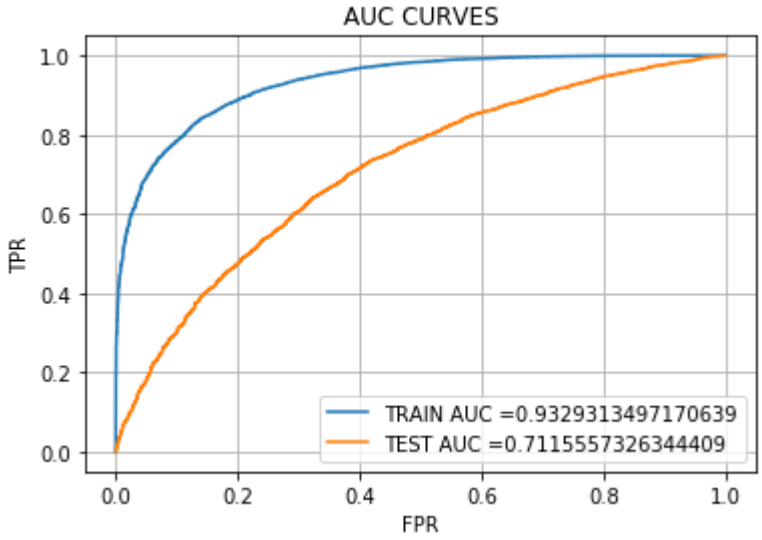
```
 75%|
| 6/8 [15:53<05:33, 166.87s/it]
 88%|
    | 7/8 [19:27<03:00, 180.78s/it]
100%|
| 7/7 [54:38<00:00, 694.19s/it]
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators in the range 2 to 10 and 50 to 1000 respectively and we choose our best max depth and number of estimators to be 3 and 150 respectively.

In [95]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
RandomForest=RandomForestClassifier(max_depth=3,n_estimators=150,class_weight='balance
d')
calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
calib_cv.fit(final_train_tfidf,y_train)

y_train_pred=calib_cv.predict_proba(final_train_tfidf)[:,1]
y_test_pred=calib_cv.predict_proba(final_test_tfidf)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(y_train,y_tr_pred)
test_fpr,test_tpr,te_threshold=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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=3 and number of estimators=150 we got train auc of 93.29% which is pretty good and test auc of 71.11%. And our model is overfitting as the gap between train and test auc scores is a bit high.

In [96]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(y_train,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(y_test,predict_with_best_t(y_test_pred,best_t)))
```

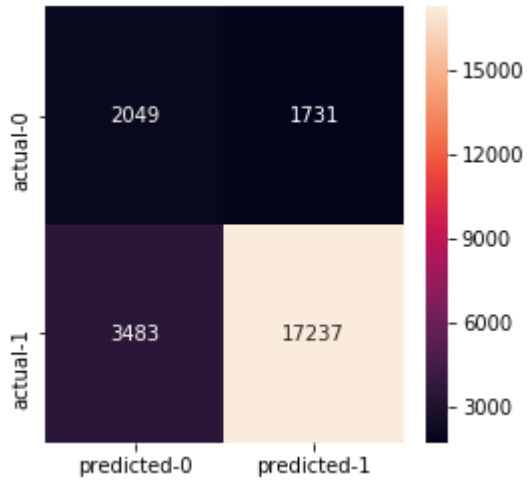
```
====================================================================================================
the maximum value of tpr*(1-fpr) 0.7236595268738126 for threshold 0.794
TRAIN CONFUSION MATRIX
[[ 2049  1731]
 [ 3483 17237]]
test confusion matrix
[[ 1124  1190]
 [ 2538 10148]]
```

In [97]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[2049,1731],[3483,17237]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[97]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee0b9195c0>
```



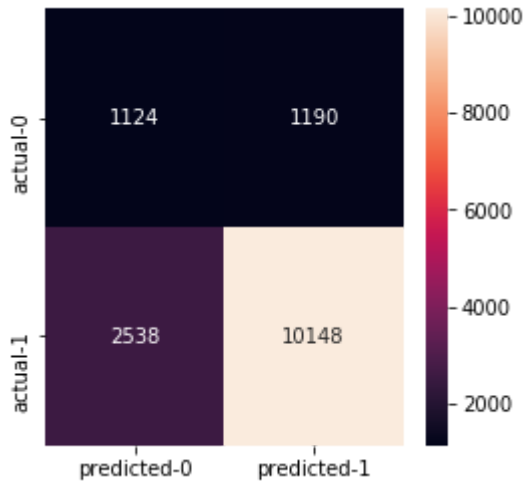
OBSERVATIONS: For max depth =3 and number of estimators=150 the tpr and tnr values for train data set are good.

In [98]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[1124,1190],[2538,10148]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[98]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee1b4d43c8>
```



OBSERVATIONS: For max depth =3 and number of estimators=150 the tpr value for test data set are good.

2.4.3 Applying Random Forests on AVG W2V, SET 3

In [0]:

```
# Please write all the code with proper documentation
```

In [75]:

```
with open('glove_vectors', 'rb') as f:
    model = pickle.load(f)
    glove_words =  set(model.keys())
```

In [76]:

```
# average Word2Vec
# compute average word2vec for each review.
from tqdm import tqdm
avg_w2v_train = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_train['essay'][0:10000].values): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_train.append(vector)

print(len(avg_w2v_train))
print(len(avg_w2v_train[0]))
#print(avg_w2v_train[0])
```

```
100%|████████████████████████████████████████████████████████████████
█████| 10000/10000 [00:03<00:00, 2832.00it/s]

10000
300
```

In [77]:

```
avg_w2v_cv = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_cv['essay'][0:5000].values): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_cv.append(vector)

print(len(avg_w2v_cv))
print(len(avg_w2v_cv[0]))
```

```
100%|████████████████████████████████████████████████████████████████
███████| 5000/5000 [00:01<00:00, 2882.26it/s]

5000
300
```

In [78]:

```
avg_w2v_test =[];
for sentence in tqdm(X_test['essay'][0:5000].values):
    vector=np.zeros(300)
    cnt_words = 0;
    for word in sentence.split():
        if word in glove_words:
            vector += model[word]
            cnt_words +=1
    if cnt_words != 0:
            vector /= cnt_words
    avg_w2v_test.append(vector)
print(len(avg_w2v_test))
```

```
100%|████████████████████████████████████████████████████████████████████
███████| 5000/5000 [00:01<00:00, 2885.23it/s]

5000
```

In [79]:

```
title_train_avgw2v=[]
for sentence in tqdm(X_train['project_title'][0:10000].values):
    vector=np.zeros(300)
    cnt_words=0;
    for word in sentence.split():
        if word in glove_words:
            vector+=model[word]
            cnt_words+=1
    if cnt_words !=0:
        vector/=cnt_words
    title_train_avgw2v.append(vector)
print(len(title_train_avgw2v))
print(len(title_train_avgw2v[0]))
```

```
100%|████████████████████████████████████████████████████████████████████
████| 10000/10000 [00:00<00:00, 89311.20it/s]

10000
300
```

In [80]:

```
title_cv_avgw2v=[]
for sentence in tqdm(X_cv['project_title'][0:5000].values):
    vector=np.zeros(300)
    cnt_words=0;
    for word in sentence.split():
        if word in glove_words:
            vector+=model[word]
            cnt_words+=1
    if cnt_words !=0:
        vector/=cnt_words
    title_cv_avgw2v.append(vector)

print(len(title_cv_avgw2v))
print(len(title_cv_avgw2v[0]))
```

```
100%|████████████████████████████████████████████████████████████████
█████| 5000/5000 [00:00<00:00, 104191.81it/s]

5000
300
```

In [81]:

```
title_test_avgw2v=[]
for sentence in tqdm(X_test['project_title'][0:5000].values):
    vector=np.zeros(300)
    cnt_ords=0;
    for word in sentence.split():
        if word in glove_words:
            vector+=model[word]
            cnt_words+=1
    if cnt_words !=0:
        vector/=cnt_words
    title_test_avgw2v.append(vector)
print(len(title_test_avgw2v))
```

```
100%|████████████████████████████████████████████████████████████████
██████| 5000/5000 [00:00<00:00, 65806.63it/s]

5000
```

In [88]:

```
#creating final data matrix
from scipy.sparse import hstack
final_train_avgw2v=hstack((final_train[0:10000],price_train_norm[0:10000],train_quantit
y_norm[0:10000],projects_train_norm[0:10000],avg_w2v_train,title_train_avgw2v)).tocsr()
final_cv_avgw2v=hstack((final_cv[0:5000],price_cv_norm[0:5000],cv_quantity_norm[0:5000
],projects_cv_norm[0:5000],avg_w2v_cv,title_cv_avgw2v)).tocsr()
final_test_avgw2v=hstack((final_test[0:5000],price_test_norm[0:5000],test_quantity_norm
[0:5000],projects_test_norm[0:5000],avg_w2v_test,title_test_avgw2v)).tocsr()
print(final_train_avgw2v.shape,y_train[0:1000].shape)
print(final_cv_avgw2v.shape,y_cv[0:5000].shape)
print(final_test_avgw2v.shape,y_test[0:5000].shape)
```

```
(10000, 613) (1000,)
(5000, 613) (5000,)
(5000, 613) (5000,)
```

In [89]:

```
y_train_avgw2v=y_train[0:10000]
y_cv_avgw2v=y_cv[0:5000]
y_test_avgw2v=y_test[0:5000]
```

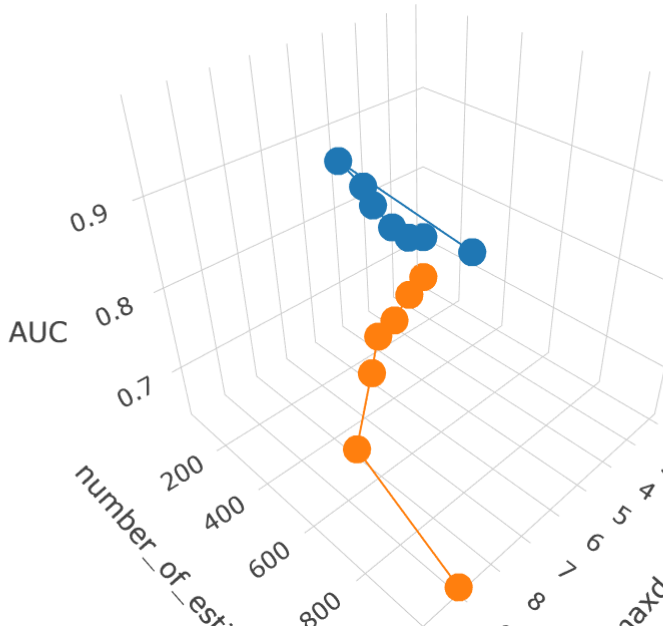
In [104]:

```
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import RandomForestClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8,9,10]
number_of_estimators=[50,100,150,200,300,500,1000]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        RandomForest=RandomForestClassifier(n_estimators=i,max_depth=j,class_weight='ba
lanced')
        calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
        calib_cv.fit(final_train_avgw2v,y_train_avgw2v)

        y_tr_pred=calib_cv.predict_proba(final_train_avgw2v)[:,1]
        y_cv_pred=calib_cv.predict_proba(final_cv_avgw2v)[:,1]

        train_auc.append(roc_auc_score(y_train_avgw2v,y_tr_pred))
        cv_auc.append(roc_auc_score(y_cv_avgw2v,y_cv_pred))

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()

layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

```
  0%|
| 0/7 [00:00<?, ?it/s]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:06<00:42,  6.04s/it]
 25%|
| 2/8 [00:13<00:38,  6.47s/it]
 38%|
| 3/8 [00:23<00:37,  7.40s/it]
 50%|
| 4/8 [00:34<00:34,  8.71s/it]
 62%|
| 5/8 [00:49<00:31, 10.49s/it]
 75%|
| 6/8 [01:10<00:26, 13.49s/it]
 88%|
| 7/8 [01:32<00:16, 16.31s/it]
 14%|
| 1/7 [01:58<11:53, 118.98s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:11<01:21, 11.58s/it]
 25%|
| 2/8 [00:26<01:15, 12.56s/it]
 38%|
| 3/8 [00:45<01:13, 14.61s/it]
 50%|
| 4/8 [01:10<01:10, 17.62s/it]
 62%|
| 5/8 [01:38<01:02, 20.87s/it]
 75%|
| 6/8 [02:19<00:53, 26.91s/it]
 88%|
| 7/8 [03:07<00:33, 33.06s/it]
 29%|
| 2/7 [05:58<12:55, 155.11s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:19<02:14, 19.17s/it]
 25%|
| 2/8 [00:44<02:06, 21.00s/it]
 38%|
| 3/8 [01:11<01:54, 22.88s/it]
 50%|
| 4/8 [01:45<01:44, 26.02s/it]
 62%|
| 5/8 [02:28<01:33, 31.31s/it]
 75%|
| 6/8 [03:28<01:19, 39.84s/it]
 88%|
| 7/8 [04:34<00:47, 47.74s/it]
 43%|
| 3/7 [11:54<14:21, 215.35s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:23<02:41, 23.10s/it]
 25%|
```

```
| 2/8 [00:52<02:29, 24.94s/it]
 38%|
| 3/8 [01:29<02:23, 28.74s/it]
 50%|
| 4/8 [02:16<02:16, 34.03s/it]
 62%|
| 5/8 [03:12<02:02, 40.76s/it]
 75%|
| 6/8 [04:32<01:45, 52.53s/it]
 88%|
    | 7/8 [06:06<01:04, 64.83s/it]
 57%|
| 4/7 [19:45<14:36, 292.05s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:34<03:58, 34.14s/it]
 25%|
| 2/8 [01:19<03:45, 37.55s/it]
 38%|
| 3/8 [02:15<03:34, 42.91s/it]
 50%|
| 4/8 [03:24<03:23, 50.79s/it]
 62%|
| 5/8 [04:48<03:02, 60.82s/it]
 75%|
| 6/8 [06:48<02:36, 78.47s/it]
 88%|
    | 7/8 [09:05<01:36, 96.27s/it]
 71%|
| 5/7 [31:27<13:49, 414.96s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:56<06:33, 56.21s/it]
 25%|
| 2/8 [02:09<06:07, 61.21s/it]
 38%|
| 3/8 [03:40<05:50, 70.18s/it]
 50%|
| 4/8 [05:35<05:35, 83.85s/it]
 62%|
| 5/8 [07:56<05:02, 100.91s/it]
 75%|
| 6/8 [11:15<04:20, 130.37s/it]
 88%|
    | 7/8 [15:06<02:40, 160.46s/it]
 86%|
    | 6/7 [50:53<10:40, 640.51s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [01:53<13:16, 113.81s/it]
 25%|
| 2/8 [04:19<12:20, 123.36s/it]
 38%|
| 3/8 [07:23<11:48, 141.60s/it]
 50%|
| 4/8 [11:12<11:10, 167.66s/it]
 62%|
| 5/8 [15:51<10:03, 201.17s/it]
```

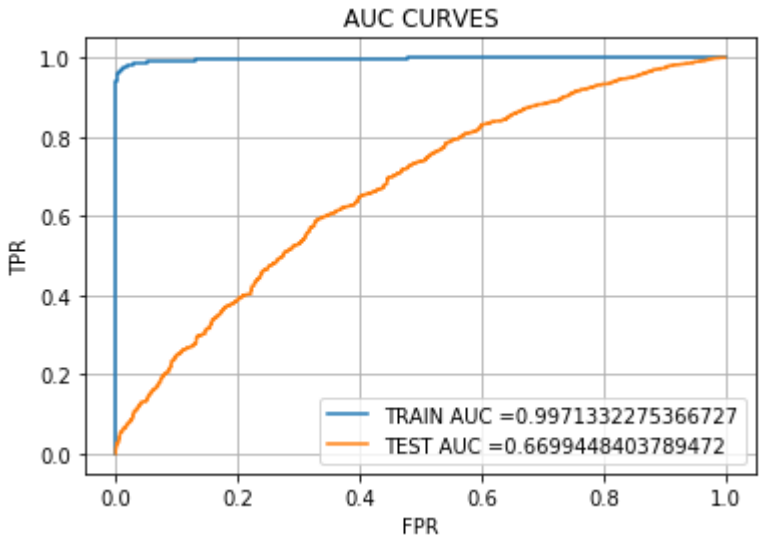
```
 75%|
| 6/8 [22:28<08:40, 260.04s/it]
 88%|
    | 7/8 [30:09<05:20, 320.28s/it]
100%|
| 7/7 [1:29:45<00:00, 1147.90s/it]
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators in the range 2 to 10 and 50 to 1000 respectively and we choose our best max depth and number of estimators to be 2 and 50 respectively.

In [108]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
RandomForest=RandomForestClassifier(max_depth=2,n_estimators=50,class_weight='balanced'
)
calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
calib_cv.fit(final_train_avgw2v,y_train_avgw2v)

y_train_pred=calib_cv.predict_proba(final_train_avgw2v)[:,1]
y_test_pred=calib_cv.predict_proba(final_test_avgw2v)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(y_train_avgw2v,y_tr_pred)
test_fpr,test_tpr,te_threshold=roc_curve(y_test_avgw2v,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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=2 and number of estimators=50 we got train auc of 99.7% which is perfect almost and test auc of 66.99%. And our model is overfitting as the gap between train and test auc scores is a bit high.

In [111]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(y_train_avgw2v,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(y_test_avgw2v,predict_with_best_t(y_test_pred,best_t)))
```

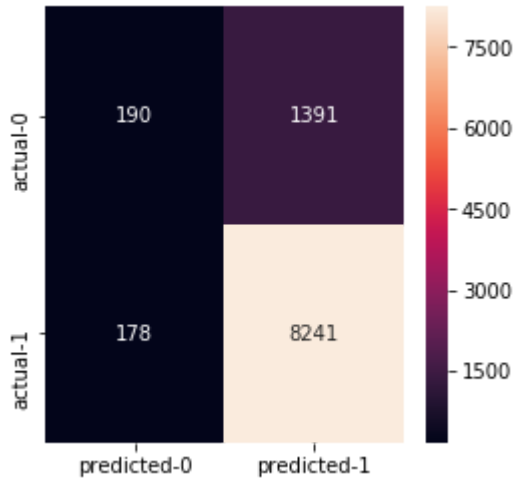
```
====================================================================================================
the maximum value of tpr*(1-fpr) 0.9638630251038302 for threshold 0.707
TRAIN CONFUSION MATRIX
[[ 190 1391]
 [ 178 8241]]
test confusion matrix
[[  75  682]
 [ 111 4132]]
```

In [112]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[190,1391],[178,8241]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[112]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25a9ae3a710>
```



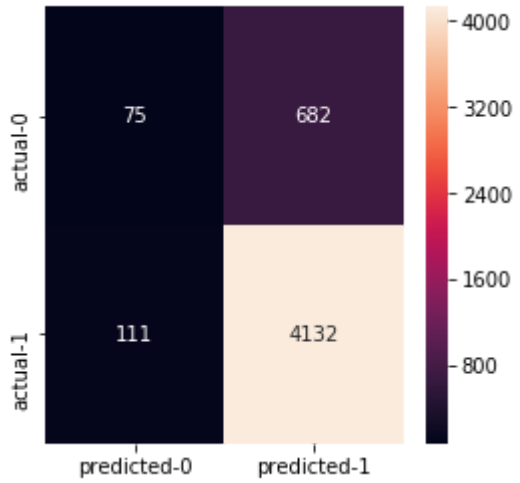
OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for train data set is good and fnr is a bit high.

In [113]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[75,682],[111,4132]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[113]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25a9a5b3d68>
```



OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for test data set is good.

2.4.4 Applying Random Forests on TFIDF W2V, SET 4

In [0]:

```
# Please write all the code with proper documentation
```

In [100]:

```
# Please write all the code with proper documentation
from tqdm import tqdm
preprocessed_essays_train = []
# tqdm is for printing the status bar
for sentance in tqdm(X_train['essay'][0:10000].values):
    sent = decontracted(sentance)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    sent = sent.lower()
    # https://gist.github.com/sebleier/554280
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    preprocessed_essays_train.append(sent.lower().strip())
```

```
100%|████████████████████████████████████████████████████████████████
█████| 10000/10000 [00:09<00:00, 1071.67it/s]
```

In [101]:

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

In [102]:

```
# compute average word2vec for each review.
tfidf_w2v_train = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in (X_train['essay'][0:10000]): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf value((sen
tence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split())) # ge
tting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_train.append(vector)

print(len(tfidf_w2v_train))
print(len(tfidf_w2v_train[0]))
```

```
10000
300
```

In [103]:

```
tfidf_w2v_cv=[]
for sentence in tqdm(X_cv['essay'][0:5000]):
    vector=np.zeros(300)
    tf_idf_weight=0;
    for word in sentence.split():
        if (word in glove_words) and (word in tfidf_words):
            vec=model[word]
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split()))
            vector +=(vec * tf_idf)
            tf_idf_weight+= tf_idf
    if tf_idf_weight !=0:
        vector /=tf_idf_weight
    tfidf_w2v_cv.append(vector)

print(len(tfidf_w2v_cv))
print(len(tfidf_w2v_cv[0]))
```

```
100%|████████████████████████████████████████████████████████████████████
████████| 5000/5000 [00:12<00:00, 386.74it/s]

5000
300
```

In [104]:

```
tfidf_w2v_test=[]
for sentence in tqdm(X_test['essay'][0:5000]):
    vector=np.zeros(300)
    tf_idf_weight=0;
    for word in sentence.split():
        if (word in glove_words)and(word in tfidf_words):
            vec=model[word]
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split()))
            vector+=(vec*tf_idf)
            tf_idf_weight+=tf_idf
    if tf_idf_weight !=0:
        vector /=tf_idf_weight
    tfidf_w2v_test.append(vector)
print(len(tfidf_w2v_test))
print(len(tfidf_w2v_test[0]))
```

```
100%|████████████████████████████████████████████████████████████████████
████████| 5000/5000 [00:13<00:00, 380.37it/s]

5000
300
```

In [105]:

```
#tfidf weighted w2v vectorizing project title
title_train_tfidfw2v=[]
for sentence in tqdm(X_train['project_title'][0:10000]):
    vector=np.zeros(300)
    tf_idf_weight=0;
    for word in sentence.split():
        if(word in glove_words)and(word in tfidf_words):
            vec=model[word]
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split()))
            vector+=(vec*tf_idf)
            tf_idf_weight+=tf_idf
    if tf_idf_weight !=0:
        vector /=tf_idf_weight
    title_train_tfidfw2v.append(vector)
print(len(title_train_tfidfw2v))
print(len(title_train_tfidfw2v[0]))
```

```
100%|████████████████████████████████████████████████████████████████████
████| 10000/10000 [00:00<00:00, 76377.69it/s]

10000
300
```

In [106]:

```
title_cv_tfidfw2v=[]
for sentence in tqdm(X_cv['project_title'][0:5000]):
    vector=np.zeros(300)
    tf_idf_weight=0;
    for word in sentence.split():
        if (word in glove_words) and (word in tfidf_words):
            vec=model[word]
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split()))
            vector+=(vec*tf_idf)
            tf_idf_weight+=tf_idf
    if tf_idf_weight !=0:
        vector /=tf_idf_weight
    title_cv_tfidfw2v.append(vector)
print(len(title_cv_tfidfw2v))
print(len(title_cv_tfidfw2v[0]))
```

```
100%|████████████████████████████████████████████████████████████████████
██████| 5000/5000 [00:00<00:00, 78170.56it/s]

5000
300
```

In [107]:

```
title_test_tfidfw2v=[]
for sentence in tqdm(X_test['project_title'][0:5000]):
    vector=np.zeros(300)
    tf_idf_weight=0;
    for word in sentence.split():
        if (word in glove_words)and(word in tfidf_words):
            vec=model[word]
            tf_idf=dictionary[word]*(sentence.count(word)/len(sentence.split()))
            vector+=(vec*tf_idf)
            tf_idf_weight+=tf_idf
    if tf_idf_weight !=0:
        vector /=tf_idf_weight
    title_test_tfidfw2v.append(vector)

print(len(title_test_tfidfw2v))
print(len(title_test_tfidfw2v[0]))
```

```
100%|████████████████████████████████████████████████████████████████████
██████| 5000/5000 [00:00<00:00, 83380.79it/s]

5000
300
```

In [108]:

```
#creating final data matrix
from scipy.sparse import hstack
final_train_tfidfw2v=hstack((final_train[0:10000],price_train_norm[0:10000],train_quant
ity_norm[0:10000],projects_train_norm[0:10000],tfidf_w2v_train,title_train_tfidfw2v)).t
ocsr()
final_cv_tfidfw2v=hstack((final_cv[0:5000],price_cv_norm[0:5000],cv_quantity_norm[0:500
0],projects_cv_norm[0:5000],tfidf_w2v_cv,title_cv_tfidfw2v)).tocsr()
final_test_tfidfw2v=hstack((final_test[0:5000],price_test_norm[0:5000],test_quantity_no
rm[0:5000],projects_test_norm[0:5000],tfidf_w2v_test,title_test_tfidfw2v)).tocsr()
print(final_train_tfidfw2v.shape,y_train[0:1000].shape)
print(final_cv_tfidfw2v.shape,y_cv[0:5000].shape)
print(final_test_tfidfw2v.shape,y_test[0:5000].shape)
```

```
(10000, 613) (1000,)
(5000, 613) (5000,)
(5000, 613) (5000,)
```

In [109]:

```
y_train_tfidfw2v=y_train[0:10000]
y_cv_tfidfw2v=y_cv[0:5000]
y_test_tfidfw2v=y_test[0:5000]
```

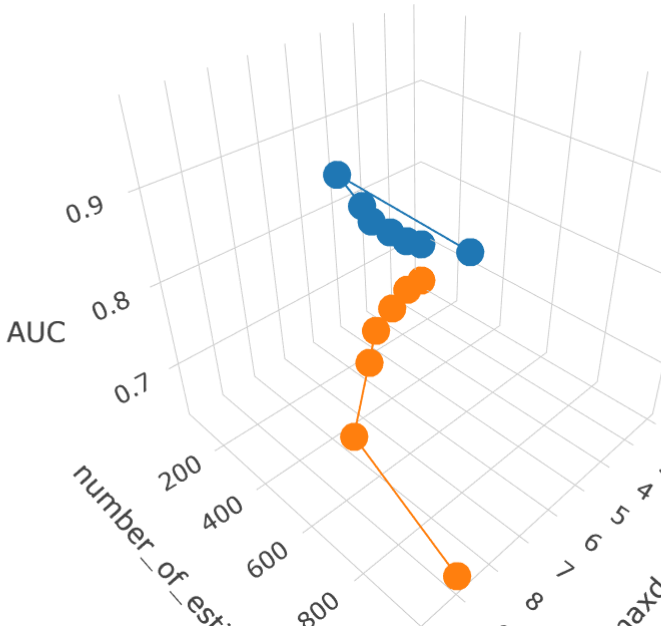
In [124]:

```
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import RandomForestClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8,9,10]
number_of_estimators=[50,100,150,200,300,500,1000]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        RandomForest=RandomForestClassifier(n_estimators=i,max_depth=j,class_weight='ba
lanced')
        calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
        calib_cv.fit(final_train_tfidfw2v,y_train_tfidfw2v)

        y_tr_pred=calib_cv.predict_proba(final_train_tfidfw2v)[:,1]
        y_cv_pred=calib_cv.predict_proba(final_cv_tfidfw2v)[:,1]

        train_auc.append(roc_auc_score(y_train_tfidfw2v,y_tr_pred))
        cv_auc.append(roc_auc_score(y_cv_tfidfw2v,y_cv_pred))

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()

layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

```
  0%|
| 0/7 [00:00<?, ?it/s]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:06<00:44,  6.36s/it]
 25%|
| 2/8 [00:14<00:40,  6.81s/it]
 38%|
| 3/8 [00:23<00:38,  7.68s/it]
 50%|
| 4/8 [00:35<00:35,  8.92s/it]
 62%|
| 5/8 [00:50<00:31, 10.55s/it]
 75%|
| 6/8 [01:10<00:26, 13.45s/it]
 88%|
| 7/8 [01:38<00:17, 17.79s/it]
 14%|
| 1/7 [02:12<13:13, 132.20s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:14<01:43, 14.83s/it]
 25%|
| 2/8 [00:32<01:33, 15.61s/it]
 38%|
| 3/8 [00:54<01:28, 17.67s/it]
 50%|
| 4/8 [01:22<01:22, 20.56s/it]
 62%|
| 5/8 [01:54<01:12, 24.16s/it]
 75%|
| 6/8 [02:39<01:00, 30.43s/it]
 88%|
| 7/8 [03:31<00:36, 36.76s/it]
 29%|
| 2/7 [06:44<14:31, 174.22s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:20<02:20, 20.04s/it]
 25%|
| 2/8 [00:45<02:10, 21.72s/it]
 38%|
| 3/8 [01:18<02:05, 25.03s/it]
 50%|
| 4/8 [01:58<01:58, 29.57s/it]
 62%|
| 5/8 [02:49<01:47, 35.82s/it]
 75%|
| 6/8 [03:51<01:27, 43.90s/it]
 88%|
| 7/8 [04:57<00:50, 50.52s/it]
 43%|
| 3/7 [12:55<15:32, 233.23s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:21<02:27, 21.11s/it]
 25%|
```

```
| 2/8 [00:48<02:18, 23.04s/it]
 38%|
| 3/8 [01:23<02:13, 26.67s/it]
 50%|
| 4/8 [02:07<02:06, 31.63s/it]
 62%|
| 5/8 [03:00<01:54, 38.05s/it]
 75%|
| 6/8 [04:14<01:37, 48.88s/it]
 88%|
    | 7/8 [05:40<01:00, 60.09s/it]
 57%|
| 4/7 [20:13<14:44, 294.83s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:33<03:53, 33.41s/it]
 25%|
| 2/8 [01:14<03:34, 35.72s/it]
 38%|
| 3/8 [02:07<03:24, 40.93s/it]
 50%|
| 4/8 [03:13<03:13, 48.37s/it]
 62%|
| 5/8 [04:33<02:53, 57.88s/it]
 75%|
| 6/8 [07:15<02:58, 89.15s/it]
 88%|
    | 7/8 [10:20<01:57, 117.90s/it]
 71%|
| 5/7 [33:52<15:03, 451.97s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [01:03<07:21, 63.06s/it]
 25%|
| 2/8 [02:23<06:49, 68.20s/it]
 38%|
| 3/8 [04:00<06:24, 76.85s/it]
 50%|
| 4/8 [05:55<05:52, 88.22s/it]
 62%|
| 5/8 [08:06<05:03, 101.26s/it]
 75%|
| 6/8 [11:14<04:14, 127.16s/it]
 88%|
    | 7/8 [14:49<02:33, 153.62s/it]
 86%|
   | 6/7 [52:45<10:56, 656.34s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [01:47<12:30, 107.27s/it]
 25%|
| 2/8 [04:05<11:39, 116.55s/it]
 38%|
| 3/8 [06:59<11:08, 133.66s/it]
 50%|
| 4/8 [10:34<10:32, 158.19s/it]
 62%|
| 5/8 [14:57<09:29, 189.79s/it]
```

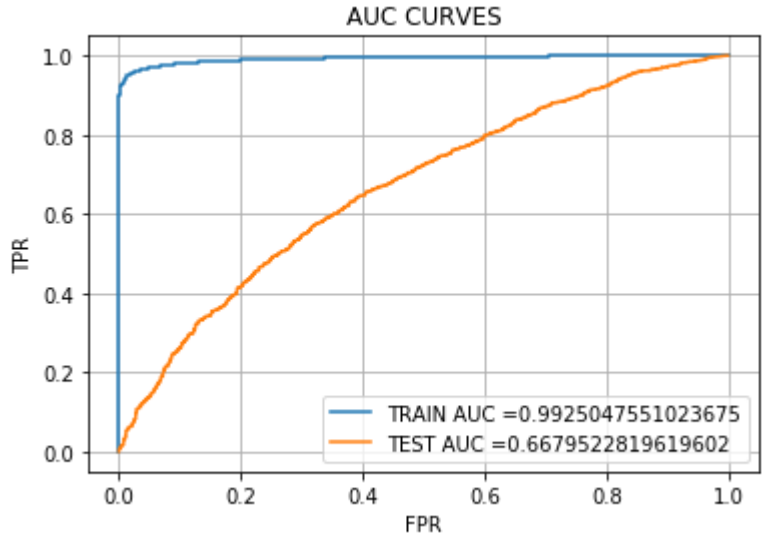
```
 75%|
| 6/8 [21:11<08:10, 245.04s/it]
 88%|
    | 7/8 [28:25<05:01, 301.69s/it]
100%|
| 7/7 [1:29:20<00:00, 1118.00s/it]
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators in the range 2 to 10 and 50 to 1000 respectively and we choose our best max depth and number of estimators to be 2 and 50 respectively.

In [127]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
RandomForest=RandomForestClassifier(max_depth=2,n_estimators=50,class_weight='balanced'
)
calib_cv=CalibratedClassifierCV(base_estimator=RandomForest)
calib_cv.fit(final_train_tfidfw2v,y_train_tfidfw2v)

y_train_pred=calib_cv.predict_proba(final_train_tfidfw2v)[:,1]
y_test_pred=calib_cv.predict_proba(final_test_tfidfw2v)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(y_train_tfidfw2v,y_tr_pred)
test_fpr,test_tpr,te_threshold=roc_curve(y_test_tfidfw2v,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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=2 and number of estimators=50 we got train auc of 99.25% which is perfect almost and test auc of 66.79%. And our model is overfitting as the gap between train and test auc scores is a bit high.

In [128]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(y_train_tfidfw2v,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(y_test_tfidfw2v,predict_with_best_t(y_test_pred,best_t)))
```

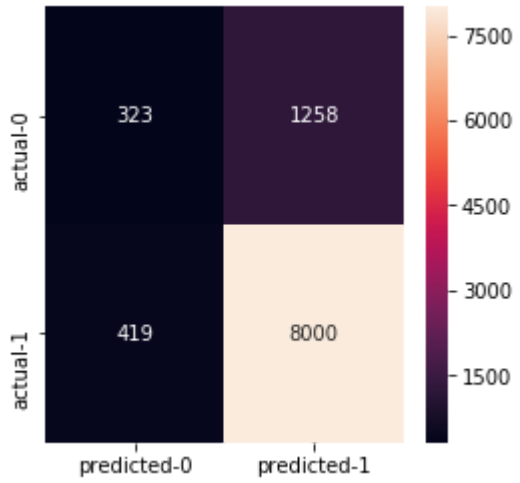
```
====================================================================================================
the maximum value of tpr*(1-fpr) 0.9366065236465905 for threshold 0.746
TRAIN CONFUSION MATRIX
[[ 323 1258]
 [ 419 8000]]
test confusion matrix
[[ 149  608]
 [ 314 3929]]
```

In [129]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[323,1258],[419,8000]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[129]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25a9a0ef908>
```



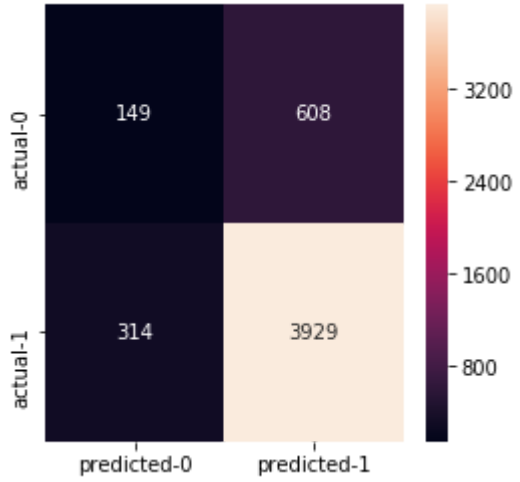
OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for train data set is good and fnr is a bit high.

In [130]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[149,608],[314,3929]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[130]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25a9c9fee48>
```



OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for test data set is good and fnr is a bit high.

2.5 Applying GBDT

Apply GBDT on different kind of featurization as mentioned in the instructions
For Every model that you work on make sure you do the step 2 and step 3 of instrucations

2.5.1 Applying XGBOOST on BOW, SET 1

In [0]:

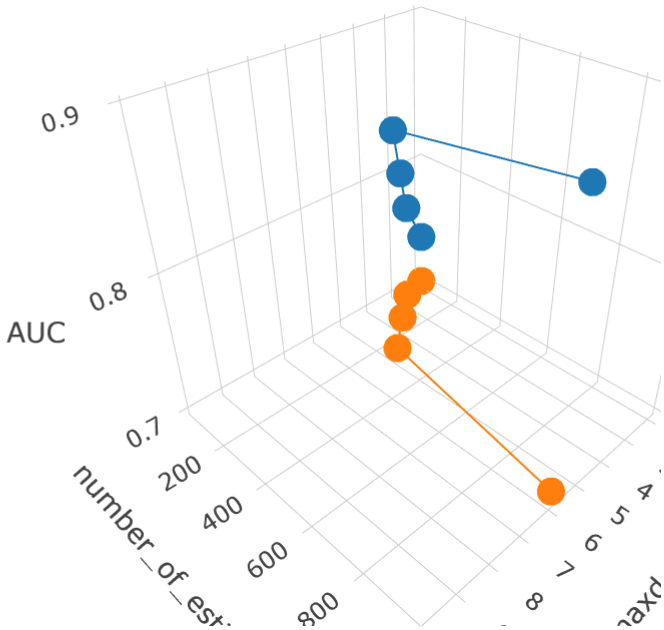
```
# Please write all the code with proper documentation
```

In [86]:

```
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import GradientBoostingClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8,10]
number_of_estimators=[50,100,200,300,1000]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        GBDT=GradientBoostingClassifier(n_estimators=i,max_depth=j,loss='deviance',lear
ning_rate=0.1)
        calib_cv=CalibratedClassifierCV(base_estimator=GBDT)
        calib_cv.fit(final_train_bow,y_train)

        y_tr_pred=calib_cv.predict_proba(final_train_bow)[:,1]
        y_cv_pred=calib_cv.predict_proba(final_cv_bow)[:,1]



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

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()

layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

```
  0%|
| 0/5 [00:00<?, ?it/s]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [00:33<03:23, 33.99s/it]
 29%|
| 2/7 [01:28<03:20, 40.11s/it]
 43%|
| 3/7 [02:48<03:28, 52.08s/it]
 57%|
| 4/7 [04:39<03:29, 69.71s/it]
 71%|
| 5/7 [07:13<03:10, 95.17s/it]
 86%|
| 6/7 [11:04<02:15, 135.94s/it]
 20%|
| 1/5 [16:29<1:05:58, 989.51s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [01:05<06:31, 65.27s/it]
 29%|
| 2/7 [02:50<06:25, 77.13s/it]
 43%|
| 3/7 [05:15<06:30, 97.63s/it]
 57%|
| 4/7 [08:29<06:19, 126.52s/it]
 71%|
| 5/7 [12:38<05:26, 163.36s/it]
 86%|
| 6/7 [18:42<03:43, 223.60s/it]
 40%|
| 2/5 [43:40<59:05, 1181.87s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [02:07<12:43, 127.23s/it]
 29%|
| 2/7 [05:24<12:20, 148.14s/it]
 43%|
| 3/7 [09:57<12:22, 185.57s/it]
 57%|
| 4/7 [15:49<11:46, 235.66s/it]
 71%|
| 5/7 [23:09<09:53, 296.98s/it]
 86%|
| 6/7 [33:38<06:36, 396.43s/it]
 60%|
| 3/5 [1:31:22<56:12, 1686.14s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [03:09<18:55, 189.25s/it]
 29%|
| 2/7 [07:57<18:14, 218.86s/it]
 43%|
| 3/7 [14:32<18:07, 271.77s/it]
 57%|
| 4/7 [22:59<17:07, 342.39s/it]
 71%|
```

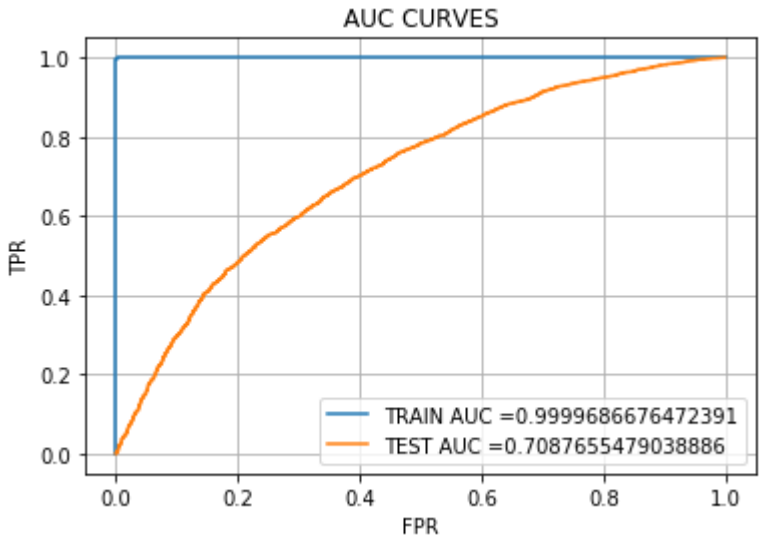
```
| 5/7 [33:26<14:15, 427.65s/it]
 86%|
| 6/7 [48:12<09:25, 565.25s/it]
 80%|
| 4/5 [2:39:07<39:59, 2399.80s/it]
  0%|
| 0/7 [00:00<?, ?it/s]
 14%|
| 1/7 [10:14<1:01:25, 614.22s/it]
 29%|
| 2/7 [25:38<58:56, 707.36s/it]
 43%|
| 3/7 [46:30<58:02, 870.57s/it]
 57%|
| 4/7 [1:20:58<1:01:29, 1229.86s/it]
 71%|
| 5/7 [1:59:05<51:33, 1546.90s/it]
 86%|
| 6/7 [2:43:52<31:28, 1888.96s/it]
100%|
| 5/5 [6:21:02<00:00, 5674.10s/it]
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators in the range 2 to 10 and 50 to 1000 respectively and we choose our best max depth and number of estimators to be 2 and 50 respectively.

In [88]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
from sklearn.ensemble import GradientBoostingClassifier
GBDT=GradientBoostingClassifier(max_depth=2,n_estimators=50,loss='deviance',learning_ra
te=0.1)
calib_cv=CalibratedClassifierCV(base_estimator=GBDT)
calib_cv.fit(final_train_bow,y_train)

y_train_pred=calib_cv.predict_proba(final_train_bow)[:,1]
y_test_pred=calib_cv.predict_proba(final_test_bow)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(y_train,y_tr_pred)
test_fpr,test_tpr,te_threshold=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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=2 and number of estimators=50 we got train auc of 99.99% which is perfect almost and test auc of 70.87%. And our model is overfitting as the gap between train and test auc scores is a bit high.

In [90]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(y_train,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(y_test,predict_with_best_t(y_test_pred,best_t)))
```

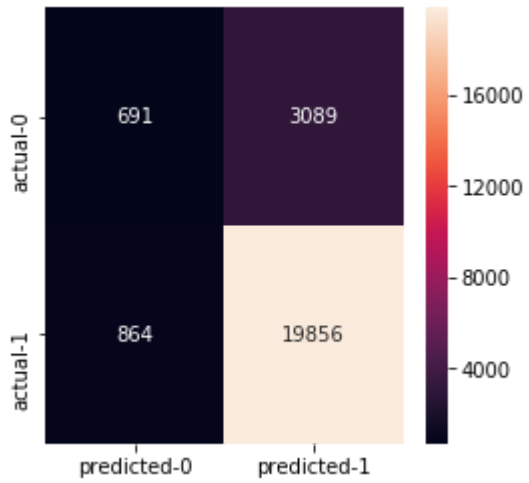
```
====================================================================================================
the maximum value of tpr*(1-fpr) 0.9972007722007722 for threshold 0.662
TRAIN CONFUSION MATRIX
[[  691  3089]
 [  864 19856]]
test confusion matrix
[[  486  1828]
 [  676 12010]]
```

In [91]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[691,3089],[864,19856]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[91]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24b38fc7898>
```



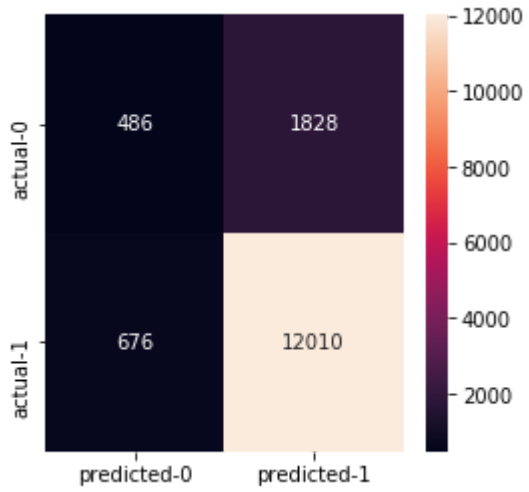
OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for train data set is good and fnr is a bit high.

In [92]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[486,1828],[676,12010]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[92]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24b3907b588>
```



OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for test data set is good and fnr is a bit high.

2.5.2 Applying XGBOOST on TFIDF, SET 2

In [0]:

```
 # Please write all the code with proper documentation
```

In [99]:

```
gbdt_train_tfidf=final_train_tfidf[0:10000]
gbdt_cv_tfidf=final_cv_tfidf[0:5000]
gbdt_test_tfidf=final_test_tfidf[0:5000]
gbdt_ytrain=y_train[0:10000]
gbdt_ycv=y_cv[0:5000]
gbdt_ytest=y_test[0:5000]
```

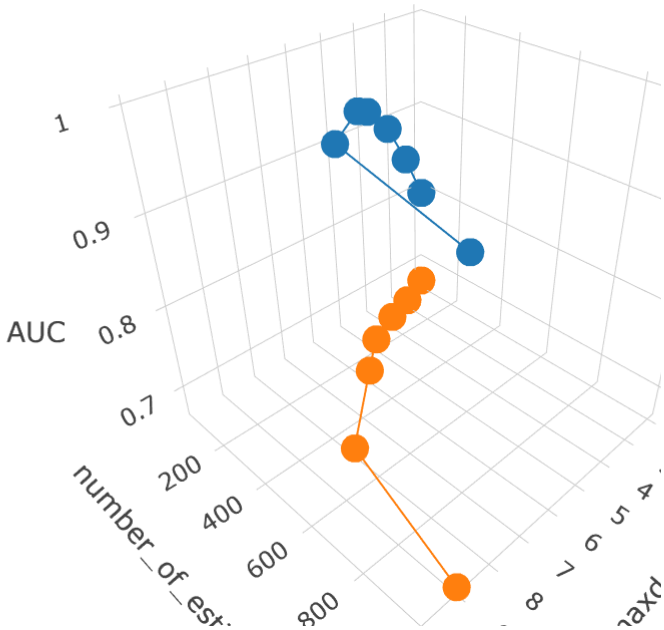
In [100]:

```
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import GradientBoostingClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8,9,10]
number_of_estimators=[50,100,150,200,300,500,1000]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        GBDT_TFIDF= GradientBoostingClassifier(n_estimators=i,max_depth=j,loss='devianc
e',learning_rate=0.1)
        calib_cv=CalibratedClassifierCV(base_estimator=GBDT_TFIDF)
        calib_cv.fit(gbdt_train_tfidf,gbdt_ytrain)

        y_tr_pred=calib_cv.predict_proba(gbdt_train_tfidf)[:,1]
        y_cv_pred=calib_cv.predict_proba(gbdt_cv_tfidf)[:,1]

        train_auc.append(roc_auc_score(gbdt_ytrain,y_tr_pred))
        cv_auc.append(roc_auc_score(gbdt_ycv,y_cv_pred))

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()

layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

```
  0%|
| 0/7 [00:00<?, ?it/s]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [00:42<04:56, 42.41s/it]
 25%|
| 2/8 [01:46<04:53, 48.89s/it]
 38%|
| 3/8 [03:12<05:00, 60.03s/it]
 50%|
| 4/8 [05:00<04:58, 74.55s/it]
 62%|
| 5/8 [07:13<04:36, 92.10s/it]
 75%|
| 6/8 [10:17<03:59, 119.61s/it]
 88%|
| 7/8 [13:50<02:27, 147.53s/it]
 14%|
| 1/7 [17:50<1:47:04, 1070.78s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [01:24<09:49, 84.18s/it]
 25%|
| 2/8 [03:30<09:40, 96.83s/it]
 38%|
| 3/8 [06:19<09:52, 118.57s/it]
 50%|
| 4/8 [09:53<09:47, 146.99s/it]
 62%|
| 5/8 [14:11<09:01, 180.43s/it]
 75%|
| 6/8 [20:04<07:44, 232.07s/it]
 88%|
| 7/8 [26:44<04:42, 282.57s/it]
 29%|
| 2/7 [52:06<1:53:51, 1366.21s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [02:05<14:38, 125.49s/it]
 25%|
| 2/8 [05:14<14:26, 144.41s/it]
 38%|
| 3/8 [09:26<14:43, 176.78s/it]
 50%|
| 4/8 [14:42<14:34, 218.53s/it]
 62%|
| 5/8 [21:02<13:21, 267.14s/it]
 75%|
| 6/8 [29:40<11:24, 342.38s/it]
 88%|
| 7/8 [39:26<06:55, 415.42s/it]
 43%|
| 3/7 [1:42:31<2:04:16, 1864.04s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [02:48<19:37, 168.25s/it]
 25%|
```

```
| 2/8 [06:58<19:16, 192.77s/it]
 38%|
| 3/8 [12:32<19:35, 235.10s/it]
 50%|
| 4/8 [19:29<19:18, 289.74s/it]
 62%|
| 5/8 [27:53<17:42, 354.19s/it]
 75%|
| 6/8 [49:34<21:16, 638.22s/it]
 88%|
          | 7/8 [1:02:30<11:19, 679.33s/it]
 57%|
| 4/7 [2:59:29<2:14:30, 2690.06s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [04:09<29:09, 249.94s/it]
 25%|
| 2/8 [10:22<28:41, 286.87s/it]
 38%|
| 3/8 [18:40<29:10, 350.20s/it]
 50%|
| 4/8 [29:03<28:47, 431.78s/it]
 62%|
| 5/8 [41:35<26:23, 527.84s/it]
 75%|
| 6/8 [1:20:45<35:49, 1074.62s/it]
 88%|
          | 7/8 [1:39:48<18:15, 1095.11s/it]
 71%|
| 5/7 [5:00:44<2:15:31, 4065.64s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [07:00<49:05, 420.82s/it]
 25%|
| 2/8 [34:32<1:18:59, 789.98s/it]
 38%|
| 3/8 [48:19<1:06:45, 801.14s/it]
 50%|
| 4/8 [7:51:46<9:05:31, 8182.96s/it]
 62%|
| 5/8 [8:12:34<5:05:07, 6102.54s/it]
 75%|
| 6/8 [9:30:42<3:09:16, 5678.14s/it]
 88%|
| 7/8 [10:02:05<1:15:39, 4539.48s/it]
 86%|
| 6/7 [15:37:46<3:58:32, 14312.35s/it]
  0%|
| 0/8 [00:00<?, ?it/s]
 12%|
| 1/8 [13:52<1:37:04, 832.07s/it]
 25%|
| 2/8 [34:25<1:35:15, 952.54s/it]
 38%|
| 3/8 [1:06:05<1:43:02, 1236.59s/it]
 50%|
| 4/8 [1:40:19<1:38:48, 1482.01s/it]
 62%|
| 5/8 [2:21:34<1:28:59, 1779.94s/it]
```

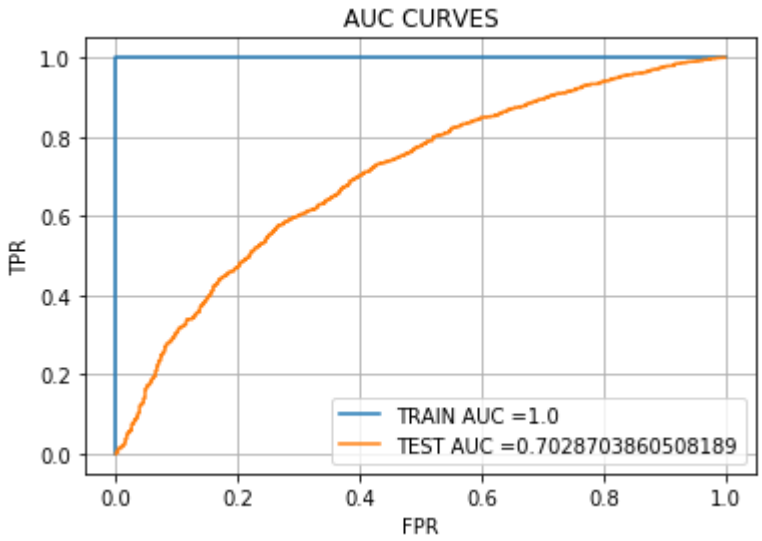
```
 75%|
| 6/8 [3:18:16<1:15:32, 2266.41s/it]
 88%|
          | 7/8 [4:38:56<50:38, 3038.66s/it]
100%|
         | 7/7 [21:35:44<00:00, 16462.16s/it]
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators in the range 2 to 10 and 50 to 1000 respectively and we choose our best max depth and number of estimators to be 2 and 50 respectively.

In [103]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
from sklearn.ensemble import GradientBoostingClassifier
GBDT=GradientBoostingClassifier(max_depth=2,n_estimators=50,loss='deviance',learning_ra
te=0.1)
calib_cv=CalibratedClassifierCV(base_estimator=GBDT)
calib_cv.fit(gbdt_train_tfidf,gbdt_ytrain)

y_train_pred=calib_cv.predict_proba(gbdt_train_tfidf)[:,1]
y_test_pred=calib_cv.predict_proba(gbdt_test_tfidf)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(gbdt_ytrain,y_tr_pred)
test_fpr,test_tpr,te_threshold=roc_curve(gbdt_ytest,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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=2 and number of estimators=50 we got train auc of 100% which is perfect almost and test auc of 70.28%. And our model is overfitting as the gap between train and test auc scores is a bit high.

In [104]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(gbdt_ytrain,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(gbdt_ytest,predict_with_best_t(y_test_pred,best_t)))
```

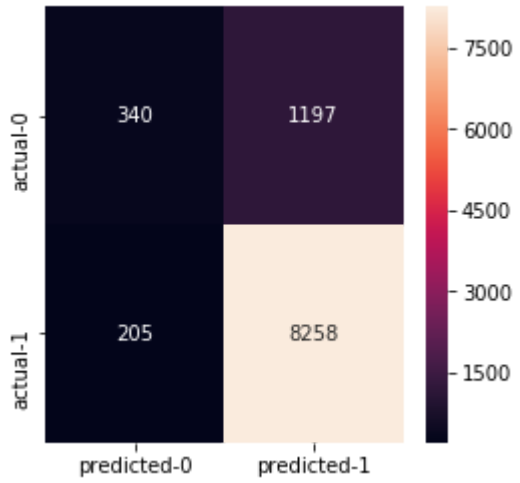
```
====================================================================================================
the maximum value of tpr*(1-fpr) 1.0 for threshold 0.667
TRAIN CONFUSION MATRIX
[[ 340 1197]
 [ 205 8258]]
test confusion matrix
[[ 115  654]
 [ 173 4058]]
```

In [105]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[340,1197],[205,8258]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[105]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24b394da1d0>
```



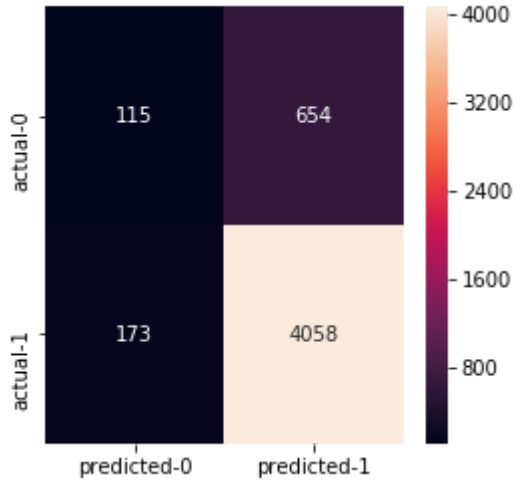
OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for train data set is good and fnr is a bit high.

In [106]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[115,654],[173,4058]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[106]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24b39465f98>
```



OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for test data set is good and fnr is a bit high.

2.5.3 Applying XGBOOST on AVG W2V, SET 3

In [0]:

```
# Please write all the code with proper documentation
```

In [90]:

```
#creating final data matrix
from scipy.sparse import hstack
final_train_avgw2v=hstack((final_train[0:10000],price_train_norm[0:10000],train_quantit
y_norm[0:10000],projects_train_norm[0:10000],avg_w2v_train,title_train_avgw2v)).tocsr()
final_cv_avgw2v=hstack((final_cv[0:5000],price_cv_norm[0:5000],cv_quantity_norm[0:5000
],projects_cv_norm[0:5000],avg_w2v_cv,title_cv_avgw2v)).tocsr()
final_test_avgw2v=hstack((final_test[0:5000],price_test_norm[0:5000],test_quantity_norm
[0:5000],projects_test_norm[0:5000],avg_w2v_test,title_test_avgw2v)).tocsr()
print(final_train_avgw2v.shape,y_train[0:1000].shape)
print(final_cv_avgw2v.shape,y_cv[0:5000].shape)
print(final_test_avgw2v.shape,y_test[0:5000].shape)
```

```
(10000, 613) (1000,)
(5000, 613) (5000,)
(5000, 613) (5000,)
```

In [91]:

```
y_train_avgw2v=y_train[0:10000]
y_cv_avgw2v=y_cv[0:5000]
y_test_avgw2v=y_test[0:5000]
```

In [92]:

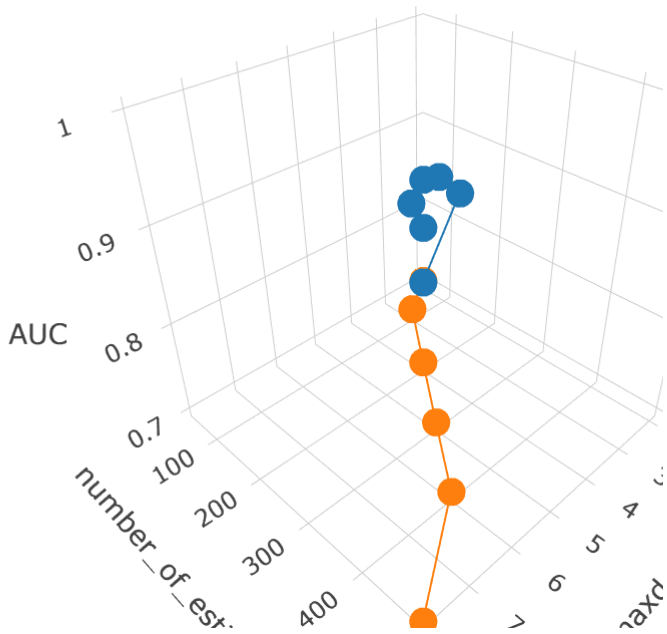
```
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import GradientBoostingClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8]
number_of_estimators=[50,100,200,300,400,500]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        GBDT= GradientBoostingClassifier(n_estimators=i,max_depth=j,loss='deviance',lea
rning_rate=0.1)
        calib_cv=CalibratedClassifierCV(base_estimator=GBDT)
        calib_cv.fit(final_train_avgw2v,y_train_avgw2v)

        y_tr_pred=calib_cv.predict_proba(final_train_avgw2v)[:,1]
        y_cv_pred=calib_cv.predict_proba(final_cv_avgw2v)[:,1]

        train_auc.append(roc_auc_score(y_train_avgw2v,y_tr_pred))
        cv_auc.append(roc_auc_score(y_cv_avgw2v,y_cv_pred))

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()
```

```
C:\Users\HP\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\en
semble\weight_boosting.py:29: DeprecationWarning:

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

  0%|
| 0/6 [00:00<?, ?it/s]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [01:39<08:17, 99.45s/it]
 33%|
| 2/6 [04:08<07:37, 114.45s/it]
 50%|
| 3/6 [07:28<07:00, 140.08s/it]
 67%|
| 4/6 [11:44<05:49, 174.66s/it]
 83%|
|               | 5/6 [17:02<03:37, 217.65s/it]
 17%|
| 1/6 [24:49<2:04:06, 1489.28s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [03:12<16:02, 192.48s/it]
 33%|
| 2/6 [08:03<14:48, 222.06s/it]
 50%|
| 3/6 [14:37<13:40, 273.60s/it]
 67%|
| 4/6 [23:00<11:25, 342.53s/it]
 83%|
|               | 5/6 [33:25<07:07, 427.32s/it]
 33%|
| 2/6 [1:24:08<2:20:40, 2110.21s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [06:23<31:57, 383.51s/it]
 33%|
| 2/6 [16:02<29:28, 442.06s/it]
 50%|
| 3/6 [1:36:01<1:27:27, 1749.10s/it]
 67%|
| 4/6 [1:52:42<50:49, 1524.80s/it]
 83%|
| 5/6 [2:13:23<23:59, 1439.71s/it]
 50%|
| 3/6 [4:07:31<3:40:54, 4418.03s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [09:36<48:00, 576.18s/it]
 33%|
| 2/6 [24:16<44:29, 667.39s/it]
 50%|
| 3/6 [44:58<41:59, 839.85s/it]
 67%|
| 4/6 [1:10:03<34:38, 1039.23s/it]
 83%|
```

```
| 5/6 [1:40:50<21:21, 1281.80s/it]
 67%|
| 4/6 [6:32:31<3:10:05, 5702.80s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [51:17<4:16:26, 3077.26s/it]
 33%|
| 2/6 [8:47:33<11:55:07, 10726.88s/it]
 50%|
| 3/6 [9:59:27<7:20:09, 8803.18s/it]
 67%|
| 4/6 [10:32:47<3:45:24, 6762.22s/it]
 83%|
| 5/6 [11:43:54<1:40:13, 6013.65s/it]
 83%|
| 5/6 [19:15:00<4:55:16, 17716.58s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [15:59<1:19:55, 959.01s/it]
 33%|
| 2/6 [40:10<1:13:47, 1106.83s/it]
 50%|
| 3/6 [1:12:56<1:08:13, 1364.49s/it]
 67%|
| 4/6 [1:54:34<56:49, 1704.50s/it]
 83%|
| 5/6 [3:11:22<42:55, 2575.68s/it]
100%|
| 6/6 [23:58:19<00:00, 17501.34s/it]
```

In [93]:

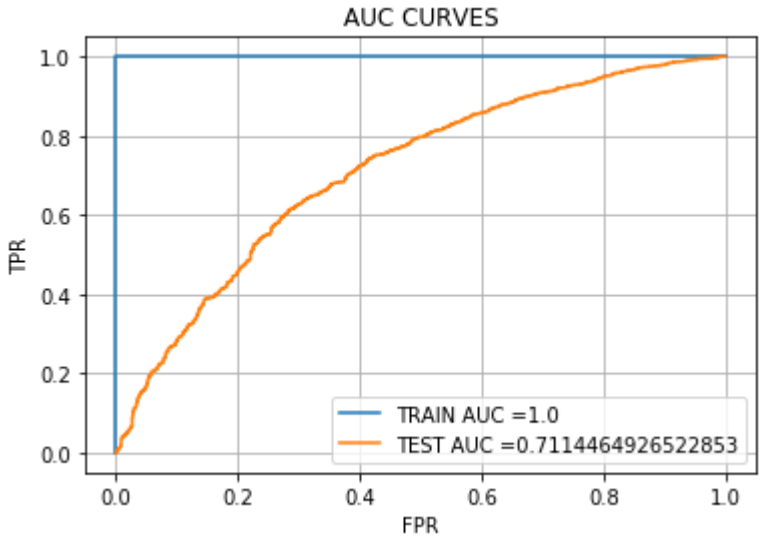
```
layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators in the range 2 to 10 and 50 to 1000 respectively and we choose our best max depth and number of estimators to be 2 and 50 respectively.

In [94]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
from sklearn.ensemble import GradientBoostingClassifier
GBDT=GradientBoostingClassifier(max_depth=2,n_estimators=50,loss='deviance',learning_ra
te=0.1)
calib_cv=CalibratedClassifierCV(base_estimator=GBDT)
calib_cv.fit(final_train_avgw2v,y_train_avgw2v)

y_train_pred=calib_cv.predict_proba(final_train_avgw2v)[:,1]
y_test_pred=calib_cv.predict_proba(final_test_avgw2v)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(y_train_avgw2v,y_tr_pred)
test_fpr,test_tpr,te_threshold=roc_curve(y_test_avgw2v,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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=2 and number of estimators=50 we got train auc of 100% which is perfect almost and test auc of 71.11%. And our model is overfitting as the gap between train and test auc scores is a bit high.

In [97]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(y_train_avgw2v,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(y_test_avgw2v,predict_with_best_t(y_test_pred,best_t)))
```

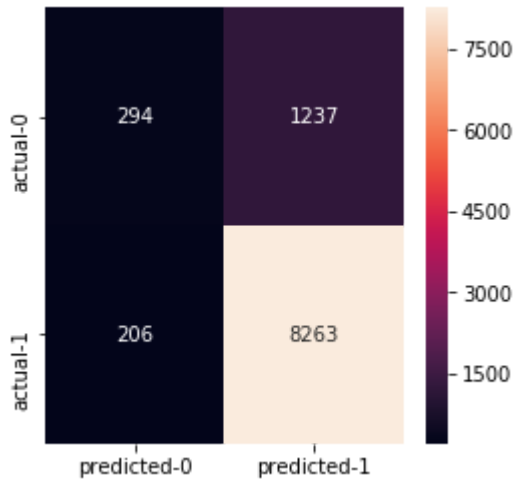
```
====================================================================================================
the maximum value of tpr*(1-fpr) 1.0 for threshold 0.695
TRAIN CONFUSION MATRIX
[[ 294 1237]
 [ 206 8263]]
test confusion matrix
[[ 145  638]
 [ 195 4022]]
```

In [98]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[294,1237],[206,8263]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[98]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a347bb6dd8>
```



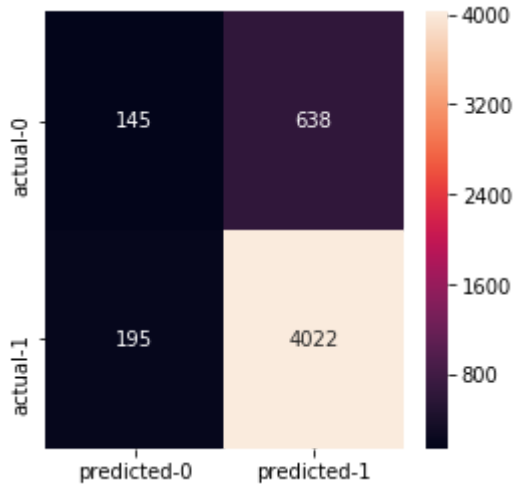
OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for train data set is good and fnr is a bit high.

In [99]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[145,638],[195,4022]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[99]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a347ad6a20>
```



OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for test data set is good and fnr is a bit high.

2.5.4 Applying XGBOOST on TFIDF W2V, SET 4

In [0]:

```
# Please write all the code with proper documentation
```

In [110]:

```
#creating final data matrix
from scipy.sparse import hstack
final_train_tfidfw2v=hstack((final_train[0:10000],price_train_norm[0:10000],train_quant
ity_norm[0:10000],projects_train_norm[0:10000],tfidf_w2v_train,title_train_tfidfw2v)).t
ocsr()
final_cv_tfidfw2v=hstack((final_cv[0:5000],price_cv_norm[0:5000],cv_quantity_norm[0:500
0],projects_cv_norm[0:5000],tfidf_w2v_cv,title_cv_tfidfw2v)).tocsr()
final_test_tfidfw2v=hstack((final_test[0:5000],price_test_norm[0:5000],test_quantity_no
rm[0:5000],projects_test_norm[0:5000],tfidf_w2v_test,title_test_tfidfw2v)).tocsr()
print(final_train_tfidfw2v.shape,y_train[0:1000].shape)
print(final_cv_tfidfw2v.shape,y_cv[0:5000].shape)
print(final_test_tfidfw2v.shape,y_test[0:5000].shape)
```

```
(10000, 613) (1000,)
(5000, 613) (5000,)
(5000, 613) (5000,)
```

In [111]:

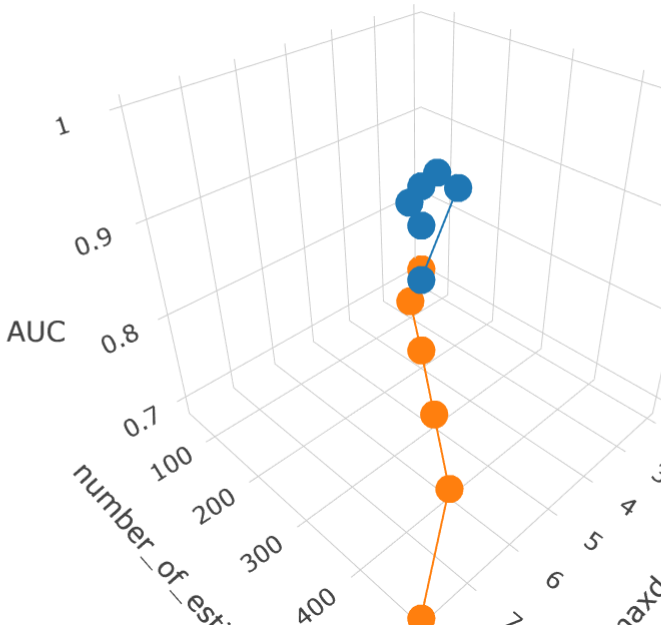
```
y_train_tfidfw2v=y_train[0:10000]
y_cv_tfidfw2v=y_cv[0:5000]
y_test_tfidfw2v=y_test[0:5000]
```

In [112]:

```
#plotting error plots
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import GradientBoostingClassifier
import plotly.graph_objs as go
from sklearn.calibration import CalibratedClassifierCV


train_auc=[]
cv_auc=[]
maxdepth=[2,3,4,5,6,8]
number_of_estimators=[50,100,200,300,400,500]
for i in tqdm(number_of_estimators):
    for j in tqdm(maxdepth):
        GBDT= GradientBoostingClassifier(n_estimators=i,max_depth=j,loss='deviance',lea
rning_rate=0.1)
        calib_cv=CalibratedClassifierCV(base_estimator=GBDT)
        calib_cv.fit(final_train_tfidfw2v,y_train_tfidfw2v)

        y_tr_pred=calib_cv.predict_proba(final_train_tfidfw2v)[:,1]
        y_cv_pred=calib_cv.predict_proba(final_cv_tfidfw2v)[:,1]

        train_auc.append(roc_auc_score(y_train_tfidfw2v,y_tr_pred))
        cv_auc.append(roc_auc_score(y_cv_tfidfw2v,y_cv_pred))

trace1=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=train_auc,name='TRAIN AUC')
trace2=go.Scatter3d(x=maxdepth,y=number_of_estimators,z=cv_auc,name='CV AUC')
data=[trace1,trace2]
enable_plotly_in_cell()

layout=go.Layout(scene = dict(
        xaxis = dict(title='maxdepth'),
        yaxis = dict(title='number_of_estimators'),
        zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

```
  0%|
| 0/6 [00:00<?, ?it/s]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [01:37<08:09, 97.81s/it]
 33%|
| 2/6 [04:05<07:31, 112.77s/it]
 50%|
| 3/6 [07:49<07:18, 146.19s/it]
 67%|
| 4/6 [12:31<06:13, 186.92s/it]
 83%|
|               | 5/6 [18:04<03:50, 230.85s/it]
 17%|
| 1/6 [26:16<2:11:22, 1576.51s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [03:13<16:06, 193.39s/it]
 33%|
| 2/6 [08:06<14:53, 223.42s/it]
 50%|
| 3/6 [14:45<13:47, 275.83s/it]
 67%|
| 4/6 [23:19<11:34, 347.40s/it]
 83%|
|               | 5/6 [34:01<07:15, 435.80s/it]
 33%|
| 2/6 [1:16:35<2:13:57, 2009.26s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [06:26<32:10, 386.14s/it]
 33%|
| 2/6 [16:11<29:43, 445.86s/it]
 50%|
| 3/6 [29:24<27:30, 550.02s/it]
 67%|
| 4/6 [46:24<23:02, 691.19s/it]
 83%|
| 5/6 [1:07:46<14:28, 868.32s/it]
 50%|
| 3/6 [3:12:23<2:54:32, 3490.88s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [09:36<48:01, 576.24s/it]
 33%|
| 2/6 [24:13<44:25, 666.42s/it]
 50%|
| 3/6 [44:01<41:08, 822.91s/it]
 67%|
| 4/6 [1:09:31<34:30, 1035.27s/it]
 83%|
| 5/6 [1:44:19<22:30, 1350.86s/it]
 67%|
| 4/6 [6:03:37<3:04:11, 5525.90s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
```

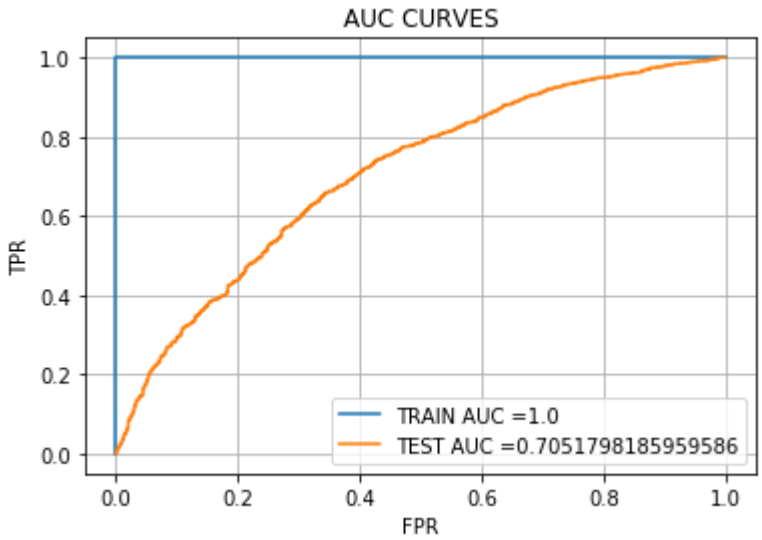
```
| 1/6 [12:46<1:03:51, 766.37s/it]
 33%|
| 2/6 [10:34:59<13:02:25, 11736.27s/it]
 50%|
| 3/6 [11:16:38<7:28:15, 8965.08s/it]
 67%|
| 4/6 [12:03:20<3:57:12, 7116.44s/it]
 83%|
| 5/6 [12:46:07<1:35:51, 5751.36s/it]
 83%|
| 5/6 [19:50:38<5:12:34, 18754.25s/it]
  0%|
| 0/6 [00:00<?, ?it/s]
 17%|
| 1/6 [15:59<1:19:58, 959.79s/it]
 33%|
| 2/6 [40:18<1:13:57, 1109.44s/it]
 50%|
| 3/6 [1:13:20<1:08:33, 1371.22s/it]
 67%|
| 4/6 [2:12:52<1:07:43, 2031.55s/it]
 83%|
| 5/6 [3:34:06<48:04, 2884.24s/it]
100%|
| 6/6 [24:26:08<00:00, 18086.95s/it]
```



OBSERVATIONS: We have plotted for different values of max depth and number of estimators in the range 2 to 10 and 50 to 1000 respectively and we choose our best max depth and number of estimators to be 2 and 50 respectively.

In [115]:

```
#plotting roc curve
from sklearn.metrics import roc_curve,auc
from sklearn.ensemble import GradientBoostingClassifier
GBDT=GradientBoostingClassifier(max_depth=2,n_estimators=50,loss='deviance',learning_ra
te=0.1)
calib_cv=CalibratedClassifierCV(base_estimator=GBDT)
calib_cv.fit(final_train_tfidfw2v,y_train_tfidfw2v)

y_train_pred=calib_cv.predict_proba(final_train_tfidfw2v)[:,1]
y_test_pred=calib_cv.predict_proba(final_test_tfidfw2v)[:,1]


train_fpr,train_tpr,tr_threshold=roc_curve(y_train_tfidfw2v,y_tr_pred)
test_fpr,test_tpr,te_threshold=roc_curve(y_test_tfidfw2v,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.title('AUC CURVES')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```



OBSERVATIONS: For max depth=2 and number of estimators=50 we got train auc of 100% which is perfect almost and test auc of 70.51%. And our model is overfitting as the gap between train and test auc scores is a bit high.

In [116]:

```
#printing confusion matrix
print('='*100)
from sklearn.metrics import confusion_matrix
best_t=find_best_threshold(tr_threshold,train_fpr,train_tpr)
print('TRAIN CONFUSION MATRIX')
print(confusion_matrix(y_train_tfidfw2v,predict_with_best_t(y_train_pred,best_t)))
print('test confusion matrix')
print(confusion_matrix(y_test_tfidfw2v,predict_with_best_t(y_test_pred,best_t)))
```

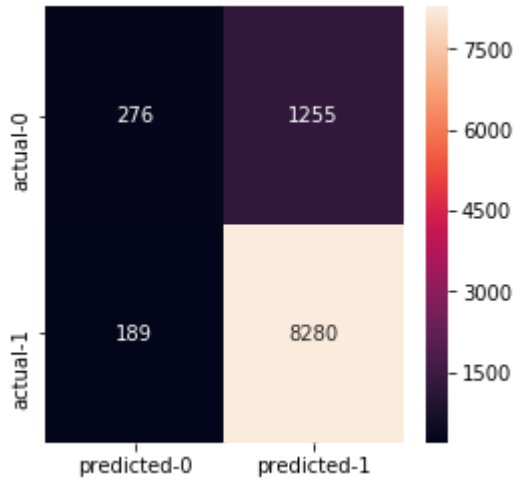
```
====================================================================================================
the maximum value of tpr*(1-fpr) 1.0 for threshold 0.696
TRAIN CONFUSION MATRIX
[[ 276 1255]
 [ 189 8280]]
test confusion matrix
[[ 133  650]
 [ 177 4040]]
```

In [117]:

```
#printing heatmap for train confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[276,1255],[189,8280]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[117]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a3569656a0>
```



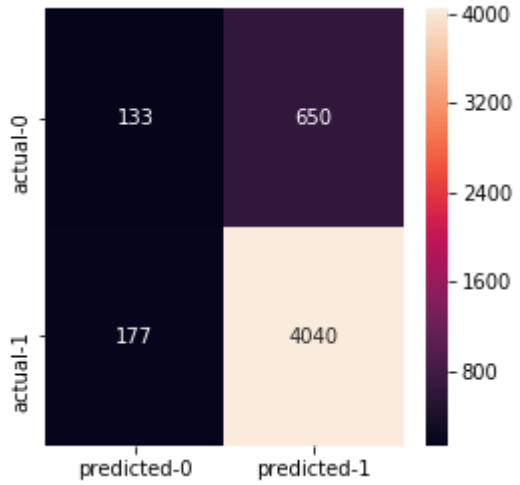
OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for train data set is good and fnr is a bit high.

In [119]:

```
#printing heatmap for test confusion matrix
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt

array=[[133,650],[177,4040]]

train=pd.DataFrame(array,columns=['predicted-0','predicted-1'],index=['actual-0','actua
l-1'])
plt.figure(figsize=(4,4))
sn.heatmap(train,annot=True,fmt='d')
```

Out[119]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a356eade48>
```



OBSERVATIONS: For max depth =2 and number of estimators=50 the tpr value for test data set is good and fnr is a bit high.

3. Conclusion

In [0]:

```
# Please compare all your models using Prettytable library
```

In [129]:

```
# Please compare all your models using Prettytable library
from prettytable import PrettyTable
x=PrettyTable(['MODEL','vectorizer','max depth','number of estimators','train_auc','tes
t_auc'])
x.add_row(['RANDOM FOREST',"bag of words",3,100,0.898042,0.707881])
x.add_row(['RANDOM FOREST',"TFIDF",3,150,0.932931,0.711555])
x.add_row(['RANDOM FOREST',"avgw2v",2,50,0.997133,0.669944])
x.add_row(['RANDOM FOREST',"TFIDFW2V",2,50,0.992504,0.667952])
x.add_row(['GB RANDOM FOREST',"bag of words",2,50,0.999968,0.708765])
x.add_row(['GB RANDOM FOREST',"TFIDF",2,50,100,0.702870])
x.add_row(['GB RANDOM FOREST',"avgw2v",2,50,100,0.711446])
x.add_row(['GB RANDOM FOREST',"TFIDFW2V",2,50,100,0.705179])



print(x.get_string(start=0,end=9))
```

```
+------------------+--------------+-----------+----------------------+----
-------+----------+
|      MODEL       |  vectorizer  | max depth | number of estimators | tra
in_auc | test_auc |
+------------------+--------------+-----------+----------------------+----
-------+----------+
|  RANDOM FOREST   | bag of words |     3     |         100          |  0.
898042 | 0.707881 |
|  RANDOM FOREST   |    TFIDF     |     3     |         150          |  0.
932931 | 0.711555 |
|  RANDOM FOREST   |    avgw2v    |     2     |          50          |  0.
997133 | 0.669944 |
|  RANDOM FOREST   |   TFIDFW2V   |     2     |          50          |  0.
992504 | 0.667952 |
| GB RANDOM FOREST | bag of words |     2     |          50          |  0.
999968 | 0.708765 |
| GB RANDOM FOREST |    TFIDF     |     2     |          50          |
100    | 0.70287  |
| GB RANDOM FOREST |    avgw2v    |     2     |          50          |
100    | 0.711446 |
| GB RANDOM FOREST |   TFIDFW2V   |     2     |          50          |
100    | 0.705179 |
+------------------+--------------+-----------+----------------------+----
-------+----------+
```

OBSERVATIONS: HERE We plotted for 2 different models random forest and gradient boosted random forest. For almost all the models the hyperparameters are same. Train auc scores for all the models are high like almost 100% and test auc scores are low compared to that. And our models are overfitting as we took very less data. Compared to random forest models, gbdt models have higher train and test auc scores.