DonorsChoose_CaseStudy

December 5, 2018

1 DonorsChoose Application Screening

Data Source: https://www.kaggle.com/c/donorschoose-application-screening

2 Introduction

About DonorsChoose: Founded in 2000 by a high school teacher in the Bronx, DonorsChoose.org empowers public school teachers from across the country to request much-needed materials and experiences for their students. At any given time, there are thousands of classroom requests that can be brought to life with a gift of any amount.

Their Mission: They make it easy for anyone to help a classroom in need, moving them closer to a nation where students in every community have the tools and experiences they need for a great education.

Objective: 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.

3 About Data

The dataset contains information from teachers' project applications to DonorsChoose.org including teacher attributes, school attributes, and the project proposals including application essays.

Files: Data is provided in Two Files:

```
<li>train.csv - the training set</li>
<li>resources.csv - resources requested by each proposal; joins with train.csv on id</li>
```

Data Fields:

Train Data Fields id - unique id of the project application
teacher_id - id of the teacher submitting the application
teacher_prefix - title of the teacher's name (Ms., Mr., etc.)
school_state - US state of the teacher's school
project_submitted_datetime - application submission timestamp
project_grade_category - school grade levels (PreK-2, 3-5, 6-8, and 9-12)
project_subject_categories - category of the project (e.g., "Music & The Arts")
project_subject_subcategories - sub-category of the project (e.g., "Visual Arts")
project_title - title of the project
project_essay_1 - first essay*
project_essay_2 - second essay*
project_essay_3 - third essay*
project_essay_4 - fourth essay*
project_resource_summary - summary of the resources needed for the project
teacher_number_of_previously_posted_projects - number of previously posted applications by the submitting teacher
project_is_approved - whether DonorsChoose proposal was accepted (0="rejected", 1="accepted"); train.csv only
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_4: "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?"

Resources Data Fields id - unique id of the project application; joins with train.csv on id
description - description of the resource requested
quantity - quantity of resource requested
price - price of resource requested

3.1 Loading Data

```
In [1]: %matplotlib inline
        from datetime import datetime

        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 warnings
warnings.filterwarnings("ignore")

import pickle
import re

import string
from nltk.corpus import stopwords

from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

from sklearn.preprocessing import LabelEncoder
from scipy.sparse import hstack
from sklearn.preprocessing import scale,StandardScaler

from sklearn.grid_search import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
#from xgboost import XGBClassifier
from sklearn import tree
from sklearn.decomposition import TruncatedSVD
import gensim
from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import lightgbm as lgb
from sklearn.metrics import roc_auc_score
from xgboost import XGBClassifier
```

In [2]:
```
train_df = pd.read_csv('train.csv')
```

In [3]:
```
train_df.head()
```

Out[3]:

	id	teacher_id	teacher_prefix	school_state	\
0	p036502	484aaf11257089a66cfedc9461c6bd0a	Ms.	NV	
1	p039565	df72a3ba8089423fa8a94be88060f6ed	Mrs.	GA	
2	p233823	a9b876a9252e08a55e3d894150f75ba3	Ms.	UT	
3	p185307	525fdbb6ec7f538a48beebaa0a51b24f	Mr.	NC	

```
4  p013780  a63b5547a7239eae4c1872670848e61a           Mr.           CA

  project_submitted_datetime project_grade_category  \
0        2016-11-18 14:45:59          Grades PreK-2
1        2017-04-26 15:57:28             Grades 3-5
2        2017-01-01 22:57:44             Grades 3-5
3        2016-08-12 15:42:11             Grades 3-5
4        2016-08-06 09:09:11             Grades 6-8

           project_subject_categories  \
0                Literacy & Language
1   Music & The Arts, Health & Sports
2  Math & Science, Literacy & Language
3                  Health & Sports
4                  Health & Sports

           project_subject_subcategories  \
0                              Literacy
1           Performing Arts, Team Sports
2  Applied Sciences, Literature & Writing
3                  Health & Wellness
4                  Health & Wellness

                                    project_title  \
0                        Super Sight Word Centers
1                          Keep Calm and Dance On
2                           Lets 3Doodle to Learn
3  \"Kid Inspired\" Equipment to Increase Activit...
4   We need clean water for our culinary arts class!

                                  project_essay_1  \
0  Most of my kindergarten students come from low...
1  Our elementary school is a culturally rich sch...
2  Hello;\r\nMy name is Mrs. Brotherton. I teach ...
3  My students are the greatest students but are ...
4  My students are athletes and students who are ...

                                  project_essay_2 project_essay_3  \
0  I currently have a differentiated sight word c...             NaN
1  We strive to provide our diverse population of...             NaN
2  We are looking to add some 3Doodler to our cla...             NaN
3  The student's project which is totally \"kid-i...             NaN
4  For some reason in our kitchen the water comes...             NaN

  project_essay_4                         project_resource_summary  \
0             NaN  My students need 6 Ipod Nano's to create and d...
1             NaN  My students need matching shirts to wear for d...
2             NaN  My students need the 3doodler. We are an SEM s...
```

```
3             NaN  My students need balls and other activity equi...
4             NaN  My students need a water filtration system for...

   teacher_number_of_previously_posted_projects  project_is_approved
0                                            26                   1
1                                             1                   0
2                                             5                   1
3                                            16                   0
4                                            42                   1
```

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

```
Out[4]: (182080, 16)
```

```
In [5]: resource_df = pd.read_csv('resources.csv')
```

```
In [6]: resource_df.shape
```

```
Out[6]: (1541272, 4)
```

3.1.1 Merging Train Data with Resource

```
In [7]: train_res_df = pd.merge(train_df,resource_df,on='id',how='left')
```

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

```
Out[8]: (1081830, 19)
```

Observation: There are around **10lakh** data points with **19 features** in the training data

4 Exploratory Data Analysis(EDA)

```
In [9]: train_res_df.head()
```

```
Out[9]:         id                         teacher_id teacher_prefix school_state  \
0  p036502  484aaf11257089a66cfedc9461c6bd0a            Ms.           NV
1  p036502  484aaf11257089a66cfedc9461c6bd0a            Ms.           NV
2  p039565  df72a3ba8089423fa8a94be88060f6ed           Mrs.           GA
3  p233823  a9b876a9252e08a55e3d894150f75ba3            Ms.           UT
4  p185307  525fdbb6ec7f538a48beebaa0a51b24f            Mr.           NC

  project_submitted_datetime project_grade_category  \
0        2016-11-18 14:45:59          Grades PreK-2
1        2016-11-18 14:45:59          Grades PreK-2
2        2017-04-26 15:57:28             Grades 3-5
3        2017-01-01 22:57:44             Grades 3-5
4        2016-08-12 15:42:11             Grades 3-5

              project_subject_categories  \
```

```
0                  Literacy & Language
1                  Literacy & Language
2    Music & The Arts, Health & Sports
3  Math & Science, Literacy & Language
4                      Health & Sports

           project_subject_subcategories  \
0                               Literacy
1                               Literacy
2            Performing Arts, Team Sports
3  Applied Sciences, Literature & Writing
4                        Health & Wellness

                                      project_title  \
0                           Super Sight Word Centers
1                           Super Sight Word Centers
2                             Keep Calm and Dance On
3                              Lets 3Doodle to Learn
4  \"Kid Inspired\" Equipment to Increase Activit...

                                    project_essay_1  \
0  Most of my kindergarten students come from low...
1  Most of my kindergarten students come from low...
2  Our elementary school is a culturally rich sch...
3  Hello;\r\nMy name is Mrs. Brotherton. I teach ...
4  My students are the greatest students but are ...

                                    project_essay_2 project_essay_3  \
0  I currently have a differentiated sight word c...             NaN
1  I currently have a differentiated sight word c...             NaN
2  We strive to provide our diverse population of...             NaN
3  We are looking to add some 3Doodler to our cla...             NaN
4  The student's project which is totally \"kid-i...             NaN

  project_essay_4                         project_resource_summary  \
0             NaN  My students need 6 Ipod Nano's to create and d...
1             NaN  My students need 6 Ipod Nano's to create and d...
2             NaN  My students need matching shirts to wear for d...
3             NaN  My students need the 3doodler. We are an SEM s...
4             NaN  My students need balls and other activity equi...

   teacher_number_of_previously_posted_projects  project_is_approved  \
0                                            26                   1
1                                            26                   1
2                                             1                   0
3                                             5                   1
4                                            16                   0
```

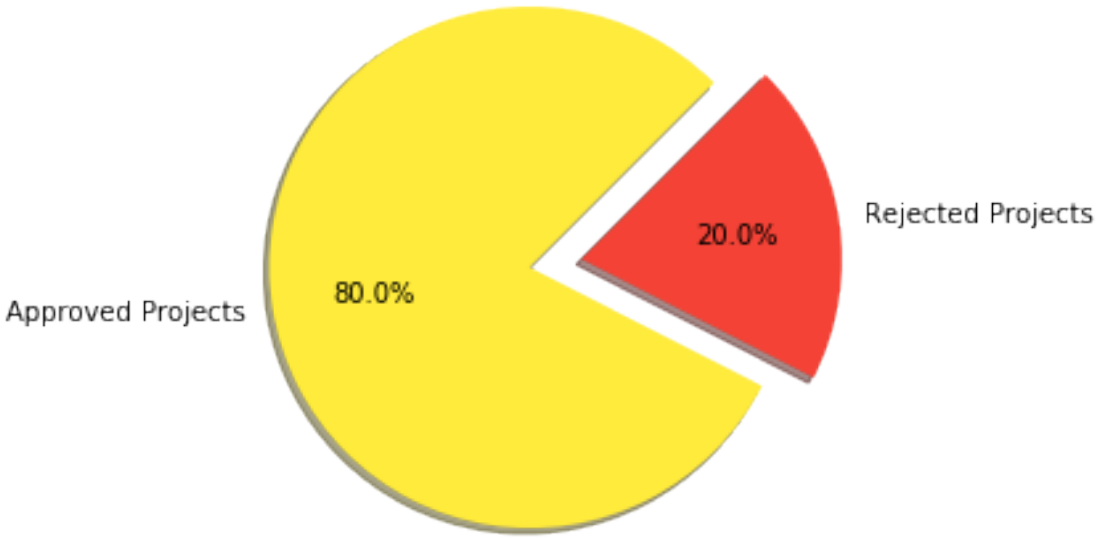
```
                                          description  quantity   price
0  Apple - iPod nano 16GB MP3 Player (8th Genera...         3  149.99
1  Apple - iPod nano 16GB MP3 Player (8th Genera...         3  149.99
2  Reebok Girls' Fashion Dance Graphic T-Shirt - ...        20   20.00
3                    3doodler Start Full Edu Bundle         1  469.99
4                  BALL PG 4'' POLY SET OF 6 COLORS         1   18.95
```

4.1 Percentage of Projects Approved/Rejected

```
In [10]: approved_percent = train_res_df[train_res_df['project_is_approved']==1].shape[0]/trai
         print("Percentage of Approved Projects: ",approved_percent)
         print("Percentage of Rejected Projects: ",100-approved_percent)
```

```
Percentage of Approved Projects:  79.23777303273157
Percentage of Rejected Projects:  20.762226967268433
```

```
In [11]: import matplotlib.pyplot as plt

         # Data to plot
         labels = 'Approved Projects', 'Rejected Projects'
         slices_hours = [9.6, 2.4]
         colors = ['#FFEB3B','#f44336']
         explode = (0.2, 0)  # explode 1st slice

         # Plot
         plt.pie(slices_hours, explode=explode, labels=labels, colors=colors,
                 autopct='%1.1f%%', shadow=True, startangle=45)

         plt.axis('equal')
         plt.show()
```



Observation: Around **80%** of submitted projects were **Approved**. Its an **"Imbalanced"** DataSet

4.2 Data Cleaning

4.2.1 Missing Data

```
In [12]: # sum of rows containing null values
         total_NaN_rows = train_res_df[['project_essay_3','project_essay_4','description']].isn
         percent = (total_NaN_rows/train_res_df[['project_essay_3','project_essay_4','descripti
         missing_train_data  = pd.concat([total_NaN_rows, percent], axis=1, keys=['Total_NaN_ro
         missing_train_data.head()
```

Out[12]:

	Total_NaN_rows	Percent
project_essay_3	1043673	96.472921
project_essay_4	1043673	96.472921
description	192	0.017748

Observation: Over **96%** of the rows **dont have Data for Columns project_essay_3 and project_essay_4** so lets ignore these Columns

Removing the above columns

```
In [13]: train_df = train_df.drop(['project_essay_3','project_essay_4'],axis=1)


         train_res_df = train_res_df.drop(['project_essay_3','project_essay_4'],axis=1)
```

4.2.2 Removing Duplicate Rows if Any

```
In [14]: train_res_df.shape
```

Out[14]: (1081830, 17)

```
In [15]: train_res_df.duplicated().sum()
```

Out[15]: 8576

Observation: There are 8576 Duplicated rows in our training data

```
In [16]: train_res_df.loc[train_res_df.duplicated(),:][0:5]
```

Out[16]:

	id	teacher_id	teacher_prefix	school_state	\
148	p000139	f68fedcb0852d8a6ce88f7b4139b9227	Mr.	TX	
153	p000139	f68fedcb0852d8a6ce88f7b4139b9227	Mr.	TX	
394	p160114	d523f258b55bb41c3bcb6dd67cfab6c1	Ms.	OK	
469	p192882	d36c158f9f95db287dc019fe6f00cdad	Mrs.	WI	

```
740  p240306  1ee0a12eeb5da1c2fb438c682c8177dc          Mrs.          GA

    project_submitted_datetime project_grade_category  \
148        2016-08-07 20:33:16            Grades 9-12
153        2016-08-07 20:33:16            Grades 9-12
394        2016-09-22 01:14:50          Grades PreK-2
469        2016-08-20 11:27:08          Grades PreK-2
740        2017-03-02 16:57:41          Grades PreK-2

            project_subject_categories project_subject_subcategories  \
148                    Music & The Arts                         Music
153                    Music & The Arts                         Music
394   Literacy & Language, Special Needs      Literacy, Special Needs
469                     Health & Sports            Health & Wellness
740  Literacy & Language, Math & Science       Literacy, Mathematics

                   project_title  \
148               Deeds for Reeds
153               Deeds for Reeds
394       Bring Our Story to Life
469                Learn and Move
740  Create - Evaluate - Celebrate

                                      project_essay_1  \
148  As I've written here before, we work to use ba...
153  As I've written here before, we work to use ba...
394  The students at my school are multicultural, c...
469  As a teacher in a low-income/high poverty scho...
740  CREATE - EVALUATE - CELEBRATE!!!!\r\nEach chil...

                                      project_essay_2  \
148  A guitar can't sound good with broken or worn ...
153  A guitar can't sound good with broken or worn ...
394  My students struggle with writing.  Sometimes ...
469  This past year I noticed how wiggly my student...
740  Osmo states that it enables the iPad to merge ...

                             project_resource_summary  \
148  My students need access to quality replacement...
153  My students need access to quality replacement...
394  My students need a printer, ink and paper.  Th...
469  My students need Hokk chairsi, wobble chairs a...
740  My students need to be challenged using techno...

     teacher_number_of_previously_posted_projects  project_is_approved  \
148                                             3                    1
153                                             3                    1
394                                             0                    1
```

```
469                                                 0                   0
740                                                 2                   1

                                              description  quantity  price
148                              Tenor Saxophone Reeds          1  20.95
153                       Traditional Bb Clarinet Reeds         6  21.95
394  Brother LC109BK - Super High Yield - black - o...          1  32.70
469  Norwood Commercial Furniture NOR-STOOLBS-SO Pl...          1  48.95
740                                    Osmo Starter Kit         1  79.99
```

Observation: These are the sum of the duplicate rows

```
In [17]: train_res_df[(train_res_df['id']=='p160114') & (train_res_df['price']==32.7)]
```

```
Out[17]:          id                         teacher_id teacher_prefix school_state  \
393  p160114  d523f258b55bb41c3bcb6dd67cfab6c1            Ms.           OK
394  p160114  d523f258b55bb41c3bcb6dd67cfab6c1            Ms.           OK

    project_submitted_datetime project_grade_category  \
393        2016-09-22 01:14:50          Grades PreK-2
394        2016-09-22 01:14:50          Grades PreK-2

             project_subject_categories project_subject_subcategories  \
393  Literacy & Language, Special Needs       Literacy, Special Needs
394  Literacy & Language, Special Needs       Literacy, Special Needs

              project_title  \
393  Bring Our Story to Life
394  Bring Our Story to Life

                                       project_essay_1  \
393  The students at my school are multicultural, c...
394  The students at my school are multicultural, c...

                                       project_essay_2  \
393  My students struggle with writing.  Sometimes ...
394  My students struggle with writing.  Sometimes ...

                              project_resource_summary  \
393  My students need a printer, ink and paper.  Th...
394  My students need a printer, ink and paper.  Th...

     teacher_number_of_previously_posted_projects  project_is_approved  \
393                                             0                   1
394                                             0                   1

                                              description  quantity  price
393  Brother LC109BK - Super High Yield - black - o...          1   32.7
394  Brother LC109BK - Super High Yield - black - o...          1   32.7
```

As we can see The above 2 rows are Exactly the same, We are removing these kind of rows.

Removing Duplicate rows

```
In [18]: train_res_df=train_res_df.drop_duplicates(keep='first', inplace=False)
         train_res_df.shape

Out[18]: (1073254, 17)
```

4.3 Lets Understand the data

```
In [19]: train_res_df['teacher_number_of_previously_posted_projects'].describe()

Out[19]: count    1.073254e+06
         mean     1.256718e+01
         std      3.042456e+01
         min      0.000000e+00
         25%      0.000000e+00
         50%      3.000000e+00
         75%      1.000000e+01
         max      4.510000e+02
         Name: teacher_number_of_previously_posted_projects, dtype: float64
```

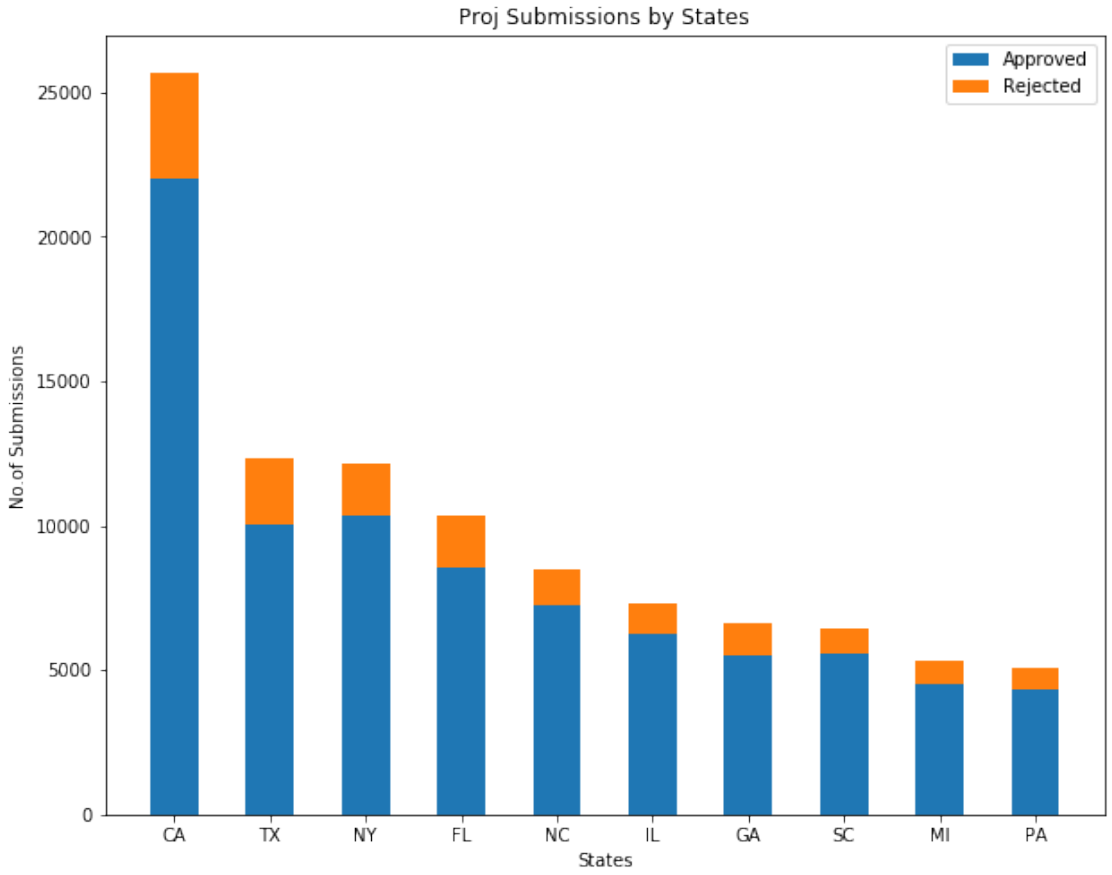
-> Observation:

```
<li>Max no.of Projects Posted by a Teacher is 45</li>
<li>75% of the Teachers have posted around 10 Projects</li>
```

4.3.1 Statewise Submissions

```
In [20]: train_df =train_df.sort_values(by=['school_state'])

In [103]: width = 0.5       # the width of the bars: can also be len(x) sequence
          N = np.arange(10)
          state_wise_count = train_df['school_state'].value_counts()
          state_wise_count_approved = []
          state_wise_count_rejected = []

          for state in state_wise_count.index:
              state_wise_count_approved.append(np.sum(train_df["project_is_approved"][train_df
              state_wise_count_rejected.append(np.sum(train_df["project_is_approved"][train_df


          plt.figure(figsize=(10,8))
          bar1 = plt.bar(N, state_wise_count_approved[0:10], width)
          bar2 = plt.bar(N, state_wise_count_rejected[0:10], width,
                         bottom=state_wise_count_approved[0:10])
```

```
plt.xticks(N,state_wise_count.index)

plt.title('Proj Submissions by States')
plt.xlabel('States')
plt.ylabel('No.of Submissions')

plt.legend((bar1[0], bar2[0]), ('Approved', 'Rejected'))

plt.show()
```

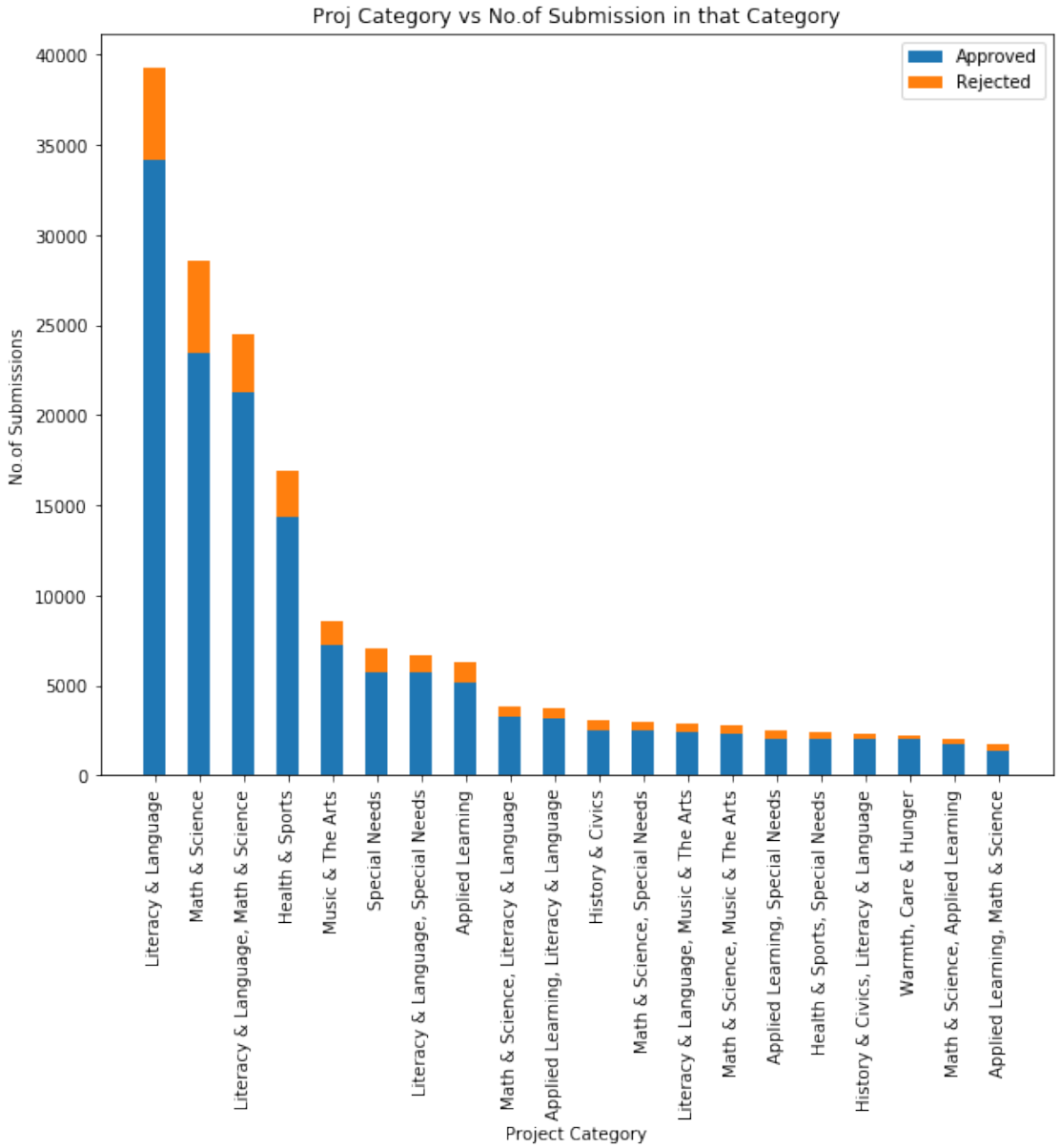


Observation: We can observe that Submissions from **California** is **More** than other States.

4.3.2 Count of Projs in each Category

```
In [110]: width = 0.5
          N = np.arange(20)
          proj_wise_count = train_df['project_subject_categories'].value_counts()
```

```
proj_wise_count_approved = []
proj_wise_count_rejected = []

for proj in proj_wise_count.index:
    proj_wise_count_approved.append(np.sum(train_df["project_is_approved"][train_df['
    proj_wise_count_rejected.append(np.sum(train_df["project_is_approved"][train_df['


plt.figure(figsize=(10,8))
bar1 = plt.bar(N, proj_wise_count_approved[0:20], width)
bar2 = plt.bar(N, proj_wise_count_rejected[0:20], width,
             bottom=proj_wise_count_approved[0:20])


plt.xticks(N,proj_wise_count.index,rotation=90)

plt.title('Proj Category vs No.of Submission in that Category')
plt.xlabel('Project Category')
plt.ylabel('No.of Submissions')

plt.legend((bar1[0], bar2[0]), ('Approved', 'Rejected'))

plt.show()
```



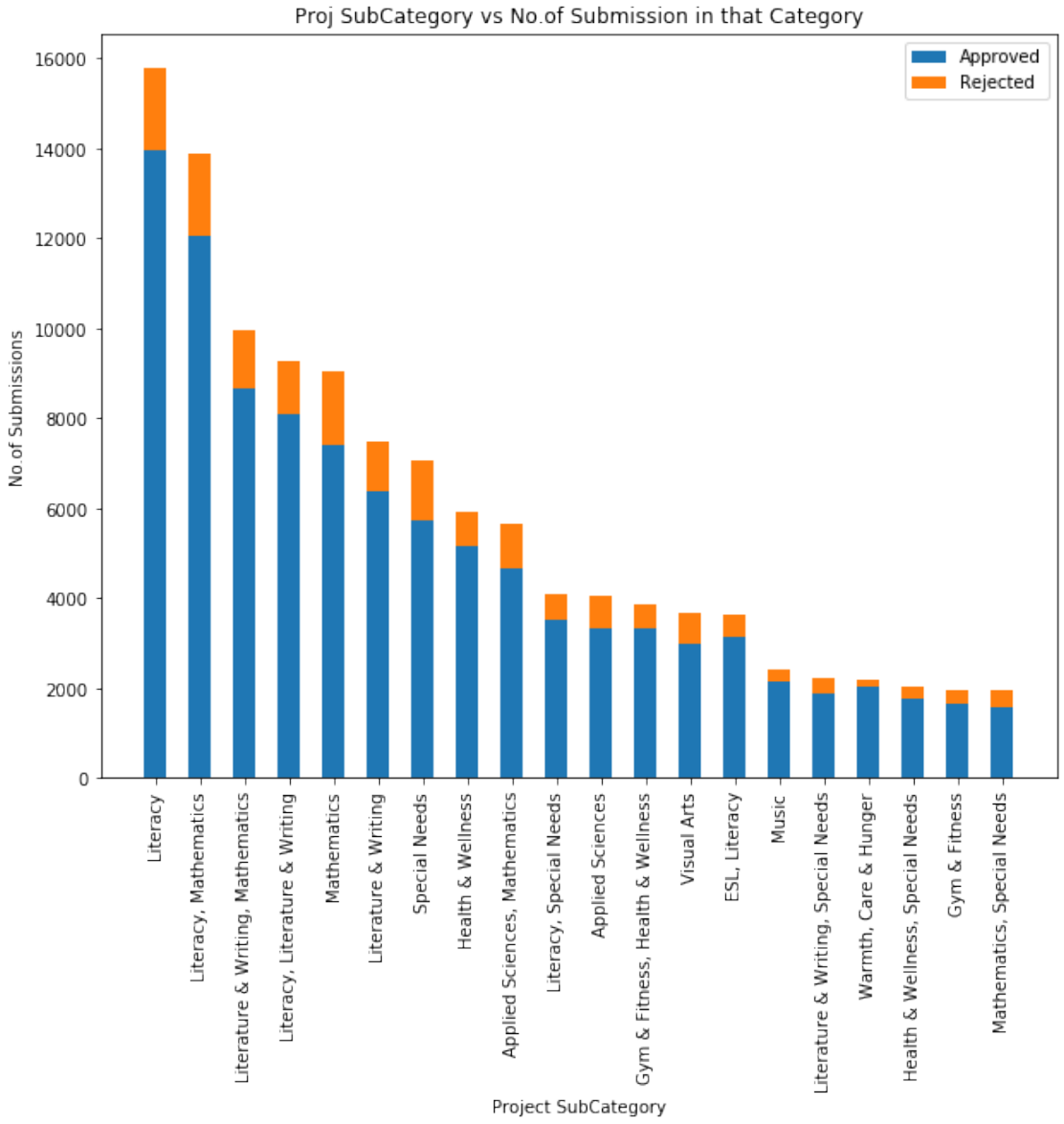

```
In [116]: print('Percentage of Literacy & Language: ',(proj_wise_count.head(1).sum()/proj_wise
          print('Percentage of Top 10 projects: ',(proj_wise_count.head(10).sum()/proj_wise_co
          print('Percentage of Approval in the Top Project category: ',(proj_wise_count_approv
```

```
Percentage of Literacy & Language:  21.560303163444637
Percentage of Top 10 projects:  79.8643453427065
Percentage of Approval in the Top Project category:  86.98576050131187
```

Observation: **Top 10 Projects** Contribute **80%** of total Submissions, among which **'Literacy & Language'** Category Ranks 1st with **21%** of which **87%** of submissions were **Approved**.

4.3.3 Count of Projs in each SubCategory

```
In [127]: width = 0.5
          N = np.arange(20)
          proj_subCatg_wise_count = train_df['project_subject_subcategories'].value_counts()
          proj_subCatg_wise_count_approved = []
          proj_subCatg_wise_count_rejected = []

          for proj in proj_subCatg_wise_count.index:
              proj_subCatg_wise_count_approved.append(np.sum(train_df["project_is_approved"][t
              proj_subCatg_wise_count_rejected.append(np.sum(train_df["project_is_approved"][t


          plt.figure(figsize=(10,8))
          bar1 = plt.bar(N, proj_subCatg_wise_count_approved[0:20], width)
          bar2 = plt.bar(N, proj_subCatg_wise_count_rejected[0:20], width,
                         bottom=proj_subCatg_wise_count_approved[0:20])


          plt.xticks(N,proj_subCatg_wise_count.index,rotation=90)

          plt.title('Proj SubCategory vs No.of Submission in that Category')
          plt.xlabel('Project SubCategory')
          plt.ylabel('No.of Submissions')

          plt.legend((bar1[0], bar2[0]), ('Approved', 'Rejected'))

          plt.show()
```



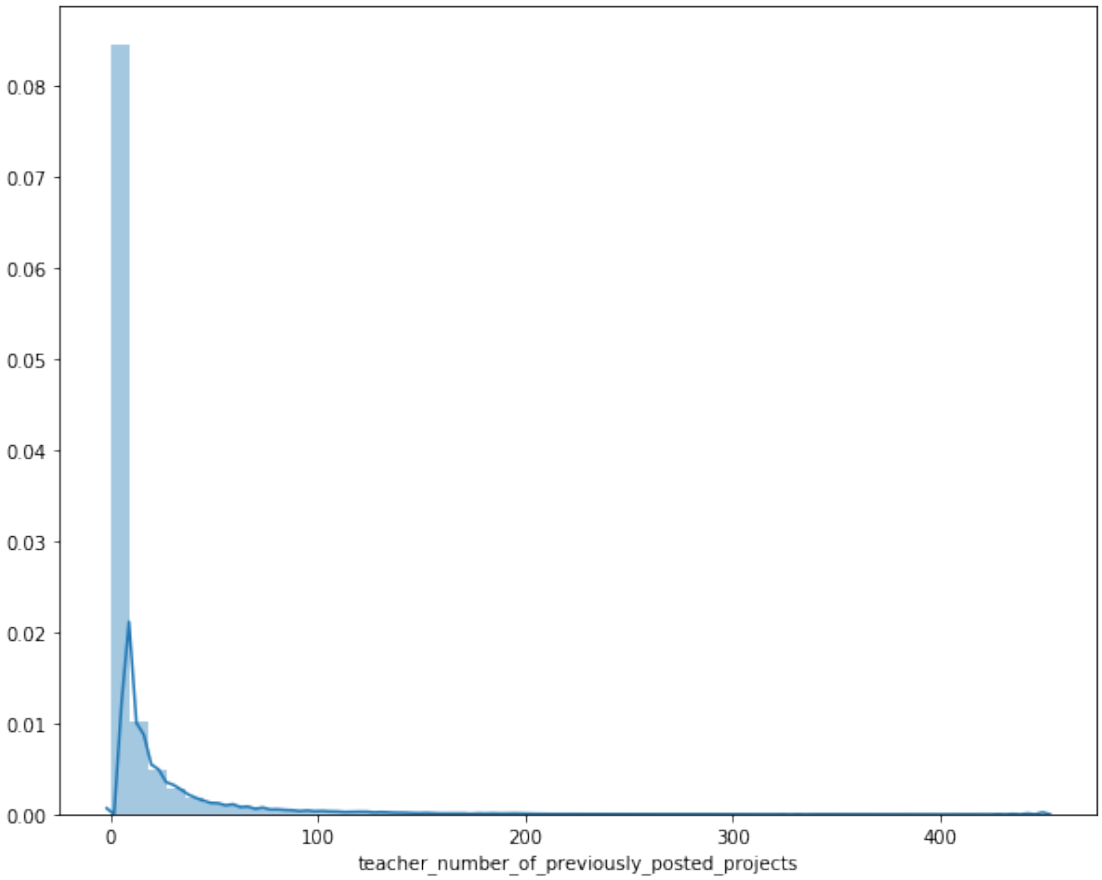

```
In [158]: print('Percentage of Sub Category - Literacy: ',(proj_subCatg_wise_count.head(1).sum
          print('Percentage of Top 10 projects: ',(proj_subCatg_wise_count.head(10).sum()/proj
          print('Percentage of Approval in the Top Project SubCategory: ',(proj_subCatg_wise_c
```

```
Percentage of Sub Category - Literacy:  8.66377416520211
Percentage of Top 10 projects:  48.408391915641474
Percentage of Approval in the Top Project SubCategory:  88.40570522979398
```

Observation: Here the distribution is widely spread among different SubCategories, among which **Literacy** is in the **top place** with **~9%** of which **88% of submission** in this category were **Approved**.

4.3.4 Lets see freq of Teacher no.of Previously Posted Projects

```
In [26]: plt.figure(figsize=(10,8))
         sns.distplot(train_df['teacher_number_of_previously_posted_projects'])
         plt.show()
```

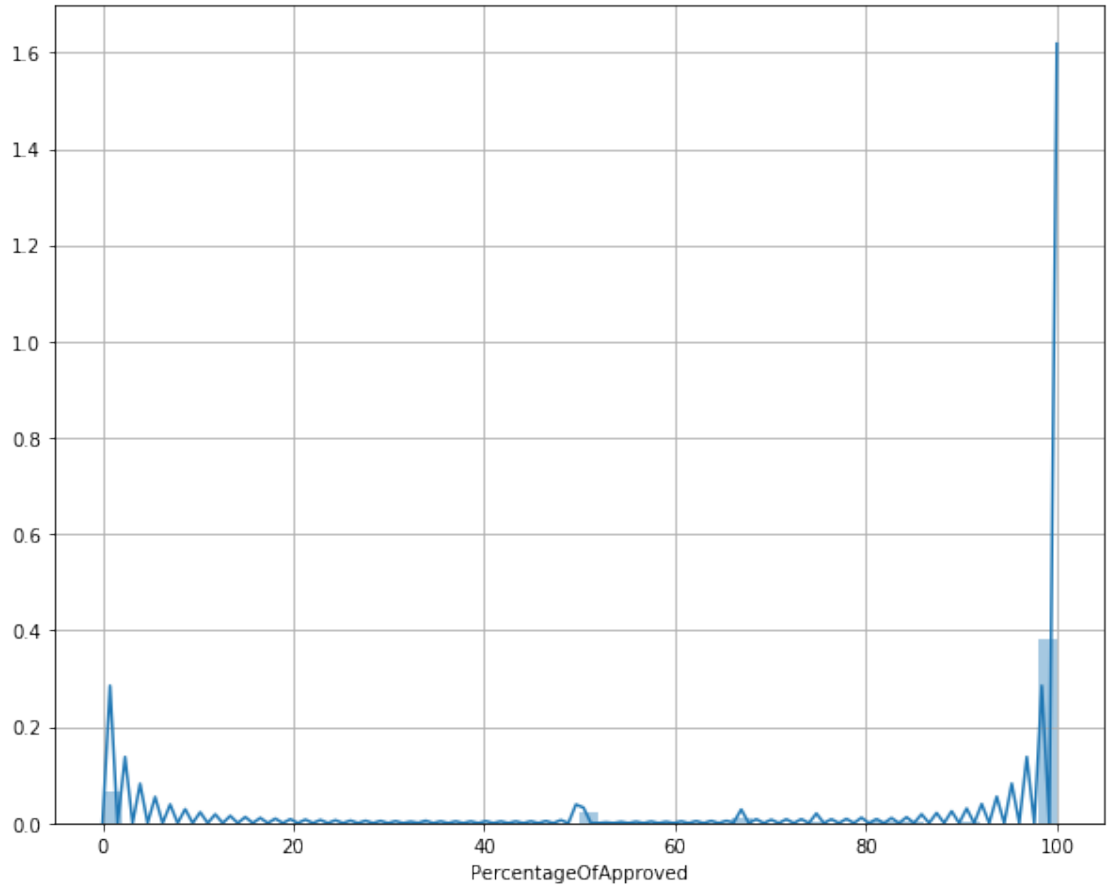


4.3.5 Lets See Success Rate of Project Approval teacher wise

```
In [27]: train_df_subset_cols = train_df[['teacher_id','teacher_number_of_previously_posted_pro
```

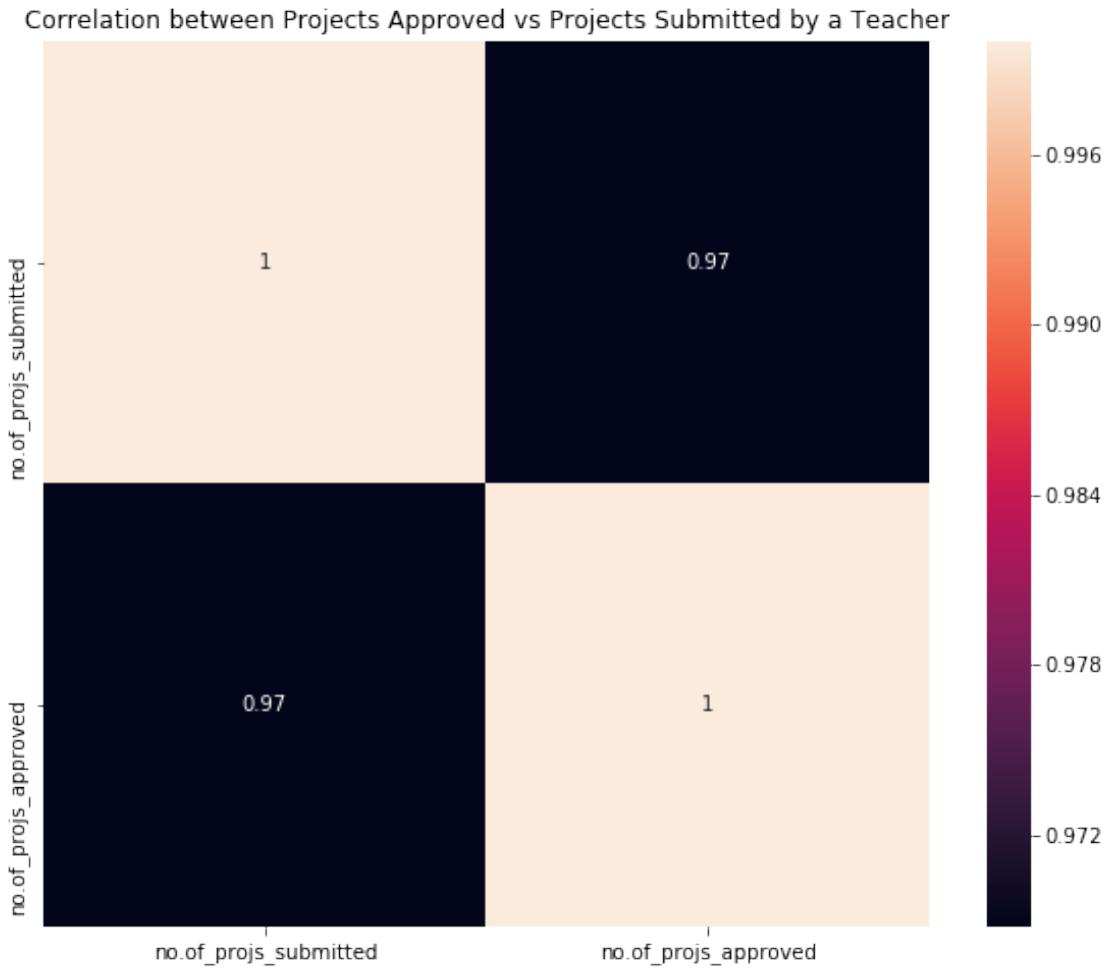
```
In [28]: grp_teacher_id = train_df_subset_cols.groupby('teacher_id')
         teacher_wise_approved_cnt = list(grp_teacher_id['project_is_approved'].sum())
         teacher_wise_proj_cnt = list(grp_teacher_id.apply(len))
         teacher_ids = list(grp_teacher_id.apply(len).index)
         df_submitted_vs_approved = pd.DataFrame(
             {'teacher_id': teacher_ids,
              'no.of_projs_submitted': teacher_wise_proj_cnt,
              'no.of_projs_approved': teacher_wise_approved_cnt
             })
         df_submitted_vs_approved['PercentageOfApproved'] = (df_submitted_vs_approved['no.of_p
```

In [29]:
```
plt.figure(figsize=(10,8))
sns.distplot(df_submitted_vs_approved['PercentageOfApproved'])
plt.grid()
plt.show()
```

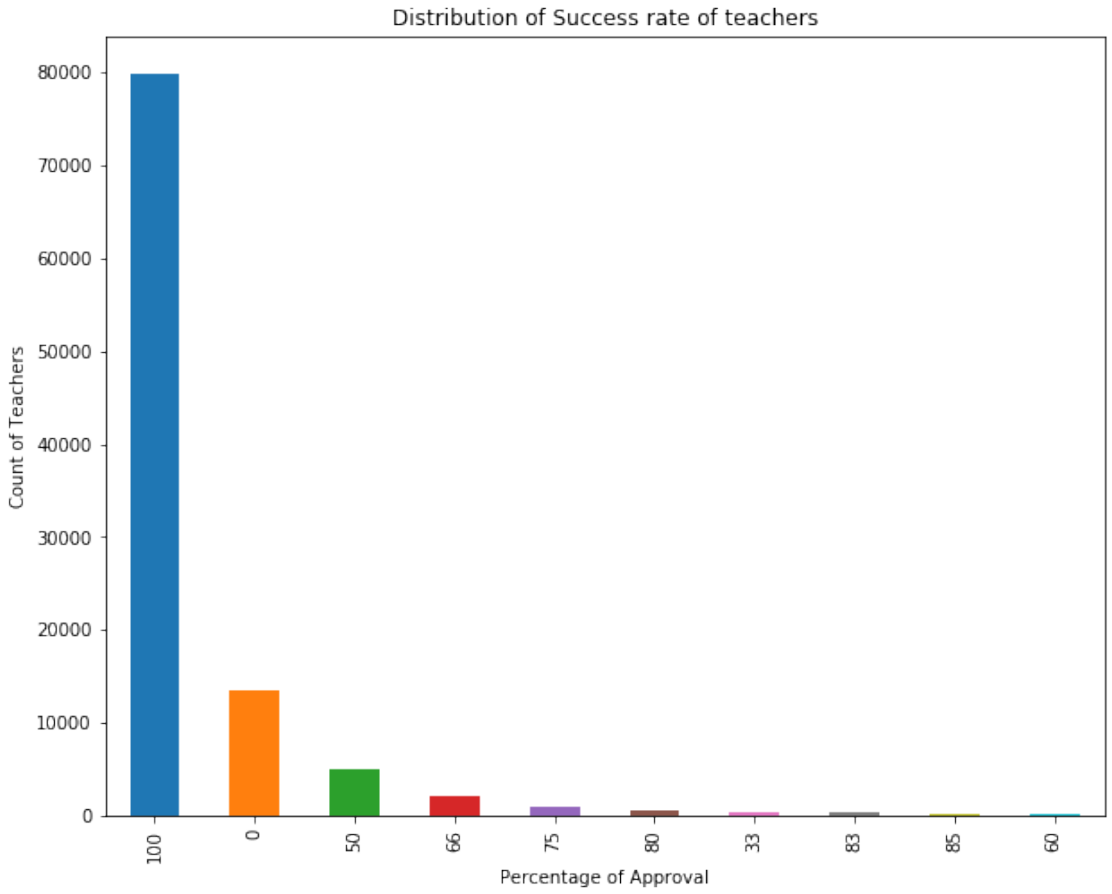


In [30]:
```
correlation = df_submitted_vs_approved[['no.of_projs_submitted','no.of_projs_approved
plt.figure(figsize=(10,8))
sns.heatmap(correlation,
            xticklabels=correlation.columns.values,
            yticklabels=correlation.columns.values, annot=True, square=True)
plt.title('Correlation between Projects Approved vs Projects Submitted by a Teacher')
```

Out[30]:
```
Text(0.5,1,'Correlation between Projects Approved vs Projects Submitted by a Teacher')
```




```
In [31]: df_submitted_vs_approved['PercentageOfApproved'] = df_submitted_vs_approved['Percentag

In [32]: count_at_that_percentage = df_submitted_vs_approved['PercentageOfApproved'].value_cou

In [33]: plt.figure(figsize=(10,8))
         count_at_that_percentage.head(10).plot(kind='bar')
         plt.title('Distribution of Success rate of teachers')
         plt.xlabel('Percentage of Approval')
         plt.ylabel('Count of Teachers')
         plt.show()
```



Explanation: We have calculated, Teacher wise "projects_approved/projects_submitted percentage" then performed Groupby on Percentage calculated to get the insights of success rate of teacher wise projects Approval.

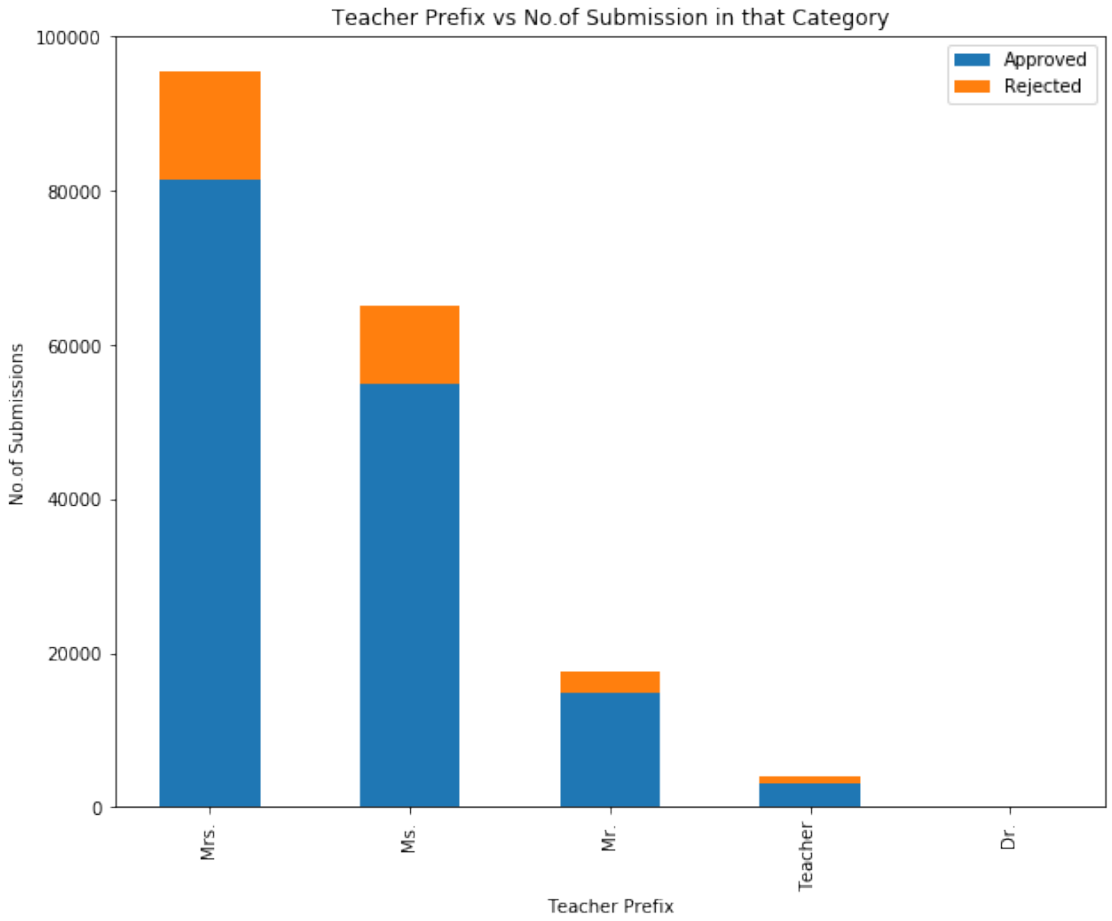
```
In [34]: count_at_that_percentage.head()

Out[34]: 100    79870
         0      13410
         50      5047
         66      2103
         75       994
         Name: PercentageOfApproved, dtype: int64

In [35]: print('Percentage of Teachers who has Success rate of 100% is: ',(count_at_that_perce
         print('Percentage of Teachers who has Success rate of 0%',(count_at_that_percentage[1

Percentage of Teachers who has Success rate of 100% is:  76.49357365870478
Percentage of Teachers who has Success rate of 0% 12.843105330702778
```

Observation: We can see that around **79870 (~76%)** of Teachers has **Success Rate of 100%** followed by 13410 (~13%) of the teachers has Success Rate of 0%.

4.3.6 Teacher Prefix wise Submissions

```
In [36]: teacher_prefix_wise_count = train_df['teacher_prefix'].value_counts()
         teacher_prefix_wise_count = (teacher_prefix_wise_count/teacher_prefix_wise_count.sum()
```

```
In [147]: width = 0.5
          N = np.arange(5)
          teacher_prefix_wise_count = train_df['teacher_prefix'].value_counts()
          #teacher_prefix_wise_count = (teacher_prefix_wise_count/teacher_prefix_wise_count.su
          teacher_prefix_wise_count_approved = []
          teacher_prefix_wise_count_rejected = []

          for teacher in teacher_prefix_wise_count.index:
              teacher_prefix_wise_count_approved.append(np.sum(train_df["project_is_approved"]
              teacher_prefix_wise_count_rejected.append(np.sum(train_df["project_is_approved"]


          plt.figure(figsize=(10,8))
          bar1 = plt.bar(N, teacher_prefix_wise_count_approved[0:5], width)
          bar2 = plt.bar(N, teacher_prefix_wise_count_rejected[0:5], width,
                         bottom=teacher_prefix_wise_count_approved[0:5])


          plt.xticks(N,teacher_prefix_wise_count.index,rotation=90)

          plt.title('Teacher Prefix vs No.of Submission in that Category')
          plt.xlabel('Teacher Prefix')
          plt.ylabel('No.of Submissions')

          plt.legend((bar1[0], bar2[0]), ('Approved', 'Rejected'))

          plt.show()
```




```
In [179]: print('Percentage of Prefix "Mrs": ',(teacher_prefix_wise_count.head(1).sum()/teache
          print('Percentage of Women i.e with Prefix "Mrs, Ms": ',(teacher_prefix_wise_count.he
          print('Percentage of Approval with prefix "Mrs": ',(teacher_prefix_wise_count_approv
```

```
Percentage of Prefix "Mrs":  52.39844899931897
Percentage of Women i.e with Prefix "Mrs, Ms":  88.13407588040158
Percentage of Approval with prefix "Mrs":  85.40852156595567
```

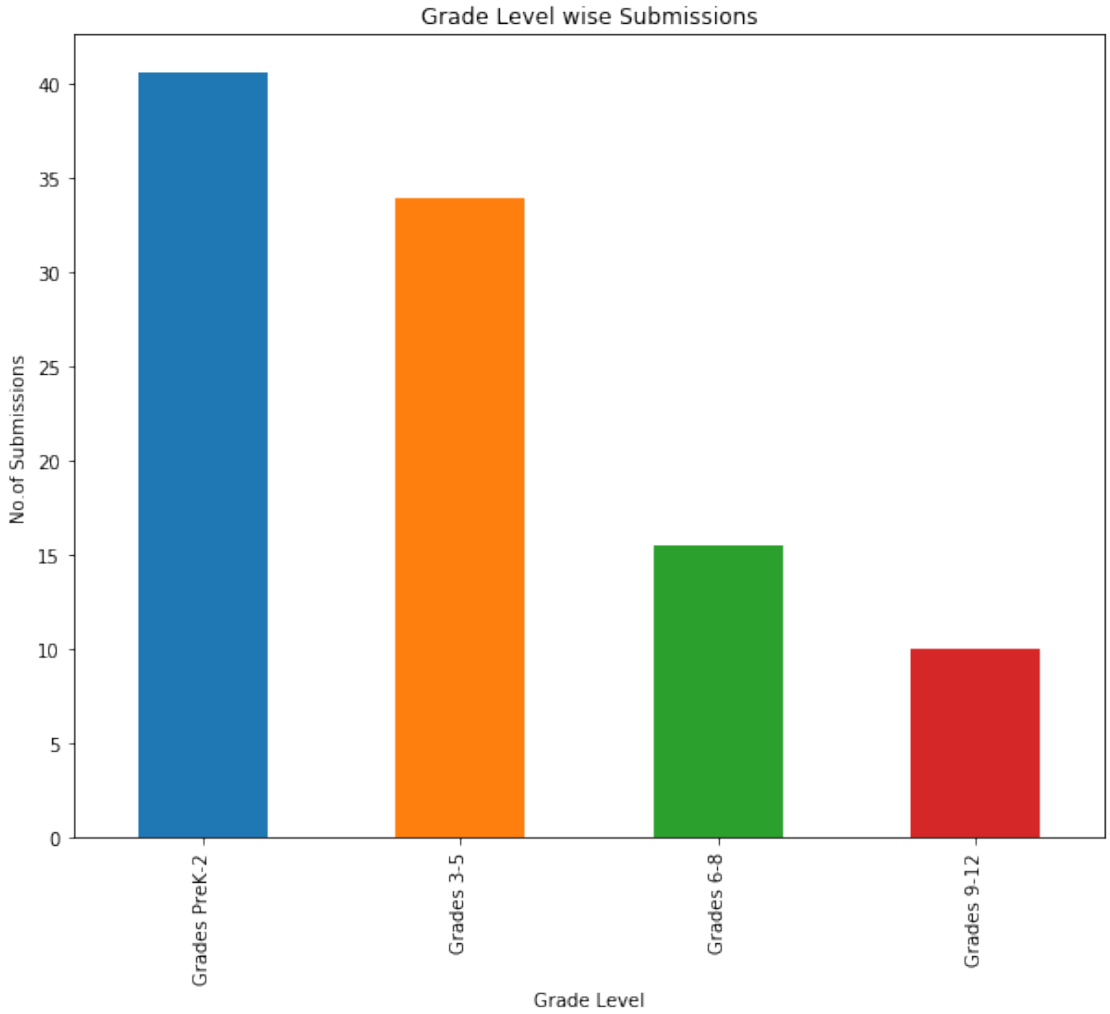
Observation: From the above stats we can see that Most of the submissions are from **Women with 88%**, Highest approval rate is 85.40 with Prefix Category 'Mrs' followed by 'Ms'.

4.3.7 School Grade wise Project submissions

```
In [38]: grade_wise_count = train_df['project_grade_category'].value_counts()
         # Converting Counts to percentages
         grade_wise_count = (grade_wise_count/grade_wise_count.sum()*100)
```

```
In [180]: grade_wise_count

Out[180]: Grades PreK-2    40.581063
          Grades 3-5       33.876318
          Grades 6-8       15.486050
          Grades 9-12      10.056569
          Name: project_grade_category, dtype: float64
```

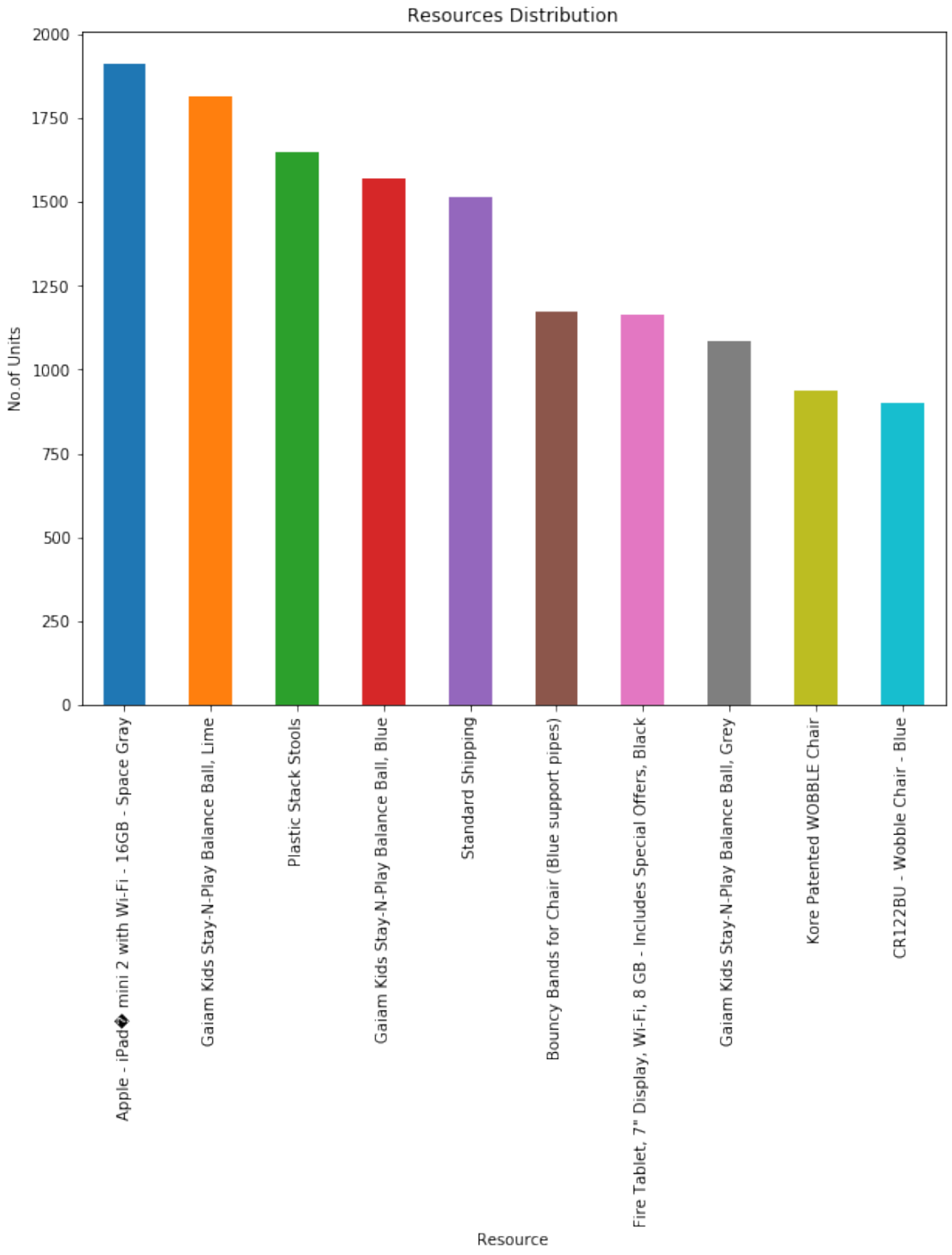
```
In [39]: plt.figure(figsize=(10,8))
         grade_wise_count.head(4).plot(kind='bar')
         plt.title('Grade Level wise Submissions')
         plt.xlabel('Grade Level')
         plt.ylabel('No.of Submissions')
         plt.show()
```



Observation: As Grade Level increases No.of Project Proposals Decreases

4.3.8 Resource Quantity Distribution

```
In [175]: resource_req_count = train_res_df['description'].value_counts()
          plt.figure(figsize=(10,8))
          resource_req_count.head(10).plot(kind='bar')
          plt.title('Resources Distribution')
          #Reducing the length of the description
          top_10_resource_names = list(resource_req_count.index[0:10])
          top_10_resource_names[2] = 'Plastic Stack Stools'
          top_10_resource_names[8] = 'Kore Patented WOBBLE Chair'
          plt.xticks(np.arange(10),top_10_resource_names)
          plt.xlabel('Resource')
          plt.ylabel('No.of Units')
          plt.ticklabel_format()

          plt.show()
```



Observation: Among all the resources, Most requested Resource is **Apple iPad mini 2**

```
In [159]: import re
          from nltk.corpus import stopwords
```

```
REPLACE_BY_SPACE_RE = re.compile('[/(){}\[\]\|@,;]')
SYMBOLS_RE = re.compile('[^0-9a-z #+_]')
STOPWORDS = set(stopwords.words('english'))
def text_prepare(text):
    text = text.lower()# lowercase text
    text = REPLACE_BY_SPACE_RE.sub(' ',text)# replace REPLACE_BY_SPACE_RE symbols by
    text = SYMBOLS_RE.sub('',text)# delete symbols which are in SYMBOLS_RE from text
    cleaned = [s.strip() for s in text.split() if s not in STOPWORDS]# delete stopwo
    new_text = ''
    for i in cleaned:
        new_text +=i+' '
    text = new_text
    return text.strip()
```

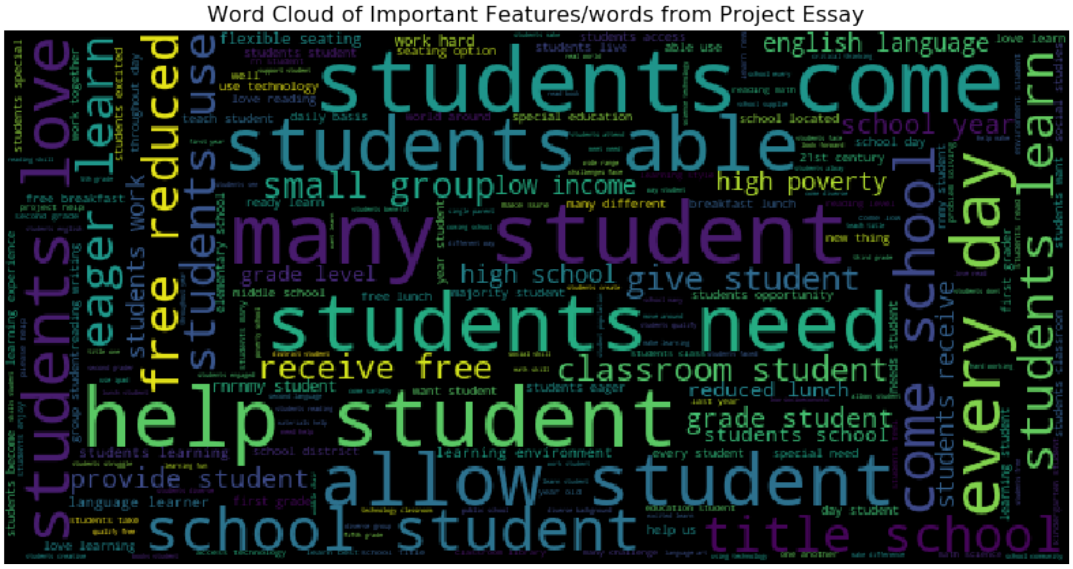
In [160]:
```
temp_data = train_res_df.dropna(subset=['description'])
# converting into lowercase
temp_data['description'] = temp_data['description'].apply(lambda x: " ".join(x.lower
temp_data['description'] = temp_data['description'].map(text_prepare)


from wordcloud import WordCloud

wordcloud = WordCloud(max_font_size=50, width=600, height=300).generate(' '.join(tem
plt.figure(figsize=(14,8))
plt.imshow(wordcloud)
plt.title("Word Cloud of Resources Requested", fontsize=16)
plt.axis("off")
plt.show()
```

Word Cloud of Resources Requested

4.3.9 WordCloud of Important Features/Words from Project Essay

```
In [46]: train_df['about_project'] = train_df[['project_essay_1', 'project_essay_2']].apply(la
```

```
In [49]: temp_data = train_df.dropna(subset=['about_project'])
         # converting into lowercase
         temp_data['about_project'] = temp_data['about_project'].apply(lambda x: " ".join(x.lo
         temp_data['about_project'] = temp_data['about_project'].map(text_prepare)


         from wordcloud import WordCloud

         wordcloud = WordCloud(max_font_size=50, width=600, height=300).generate(' '.join(temp_
         plt.figure(figsize=(14,8))
         plt.imshow(wordcloud)
         plt.title("Word Cloud of Important Features/words from Project Essay", fontsize=16)
         plt.axis("off")
         plt.show()
```



5 Data Preprocessing

5.1 Lets Remove unnecessary Features

```
In [64]: train_res_df = train_res_df.sort_values(by=['project_submitted_datetime'])
```

```
In [66]: train_res_df['Price'] = train_res_df['quantity']*train_res_df['price']
```

```
In [67]: train_res_df = train_res_df.drop(['quantity','price'],axis=1)

In [83]: '''pickle_out=open("beforeFeatures_Removal.pickle","wb")
         pickle.dump(train_res_df,pickle_out)
         pickle_out.close()'''

In [68]: train_res_dim_df = train_res_df.drop(['id','teacher_id','project_submitted_datetime',
```

5.2 Lets Convert Categorical and text Features to Numerical Features

```
In [69]: train_res_dim_df.head()
```

Out[69]:

	teacher_prefix	school_state	project_grade_category	\
763814	Ms.	CA	Grades 6-8	
1028203	Ms.	TX	Grades PreK-2	
1028198	Ms.	TX	Grades PreK-2	
1028199	Ms.	TX	Grades PreK-2	
1028200	Ms.	TX	Grades PreK-2	

	project_subject_categories	project_subject_subcategories	\
763814	Math & Science	Applied Sciences	
1028203	Literacy & Language, Math & Science	Foreign Languages, Mathematics	
1028198	Literacy & Language, Math & Science	Foreign Languages, Mathematics	
1028199	Literacy & Language, Math & Science	Foreign Languages, Mathematics	
1028200	Literacy & Language, Math & Science	Foreign Languages, Mathematics	

	project_essay_1	\
763814	I love giving my students experiences. A new e...	
1028203	We are getting closer to the end of the year, ...	
1028198	We are getting closer to the end of the year, ...	
1028199	We are getting closer to the end of the year, ...	
1028200	We are getting closer to the end of the year, ...	

	project_essay_2	\
763814	My students can vary quite dramatically. I hav...	
1028203	Welcome to our Pre-K classroom. We work hard e...	
1028198	Welcome to our Pre-K classroom. We work hard e...	
1028199	Welcome to our Pre-K classroom. We work hard e...	
1028200	Welcome to our Pre-K classroom. We work hard e...	

	teacher_number_of_previously_posted_projects	project_is_approved	\
763814	30	1	
1028203	1	1	
1028198	1	1	
1028199	1	1	
1028200	1	1	

	Price
763814	596.00

```
1028203    9.99
1028198   29.99
1028199   29.99
1028200   29.99
```

In our Data Set we have some categorical features like teacher_prefix, school_state,grade etc..
lets convert them to Numerical features

```
In [70]: categorical_features = ['teacher_prefix','school_state','project_grade_category']

         for feature in categorical_features:
             num = LabelEncoder()
             train_res_dim_df[feature] = num.fit_transform(train_res_dim_df[feature].astype('st
```

```
In [71]: train_res_dim_df.head()
```

```
Out[71]:          teacher_prefix  school_state  project_grade_category  \
         763814                3             4                       1
         1028203               3            43                       3
         1028198               3            43                       3
         1028199               3            43                       3
         1028200               3            43                       3

                          project_subject_categories   project_subject_subcategories  \
         763814                         Math & Science               Applied Sciences
         1028203  Literacy & Language, Math & Science  Foreign Languages, Mathematics
         1028198  Literacy & Language, Math & Science  Foreign Languages, Mathematics
         1028199  Literacy & Language, Math & Science  Foreign Languages, Mathematics
         1028200  Literacy & Language, Math & Science  Foreign Languages, Mathematics

                                                 project_essay_1  \
         763814   I love giving my students experiences. A new e...
         1028203  We are getting closer to the end of the year, ...
         1028198  We are getting closer to the end of the year, ...
         1028199  We are getting closer to the end of the year, ...
         1028200  We are getting closer to the end of the year, ...

                                                 project_essay_2  \
         763814   My students can vary quite dramatically. I hav...
         1028203  Welcome to our Pre-K classroom. We work hard e...
         1028198  Welcome to our Pre-K classroom. We work hard e...
         1028199  Welcome to our Pre-K classroom. We work hard e...
         1028200  Welcome to our Pre-K classroom. We work hard e...

                  teacher_number_of_previously_posted_projects  project_is_approved  \
         763814                                             30                   1
         1028203                                             1                   1
         1028198                                             1                   1
         1028199                                             1                   1
```

```
1028200                                                    1                    1
```

```
          Price
763814   596.00
1028203    9.99
1028198   29.99
1028199   29.99
1028200   29.99
```

Observation: As we can see The Categorical Features are converted to numerical Features

5.2.1 Lets Combine project_subject_categories and subcategories

```
In [72]: train_res_dim_df['project_subject'] = train_res_dim_df[['project_subject_categories',
```

```
In [73]: train_res_dim_df = train_res_dim_df.drop(['project_subject_categories','project_subje
         train_res_dim_df['project_subject'].head()
```

```
Out[73]: 763814                          Math & Science Applied Sciences
         1028203    Literacy & Language, Math & Science Foreign La...
         1028198    Literacy & Language, Math & Science Foreign La...
         1028199    Literacy & Language, Math & Science Foreign La...
         1028200    Literacy & Language, Math & Science Foreign La...
         Name: project_subject, dtype: object
```

Lets Convert Project Subject into word vector

```
In [74]: #considering only unigrams as text contains less no.of words
         tfidf_vect = TfidfVectorizer(ngram_range=(1,1))

         proj_subject_vector = tfidf_vect.fit_transform(train_res_dim_df['project_subject'])
```

```
In [75]: proj_subject_vector.shape
```

```
Out[75]: (1073254, 52)
```

5.2.2 Lets Combine Project essay 1 and Project essay 2

```
In [76]: train_res_dim_df['about_project'] = train_res_dim_df[['project_essay_1', 'project_essa
```

```
In [77]: train_res_dim_df = train_res_dim_df.drop(['project_essay_1','project_essay_2'],axis=1)
```

Lets Clean the Text

```
In [78]: def cleanpunc(sentence):
             cleaned = re.sub(r'[?|!|\'|"|#]',r'',sentence)
             cleaned = re.sub(r'[.|,|)|(|\|/]',r' ',cleaned)
             return cleaned
```

```
In [79]: start = datetime.now()
         i=0
         str1=' '
         final_string=[]
         stop = set(stopwords.words('english'))

         for sent in train_res_dim_df['about_project'].values:
             filtered_sentence=[]

             for w in sent.split():
                 for cleaned_words in cleanpunc(w).split():
                     if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                         if(cleaned_words.lower() not in stop):
                             filtered_sentence.append(cleaned_words.lower())
                         else:
                             continue
                     else:
                         continue

             #final string of cleaned words
             str1 = " ".join(filtered_sentence)


             final_string.append(str1)
             i+=1
             if(i%10000==0):
                 print("No of Sentences processed: ",i)
         print("Time Taken To Clean the data is: ",datetime.now() -start)
```

```
No of Sentences processed:  10000
No of Sentences processed:  20000
No of Sentences processed:  30000
No of Sentences processed:  40000
No of Sentences processed:  50000
No of Sentences processed:  60000
No of Sentences processed:  70000
No of Sentences processed:  80000
No of Sentences processed:  90000
No of Sentences processed:  100000
No of Sentences processed:  110000
No of Sentences processed:  120000
No of Sentences processed:  130000
No of Sentences processed:  140000
No of Sentences processed:  150000
No of Sentences processed:  160000
No of Sentences processed:  170000
No of Sentences processed:  180000
No of Sentences processed:  190000
```

```
No of Sentences processed:  200000
No of Sentences processed:  210000
No of Sentences processed:  220000
No of Sentences processed:  230000
No of Sentences processed:  240000
No of Sentences processed:  250000
No of Sentences processed:  260000
No of Sentences processed:  270000
No of Sentences processed:  280000
No of Sentences processed:  290000
No of Sentences processed:  300000
No of Sentences processed:  310000
No of Sentences processed:  320000
No of Sentences processed:  330000
No of Sentences processed:  340000
No of Sentences processed:  350000
No of Sentences processed:  360000
No of Sentences processed:  370000
No of Sentences processed:  380000
No of Sentences processed:  390000
No of Sentences processed:  400000
No of Sentences processed:  410000
No of Sentences processed:  420000
No of Sentences processed:  430000
No of Sentences processed:  440000
No of Sentences processed:  450000
No of Sentences processed:  460000
No of Sentences processed:  470000
No of Sentences processed:  480000
No of Sentences processed:  490000
No of Sentences processed:  500000
No of Sentences processed:  510000
No of Sentences processed:  520000
No of Sentences processed:  530000
No of Sentences processed:  540000
No of Sentences processed:  550000
No of Sentences processed:  560000
No of Sentences processed:  570000
No of Sentences processed:  580000
No of Sentences processed:  590000
No of Sentences processed:  600000
No of Sentences processed:  610000
No of Sentences processed:  620000
No of Sentences processed:  630000
No of Sentences processed:  640000
No of Sentences processed:  650000
No of Sentences processed:  660000
No of Sentences processed:  670000
```

```
No of Sentences processed:  680000
No of Sentences processed:  690000
No of Sentences processed:  700000
No of Sentences processed:  710000
No of Sentences processed:  720000
No of Sentences processed:  730000
No of Sentences processed:  740000
No of Sentences processed:  750000
No of Sentences processed:  760000
No of Sentences processed:  770000
No of Sentences processed:  780000
No of Sentences processed:  790000
No of Sentences processed:  800000
No of Sentences processed:  810000
No of Sentences processed:  820000
No of Sentences processed:  830000
No of Sentences processed:  840000
No of Sentences processed:  850000
No of Sentences processed:  860000
No of Sentences processed:  870000
No of Sentences processed:  880000
No of Sentences processed:  890000
No of Sentences processed:  900000
No of Sentences processed:  910000
No of Sentences processed:  920000
No of Sentences processed:  930000
No of Sentences processed:  940000
No of Sentences processed:  950000
No of Sentences processed:  960000
No of Sentences processed:  970000
No of Sentences processed:  980000
No of Sentences processed:  990000
No of Sentences processed:  1000000
No of Sentences processed:  1010000
No of Sentences processed:  1020000
No of Sentences processed:  1030000
No of Sentences processed:  1040000
No of Sentences processed:  1050000
No of Sentences processed:  1060000
No of Sentences processed:  1070000
Time Taken To Clean the data is:  0:13:37.606990
```

```
In [80]: train_res_dim_df['cleaned_about_project'] = final_string
```

```
In [81]: train_res_dim_df.head()
```

```
Out[81]:          teacher_prefix  school_state  project_grade_category  \
         763814                3             4                       1
```

```
1028203                3             43                       3
1028198                3             43                       3
1028199                3             43                       3
1028200                3             43                       3

         teacher_number_of_previously_posted_projects  project_is_approved  \
763814                                             30                   1
1028203                                             1                   1
1028198                                             1                   1
1028199                                             1                   1
1028200                                             1                   1

          Price                                    project_subject  \
763814   596.00                         Math & Science Applied Sciences
1028203    9.99  Literacy & Language, Math & Science Foreign La...
1028198   29.99  Literacy & Language, Math & Science Foreign La...
1028199   29.99  Literacy & Language, Math & Science Foreign La...
1028200   29.99  Literacy & Language, Math & Science Foreign La...

                                              about_project  \
763814   I love giving my students experiences. A new e...
1028203  We are getting closer to the end of the year, ...
1028198  We are getting closer to the end of the year, ...
1028199  We are getting closer to the end of the year, ...
1028200  We are getting closer to the end of the year, ...

                                     cleaned_about_project
763814   love giving students experiences new experienc...
1028203  getting closer end year still lot work get don...
1028198  getting closer end year still lot work get don...
1028199  getting closer end year still lot work get don...
1028200  getting closer end year still lot work get don...
```

In [82]:
```
'''pickle_out=open("cleanedData.pickle","wb")
pickle.dump(proj_subject_vector,pickle_out)
pickle.dump(final_string,pickle_out)
pickle.dump(train_res_dim_df,pickle_out)
pickle_out.close()'''
```

In [2]:
```
pickle_in=open("cleanedData.pickle","rb")
proj_subject_vector = pickle.load(pickle_in)
final_string = pickle.load(pickle_in)
train_res_dim_df = pickle.load(pickle_in)
pickle_in.close()
```

In [11]:
```
proj_subject_vector
```

Out[11]:
```
<1073254x52 sparse matrix of type '<class 'numpy.float64'>'
        with 4683669 stored elements in Compressed Sparse Row format>
```

```
In [8]: '''pickle_out=open("projSubjVector.pickle","wb")
        pickle.dump(proj_subject_vector,pickle_out)
        pickle_out.close()'''

In [3]: pickle_in=open("projSubjVector.pickle","rb")
        proj_subject_vector = pickle.load(pickle_in)

        pickle_in.close()
```

Lets Convert 'about project' into word vector

```
In [8]: start = datetime.now()
        #considering bigrams
        tfidf_bigram_vect = TfidfVectorizer(ngram_range=(1,2))

        about_proj_vector = tfidf_bigram_vect.fit_transform(train_res_dim_df['cleaned_about_pro
        print("Time Taken: ",datetime.now() -start)

Time Taken:  0:02:49.768380


In [10]: about_proj_vector.shape

Out[10]: (600000, 2273955)

In [12]: '''pickle_out=open("tfidf_vect_projEssay_30000dp.pickle","wb")
         pickle.dump(train_res_dim_df,pickle_out)
         pickle.dump(about_proj_vector,pickle_out)
         pickle.dump(proj_subject_vector,pickle_out)
         pickle_out.close()'''

In [13]: #del train_res_dim_df,about_proj_vector,proj_subject_vector

In [5]: pickle_in=open("tfidf_vect_projEssay_30000dp.pickle","rb")
        train_res_dim_df = pickle.load(pickle_in)
        about_proj_vector = pickle.load(pickle_in)
        proj_subject_vector = pickle.load(pickle_in)
        pickle_in.close()

In [11]: train_res_dim_df = train_res_dim_df[0:600000]

In [12]: scaler = StandardScaler()
         features = ['teacher_prefix','school_state', 'project_grade_category','teacher_number_

         for feature in features:
             train_res_dim_df[feature] = scaler.fit_transform(train_res_dim_df[feature].astype
```

6 Machine Learning Models

6.0.1 Features Considered

```
<li>Teacher Prefix</li>
<li>School State</li>
<li>Project Grade Category</li>
<li>Teacher no.of Previously posted Projects</li>
<li>Price</li>
<li>Project Subject(tfidf vector)</li>
<li>Project Essays(tfidf vector)</li>
```

6.0.2 Datapoints Considered

-> 600000(6 lakhs) out of 1000000(~10 lakhs) datapoints

6.0.3 Considered ML Models

LightGBM with GBDT, RF with Logloss -> As it is very suitable for large amount of data, unlike other boosting algorithms LGB grows the tree leaf wise(so that loss will be less), where as other algo(XGB) grows level wise.

XGB -> SKLearn's implementation is taking huge amount of time, so used XGB.Train, its very fast when compared to SKLearn's implimentation.

```
In [ ]: X = hstack((train_res_dim_df[['teacher_prefix', 'school_state', 'project_grade_categor
                'teacher_number_of_previously_posted_projects',
                'Price']],proj_subject_vector[0:600000],about_proj_vector))

In [14]: X = X.tocsr()

In [15]: Y = train_res_dim_df['project_is_approved'].to_sparse().as_matrix()

In [50]: '''pickle_out=open("train_test_Split_data.pickle","wb")
         pickle.dump(X,pickle_out)
         pickle.dump(Y,pickle_out)
         pickle_out.close()'''

In [2]: '''pickle_in=open("train_test_Split_data.pickle","rb")
        X = pickle.load(pickle_in)
        Y = pickle.load(pickle_in)
        pickle_in.close()'''

In [18]: x_train = X[0:550000]
         x_test = X[550000:600000]

In [19]: y_train = Y[0:550000]
         y_test = Y[550000:600000]

In [21]: y_train.sum()/y_train.shape[0]
```

```
Out[21]: 0.7809036363636364
```

```
In [182]: def confusionMatrix(y_test,pred):
              df_cm = pd.DataFrame(confusion_matrix(y_test, pred), index = ['False','True'],
                                   columns = ['False','True'])
              plt.figure(figsize=(8,6))
              sns.heatmap(df_cm, annot=True,fmt='d')

              plt.title('Confusion Matrix')
              plt.ylabel('Actual')
              plt.xlabel('Predicted')
              plt.figure(figsize=(12,10))
              plt.show()
          def auc_roc(y_test,pred):
              fpr, tpr, thresholds = roc_curve(y_test,pred)
              acc = roc_auc_score(y_test,pred)
              print("Area Under The Curve is : ",acc)

              plt.figure(figsize=(10,8))
              plt.plot(fpr, tpr, color='darkorange',
                       label='ROC curve (area/auc = %0.2f)' % acc)
              plt.plot([0, 1], [0, 1],'r--')
              plt.xlabel('False Positive Rate')
              plt.ylabel('True Positive Rate')
              plt.title('Receiver operating characteristic Curve')
              plt.legend(loc="lower right")
              plt.grid()
              plt.show()
```

6.1 LGB with GBDT Logloss

```
In [183]: def train_lgb(boosting_type='gbdt',max_depth=7,num_leaves=32,learning_rate=0.02,n_it
              params = {
                  'boosting_type': boosting_type,
                  'objective': 'binary',
                  'metric': 'auc',
                  'max_depth': max_depth,
                  'num_leaves': num_leaves,
                  'learning_rate': learning_rate,
                  'feature_fraction': 0.80,
                  'bagging_fraction': 0.80,
                  'bagging_freq': 5,
                  'verbose': 0,
                  'lambda_l2': 1,

              }
              evals_result = {}  # to record eval results for plotting
              model_lgb = lgb.train(
```

```
                    params,
                    lgb.Dataset(x_train, y_train),
                    num_boost_round=n_iter,
                    valid_sets=[lgb.Dataset(x_test, y_test)],
                    early_stopping_rounds=early_stopping_rounds,
                    evals_result=evals_result,
                    verbose_eval=verbose_eval)
            return model_lgb
```

```
In [184]: def eval_model(model,x_test,y_test):
              y_preds = model_lgb.predict(x_test, num_iteration=model.best_iteration)
              pred = [1 if i>=0.56 else 0 for i in y_preds]
              confusionMatrix(y_test,pred)
              auc_roc(y_test,y_preds)
              print(roc_auc_score(y_test, y_preds))
```

```
In [33]: '''pickle_out=open("lgb_73_auc.pickle","wb")
         pickle.dump(model_lgb,pickle_out)
         pickle.dump(evals_result,pickle_out)
         pickle.dump(x_train,pickle_out)
         pickle.dump(y_train,pickle_out)
         pickle.dump(x_test,pickle_out)
         pickle.dump(y_test,pickle_out)
         pickle_out.close()'''
```

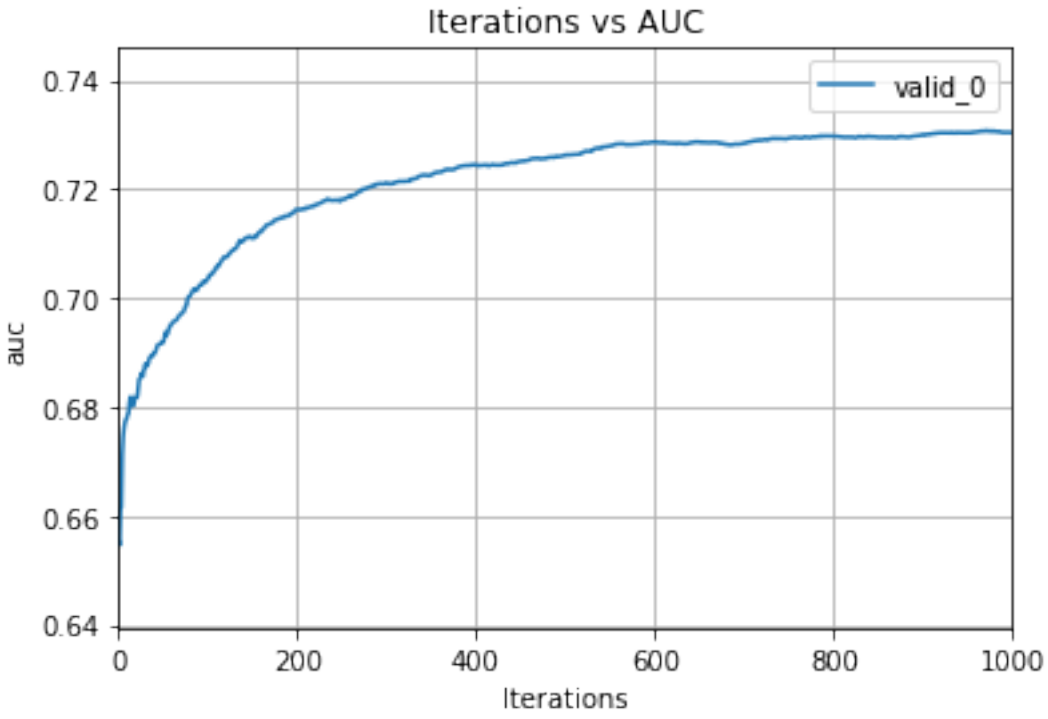
```
In [185]: pickle_in=open("lgb_73_auc.pickle","rb")
          model_lgb = pickle.load(pickle_in)
          evals_result = pickle.load(pickle_in)
          x_train = pickle.load(pickle_in)
          y_train = pickle.load(pickle_in)
          x_test = pickle.load(pickle_in)
          y_test = pickle.load(pickle_in)
          pickle_in.close()
```

```
In [186]: model_5 = train_lgb(max_depth=15,num_leaves=34)
```

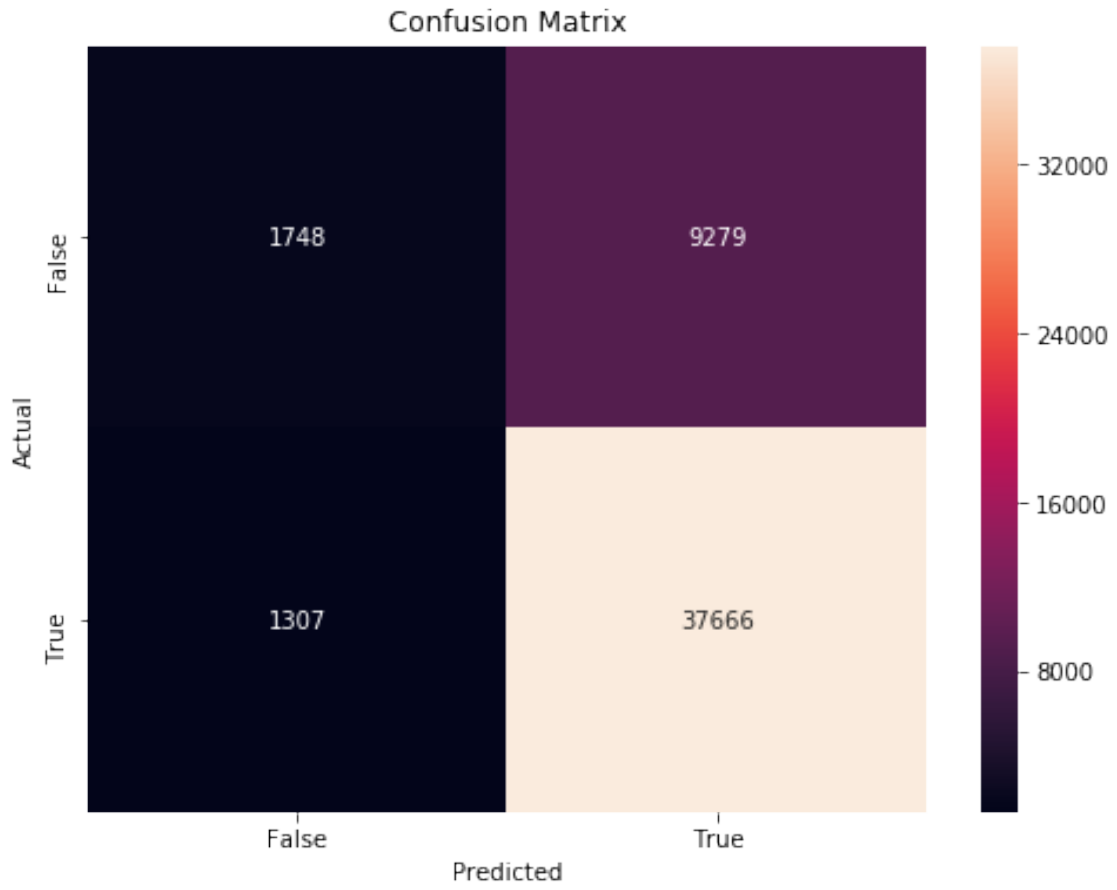
```
Training until validation scores don't improve for 25 rounds.
[25]        valid_0's auc: 0.688052
[50]        valid_0's auc: 0.693504
[75]        valid_0's auc: 0.700195
[100]        valid_0's auc: 0.705473
[125]        valid_0's auc: 0.709644
[150]        valid_0's auc: 0.713047
[175]        valid_0's auc: 0.717415
[200]        valid_0's auc: 0.719261
[225]        valid_0's auc: 0.721231
[250]        valid_0's auc: 0.72407
[275]        valid_0's auc: 0.725705
[300]        valid_0's auc: 0.727213
```

```
[325]        valid_0's auc: 0.728402
[350]        valid_0's auc: 0.729505
[375]        valid_0's auc: 0.730306
[400]        valid_0's auc: 0.731149
[425]        valid_0's auc: 0.732164
[450]        valid_0's auc: 0.732902
[475]        valid_0's auc: 0.733736
[500]        valid_0's auc: 0.734451
[525]        valid_0's auc: 0.734991
[550]        valid_0's auc: 0.735491
[575]        valid_0's auc: 0.735616
[600]        valid_0's auc: 0.73611
[625]        valid_0's auc: 0.736115
Early stopping, best iteration is:
[615]        valid_0's auc: 0.736214
```

```
In [208]: lgb.plot_metric(evals_result, metric='auc')
          plt.title('Iterations vs AUC')
          plt.show()
```

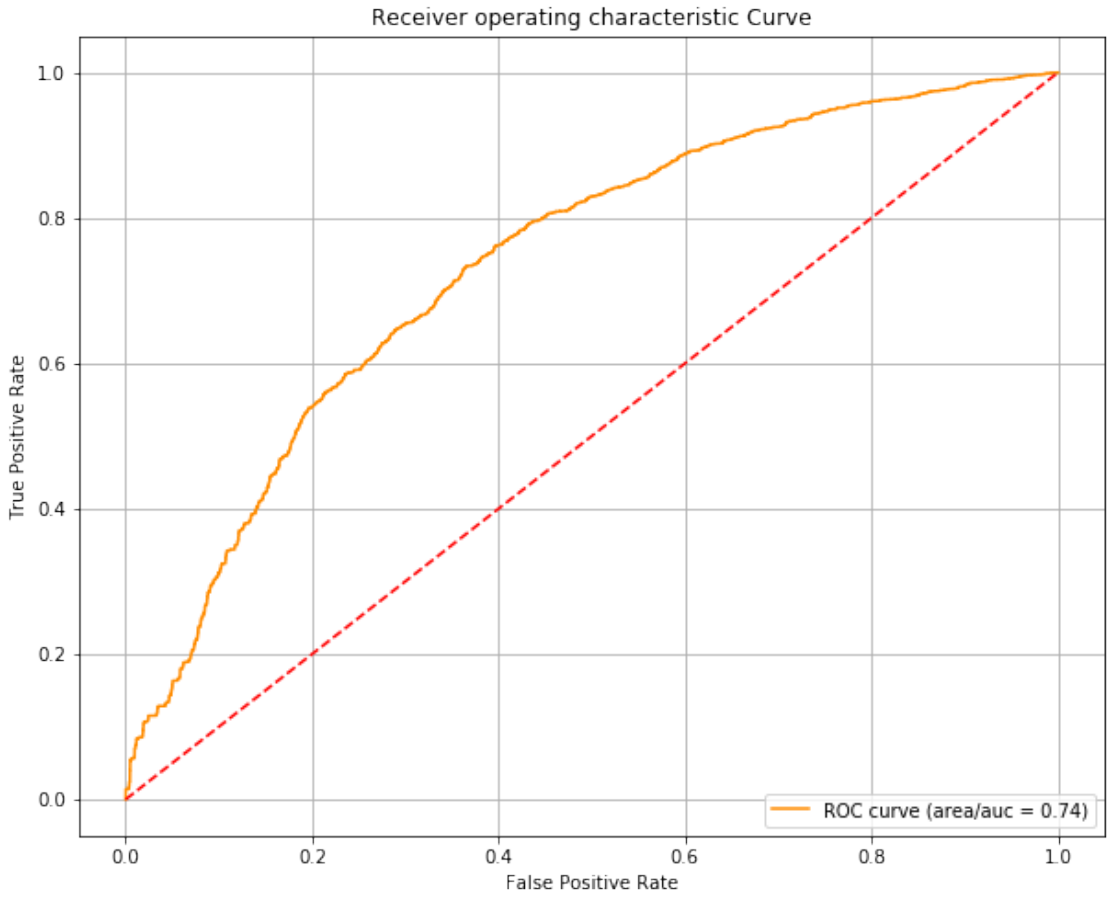



```
In [100]: eval_model(model_5,x_test,y_test)
```




```
<Figure size 864x720 with 0 Axes>
```

```
Area Under The Curve is :  0.7362143395327896
```



0.7362143395327896

6.2 XGBOOST

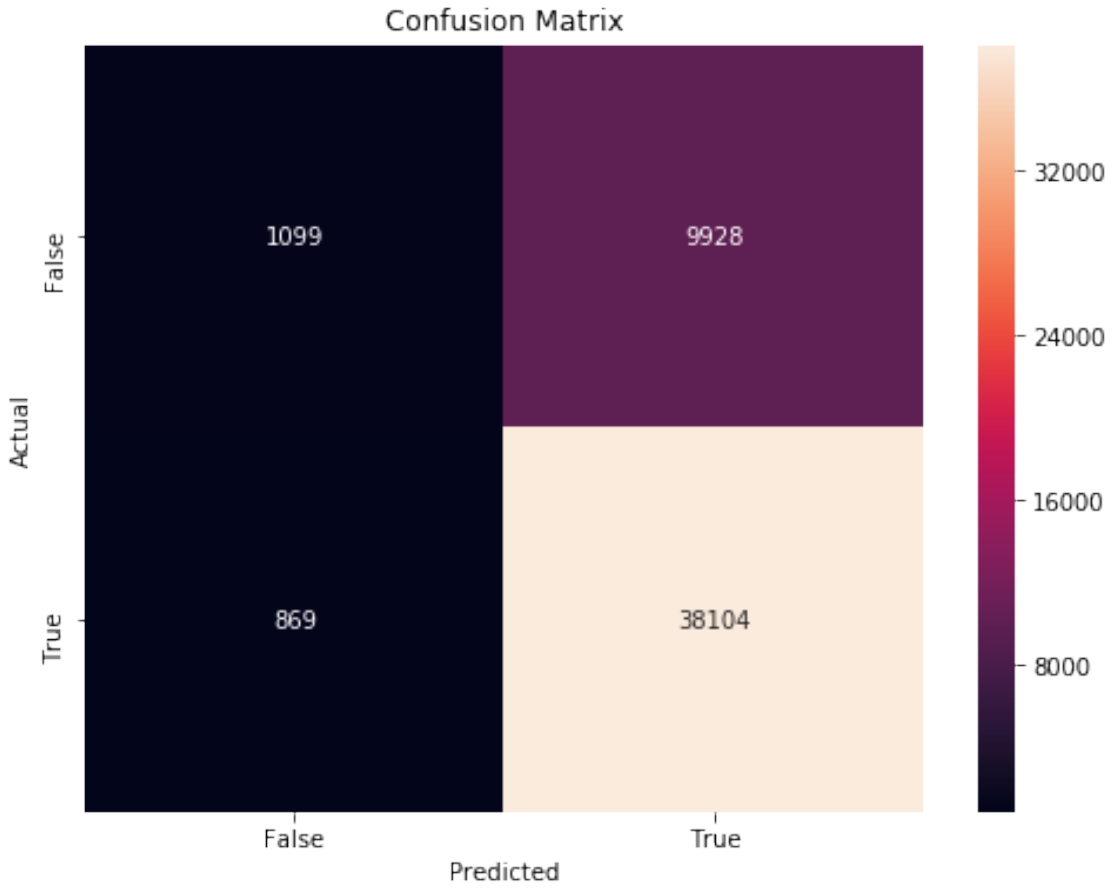
```
In [39]: import xgboost as xgb
         from sklearn.model_selection import train_test_split
         xgb_params = {'eta': 0.2,  #learning_rate
                       'max_depth': 5,
                       'subsample': 0.8,
                       'colsample_bytree': 0.8,
                       'objective': 'binary:logistic',  #logloss
                       'eval_metric': 'auc',
                       'seed': 1234
                      }

         d_train = xgb.DMatrix(x_train, y_train)
         d_valid = xgb.DMatrix(x_test, y_test)
```

```
        #watchlist -> List of items to be evaluated during training, this allows user to watch
        watchlist = [(d_train, 'train'), (d_valid, 'valid')]
        model_xgb = xgb.train(xgb_params, d_train, 200, watchlist, verbose_eval=50, early_stop

        xgb_pred_valid = model_xgb.predict(d_valid)
        auc = roc_auc_score(y_test, xgb_pred_valid)
        print('AUC:',auc)
```

```
[13:39:05] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[0]        train-auc:0.663331        valid-auc:0.652078
Multiple eval metrics have been passed: 'valid-auc' will be used for early stopping.

Will train until valid-auc hasn't improved in 5 rounds.
[13:39:15] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:39:25] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:39:37] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:39:49] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:40:01] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:40:13] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:40:24] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:40:34] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:40:44] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:40:56] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:41:06] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:41:18] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:41:30] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:41:42] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:41:52] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:42:04] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:42:16] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:42:26] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:42:36] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:42:47] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:42:57] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:43:09] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:43:21] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:43:31] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:43:43] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:43:55] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:44:05] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:44:17] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:44:28] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:44:40] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:44:51] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:45:02] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:45:12] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:45:22] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
```

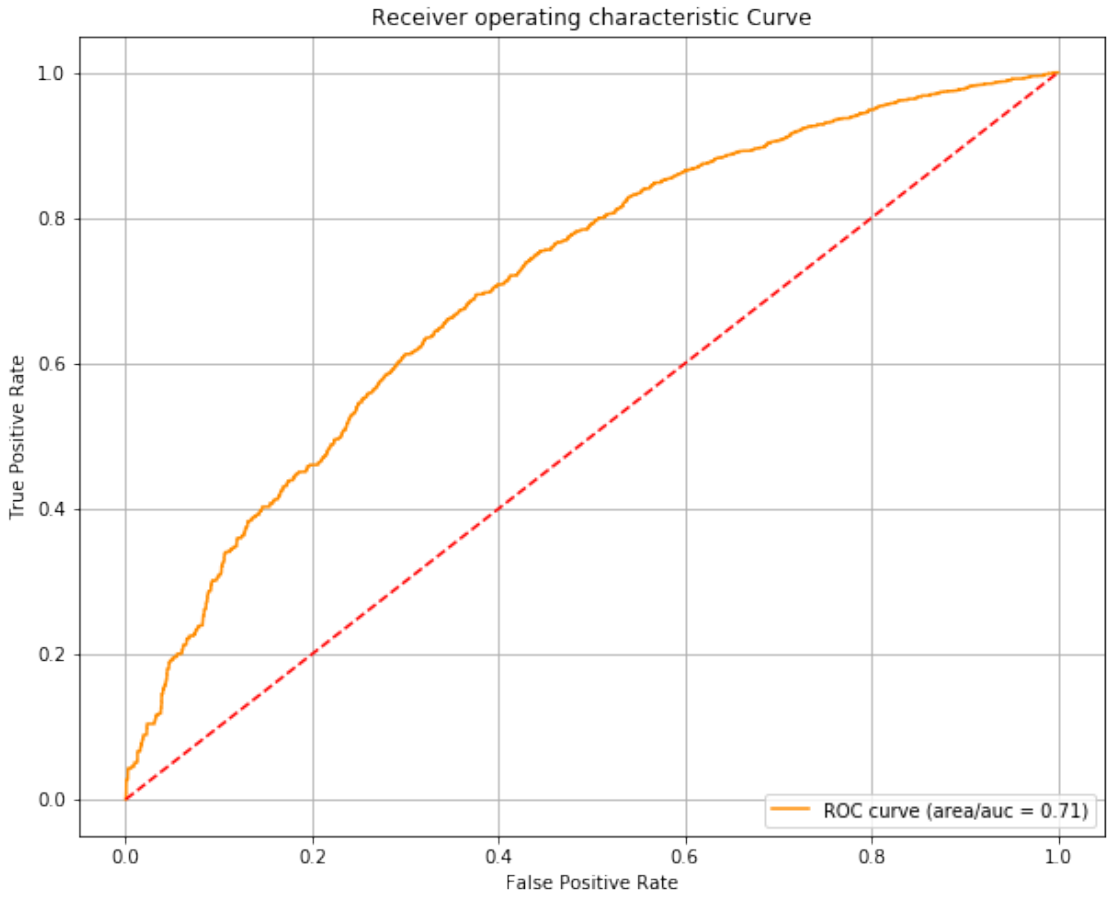
```
[13:45:32] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:45:41] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:45:53] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:46:04] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:46:15] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:46:25] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:46:38] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:46:48] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:47:00] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
[13:47:12] C:\Users\Administrator\Desktop\xgboost\src\tree\updater_prune.cc:74: tree pruning en
Stopping. Best iteration:
[39]        train-auc:0.826567        valid-auc:0.712545

AUC: 0.712009960431643
```

```
In [43]: pred = [1 if i>=0.56 else 0 for i in xgb_pred_valid]
         confusionMatrix(y_test,pred)
```




```
<Figure size 864x720 with 0 Axes>
```

```
In [49]: auc_roc(y_test,xgb_pred_valid)

Area Under The Curve is :  0.712009960431643
```



Observation: auc is 71% with max_depth of 5.

7 Summary:

7.1 Data Science

- Considered Training data for Analysis
- Data is Highly **Imbalanced** i.e **~80%** of data belongs to class 1(i.e Approved)
- Performed Data Cleaning
- Visualized the data using various types of plots to get the insights of the data.
- Some of the Observations/Conclusions:
  - **Max no.of projects** posted by a Teacher is **45**.

- Max no.of Submessions are from **California(CA)**
- Max no.of Submessions are recorded in **Literacy & Language** Project Category with **21%** of which **87%** were **Approved**
- Around **9%** of Submissions recorded from **Literacy** Project SubCategory which Stood first among other SubCategories.
- Around **76%** of Teachers has **Success rate** of **100%**, followed by **13%** with **0 Success Rate**
- Most of the Submissions are from **Married Women** Teachers with contribution of **52%** and has Highest Approval Percentage(**85%**) than other Prefixs.
- Project Proposals are very high from Women than Men, with **88%** Contribution from **Women**.
- Teachers with prefix **Dr** has shown very less interest in Project Proposals with contribution percentage of **0.014%**.
- Around **41%** of Project proposals are for **Grades PreK-2**
- **Higher Grades** has **less** no.of Project **Proposals**
- Most **Trending/Requested Resource** is **Apple iPad Mini 2 with WiFi 16GB Spacy Gray** followed by **Galam kids Stay-N-Play Balance Ball**

7.2 Machine Learning

- Done Data Preprocessing i.e removed unnecessary columns/features, converted Categorical Features to Numerical Features, Sentences to vectors(using tfidf vectorizer).
- Standardized the data.
- Total Data Points Considered **600000** out of **1000000**
  - Train data points **550000**
  - Test data points **50000**
- Machine Learning Models used:
  - **Light Gradient Boost with LogLoss**
  - **XGBoost with LogLoss**
- Validation Metric used:
  - **AUC**
- Tuned HyperParameters:
  - LGB:
    - max_depth
    - num_leaves
    - learning_rate
  - XGBoost:
    - eta
    - max_depth
- Best Result Achieved using **LGB** with **74% AUC**

In [ ]: