Exploratory Data Analysis

DonorsChoose

Reference:

Required files of this Dataset can be found from this link:

https://www.kaggle.com/datasets/vineet28vi/donors-choose-dataset?select=train_data.csv

About the DonorsChoose Data Set:

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

1. **project_id**: A unique identifier for the proposed project. Example: p036502
2. **project_title**: Title of that given project
3. **project_grade_category**: Grade level of students for which the project is targeted. One of the following enumerated values:
4. **project_subject_categories**:

   One or more (comma-separated) subject categories for the project from the following enumerated list of values:

   Applied Learning

   Care & Hunger

   Health & Sports

   History & Civics

   Literacy & Language

   Math & Science

   Music & The Arts

   Special Needs

   Warmth
5. **school_state**: State where school is located (Two-letter U.S. postal code). **Example**: WY

6. **project_subject_subcategories**:

   One or more (comma-separated) subject subcategories for the project.

   Examples:

   Literacy

   Literature & Writing, Social Sciences

7. **project_resource_summary**: An explanation of the resources needed for the project.

   Example:

   My students need hands on literacy materials to manage sensory needs!

8. **project_essay_1**: First application essay
9. **project_essay_2**: Second application essay
10. **project_essay_3**: Thired application essay
11. **project_essay_4**: Fourth applicaiton essay
12. **project_submitted_datetime**: Date and time on which project is submitted
13. **teacher_id**: A unique identifier for the teacher of the proposed project.

    Example:

    bdf8baa8fedef6bfeec7ae4ff1c15c56

14. **teacher_prefix**: Teacher's title.

    One of the following enumerated values:

    nan

    Dr.

    Mr.

    Mrs.

    Ms.

    Teacher.

15. **teacher_number_of_previously_posted_projects**: Number of project applications previously submitted by the same teacher.

    Example:

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:

1. **id**: A project_id value from the train.csv file. Example: p036502
2. **description**: Desciption of the resource. Example: Tenor Saxophone Reeds, Box of 25
3. **quantity**: Quantity of the resource required. Example: 3
4. **prince**: 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):

1. **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_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?"

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.

```
%matplotlib inline
import warnings
```

```
warnings.filterwarnings("ignore")

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

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

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

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

from tqdm import tqdm
import os

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

1.1 Reading Data

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

```
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']
```

```
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']
```

```
        id                                           description
quantity  \
0  p233245  LC652 - Lakeshore Double-Space Mobile Drying Rack
1
1  p069063        Bouncy Bands for Desks (Blue support pipes)
3


    price
0  149.00
1   14.95
```

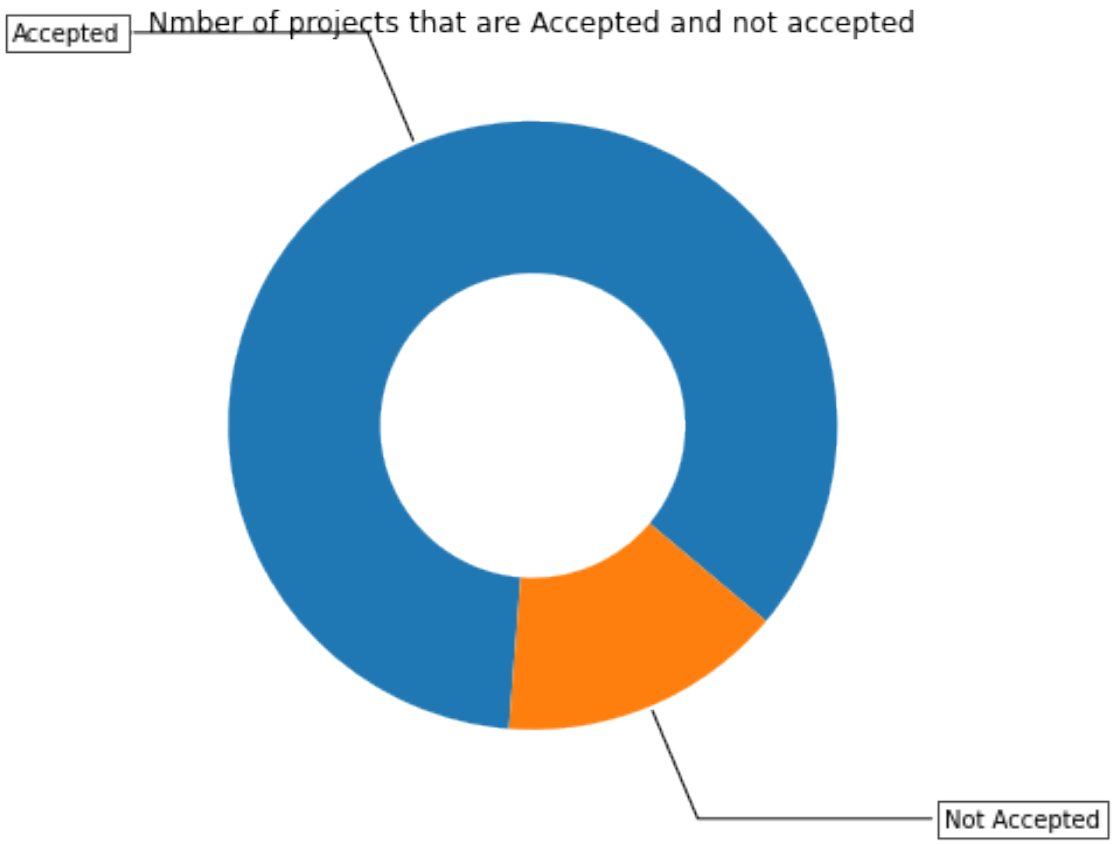
1.2 Data Analysis

```
# this code is taken from
#
https://matplotlib.org/gallery/pie_and_polar_charts/pie_and_donut_labe
ls.html#sphx-glr-gallery-pie-and-polar-charts-pie-and-donut-labels-py
```

```
y_value_counts = project_data['project_is_approved'].value_counts()
print("Number of projects thar are approved for funding ",
y_value_counts[1], ", (",
(y_value_counts[1]/(y_value_counts[1]+y_value_counts[0]))*100,"%)")
print("Number of projects thar are not approved for funding ",
y_value_counts[0], ", (",
(y_value_counts[0]/(y_value_counts[1]+y_value_counts[0]))*100,"%)")

fig, ax = plt.subplots(figsize=(6, 6),
subplot_kw=dict(aspect="equal"))
recipe = ["Accepted", "Not Accepted"]

data = [y_value_counts[1], y_value_counts[0]]

wedges, texts = ax.pie(data, wedgeprops=dict(width=0.5), startangle=-
40)

bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
```

```
kw = dict(xycoords='data', textcoords='data',
arrowprops=dict(arrowstyle="-"),
          bbox=bbox_props, zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1)/2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = "angle,angleA=0,angleB={}".format(ang)
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(recipe[i], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                horizontalalignment=horizontalalignment, **kw)

ax.set_title("Nmber of projects that are Accepted and not accepted")

plt.show()
```

```
Number of projects thar are approved for funding  92706 ,
( 84.85830404217927 %)
Number of projects thar are not approved for funding  16542 ,
( 15.141695957820739 %)
```



1.2.1 Univariate Analysis: School State

```
!pip install chart_studio
from chart_studio import plotly as py
```

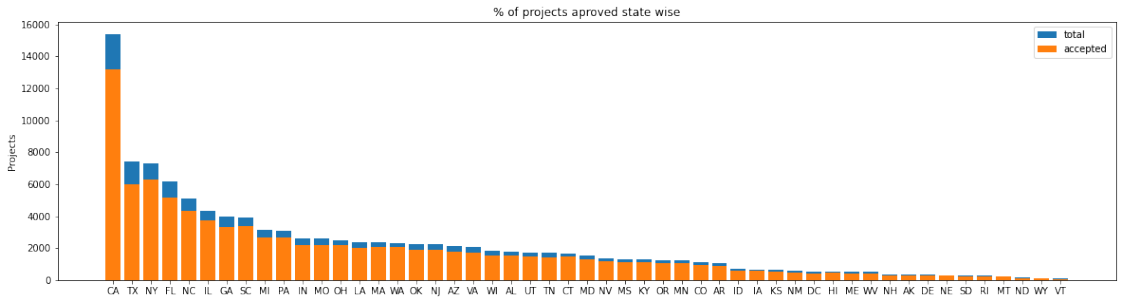
```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting chart_studio
  Downloading chart_studio-1.1.0-py3-none-any.whl (64 kB)
ent already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from chart_studio) (2.23.0)
Collecting retrying>=1.3.3
  Downloading retrying-1.3.3.tar.gz (10 kB)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-
packages (from chart_studio) (1.15.0)
Requirement already satisfied: plotly in
/usr/local/lib/python3.7/dist-packages (from chart_studio) (5.5.0)
Requirement already satisfied: tenacity>=6.2.0 in
/usr/local/lib/python3.7/dist-packages (from plotly->chart_studio)
(8.1.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /usr/local/lib/python3.7/dist-packages (from requests-
>chart_studio) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests->chart_studio)
(3.0.4)
Requirement already satisfied: idna<3,>=2.5 in
/usr/local/lib/python3.7/dist-packages (from requests->chart_studio)
(2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests->chart_studio)
(2022.9.24)
Building wheels for collected packages: retrying
  Building wheel for retrying (setup.py) ... e=retrying-1.3.3-py3-
none-any.whl size=11447
sha256=6f3fb84da5a8519c659f80cc00dcf512cfec37eb391c2ca8bddd22eb7f170fc
9
  Stored in directory:
/root/.cache/pip/wheels/f9/8d/8d/f6af3f7f9eea3553bc2fe6d53e4b287dad18b
06a861ac56ddf
Successfully built retrying
Installing collected packages: retrying, chart-studio
Successfully installed chart-studio-1.1.0 retrying-1.3.3
```

```
print("Number of projects submitted per each state in USA:")
project_data.school_state.value_counts()
```

```
Number of projects submitted per each state in USA

CA    15388
TX     7396
NY     7318
```

```
FL     6185
NC     5091
IL     4350
GA     3963
SC     3936
MI     3161
PA     3109
IN     2620
MO     2576
OH     2467
LA     2394
MA     2389
WA     2334
OK     2276
NJ     2237
AZ     2147
VA     2045
WI     1827
AL     1762
UT     1731
TN     1688
CT     1663
MD     1514
NV     1367
MS     1323
KY     1304
OR     1242
MN     1208
CO     1111
AR     1049
ID      693
IA      666
KS      634
NM      557
DC      516
HI      507
ME      505
WV      503
NH      348
AK      345
DE      343
NE      309
SD      300
RI      285
MT      245
ND      143
WY       98
VT       80
Name: school_state, dtype: int64
```

```
#stacked bar plots matplotlib:
https://matplotlib.org/gallery/lines_bars_and_markers/bar_stacked.html
def stack_plot(data, xtick, col2='project_is_approved', col3='total'):
    ind = np.arange(data.shape[0])

    plt.figure(figsize=(20,5))
    p1 = plt.bar(ind, data[col3].values)
    p2 = plt.bar(ind, data[col2].values)

    plt.ylabel('Projects')
    plt.title('% of projects aproved state wise')
    plt.xticks(ind, list(data[xtick].values))
    plt.legend((p1[0], p2[0]), ('total', 'accepted'))
    plt.show()

def univariate_barplots(data, col1, col2='project_is_approved',
top=False):
    # Count number of zeros in dataframe python:
https://stackoverflow.com/a/51540521/4084039
    temp = pd.DataFrame(project_data.groupby(col1)[col2].agg(lambda x:
x.eq(1).sum())).reset_index()

    # Pandas dataframe grouby count:
https://stackoverflow.com/a/19385591/4084039
    temp['total'] = pd.DataFrame(project_data.groupby(col1)
[col2].agg(total = 'count')).reset_index()['total']
    temp['Avg'] = pd.DataFrame(project_data.groupby(col1)
[col2].agg(Avg = 'mean')).reset_index()['Avg']

    temp.sort_values(by=['total'],inplace=True, ascending=False)

    if top:
        temp = temp[0:top]

    stack_plot(temp, xtick=col1, col2=col2, col3='total')
    print(temp.head(5))
    print("="*50)
    print(temp.tail(5))

univariate_barplots(project_data, 'school_state',
'project_is_approved', False)
```

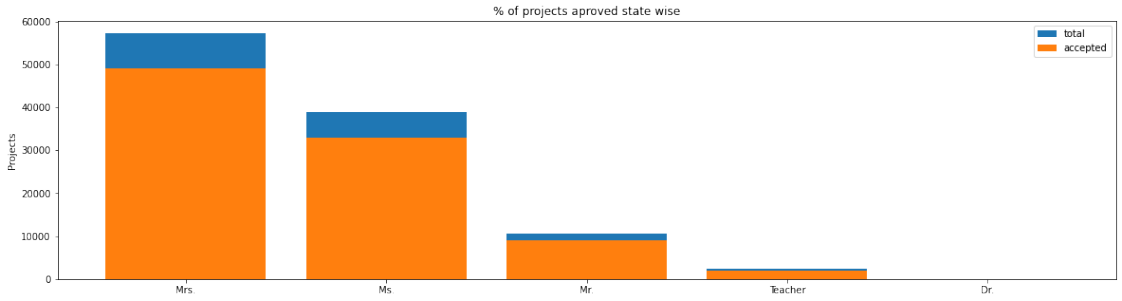


```
   school_state  project_is_approved  total       Avg
4            CA                13205  15388  0.858136
43           TX                 6014   7396  0.813142
34           NY                 6291   7318  0.859661
9            FL                 5144   6185  0.831690
27           NC                 4353   5091  0.855038
==================================================
   school_state  project_is_approved  total       Avg
39           RI                  243    285  0.852632
26           MT                  200    245  0.816327
28           ND                  127    143  0.888112
50           WY                   82     98  0.836735
46           VT                   64     80  0.800000
```

Every state is having more than 80% success rate in approval

1.2.2 Univariate Analysis: teacher_prefix

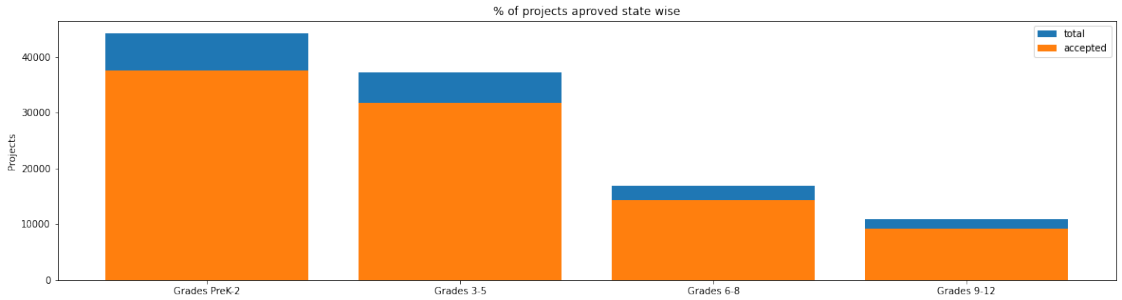
```
univariate_barplots(project_data, 'teacher_prefix',
'project_is_approved' , top=False)
```




```
  teacher_prefix  project_is_approved  total       Avg
2           Mrs.                48997  57269  0.855559
3            Ms.                32860  38955  0.843537
1            Mr.                 8960  10648  0.841473
4        Teacher                 1877   2360  0.795339
0            Dr.                    9     13  0.692308
==================================================
  teacher_prefix  project_is_approved  total       Avg
2           Mrs.                48997  57269  0.855559
3            Ms.                32860  38955  0.843537
1            Mr.                 8960  10648  0.841473
4        Teacher                 1877   2360  0.795339
0            Dr.                    9     13  0.692308
```

1.2.3 Univariate Analysis: project_grade_category

```
univariate_barplots(project_data, 'project_grade_category',
'project_is_approved', top=False)
```



	project_grade_category	project_is_approved	total	Avg
3	Grades PreK-2	37536	44225	0.848751
0	Grades 3-5	31729	37137	0.854377
1	Grades 6-8	14258	16923	0.842522
2	Grades 9-12	9183	10963	0.837636

==========

	project_grade_category	project_is_approved	total	Avg
3	Grades PreK-2	37536	44225	0.848751
0	Grades 3-5	31729	37137	0.854377
1	Grades 6-8	14258	16923	0.842522
2	Grades 9-12	9183	10963	0.837636

1.2.4 Univariate Analysis: project_subject_categories

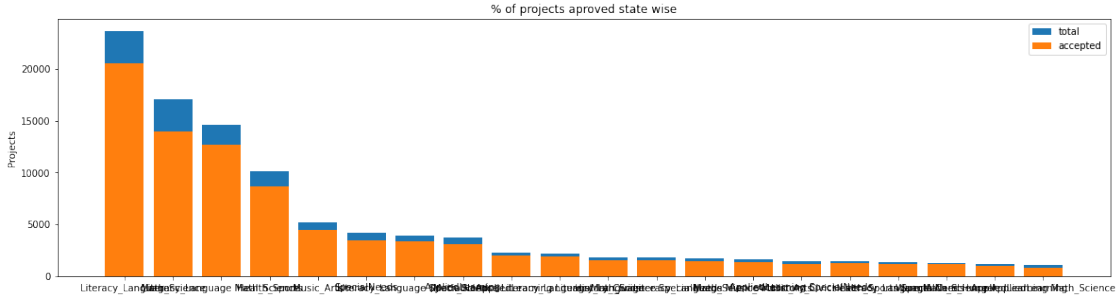
```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python:
https://stackoverflow.com/a/47301924/4084039

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

```
project_data['clean_categories'] = cat_list
project_data.drop(['project_subject_categories'], axis=1,
inplace=True)
project_data.head(2)
```

```
   Unnamed: 0       id                         teacher_id
teacher_prefix  \
0      160221  p253737  c90749f5d961ff158d4b4d1e7dc665fc
Mrs.
1      140945  p258326  897464ce9ddc600bced1151f324dd63a
Mr.

  school_state project_submitted_datetime project_grade_category  \
0           IN        2016-12-05 13:43:57          Grades PreK-2
1           FL        2016-10-25 09:22:10             Grades 6-8

      project_subject_subcategories  \
0                       ESL, Literacy
1  Civics & Government, Team Sports

                                      project_title  \
0  Educational Support for English Learners at Home
1            Wanted: Projector for Hungry Learners

                                     project_essay_1  \
0  My students are English learners that are work...
1  Our students arrive to our school eager to lea...

                                     project_essay_2
project_essay_3  \
0  \"The limits of your language are the limits o...             NaN

1  The projector we need for our school is very c...             NaN


  project_essay_4
project_resource_summary  \
0             NaN  My students need opportunities to practice beg...

1             NaN  My students need a projector to help with view...


   teacher_number_of_previously_posted_projects
project_is_approved  \
0                                             0                    0

1                                             7                    1
```

```
              clean_categories
0              Literacy_Language
1  History_Civics Health_Sports
```

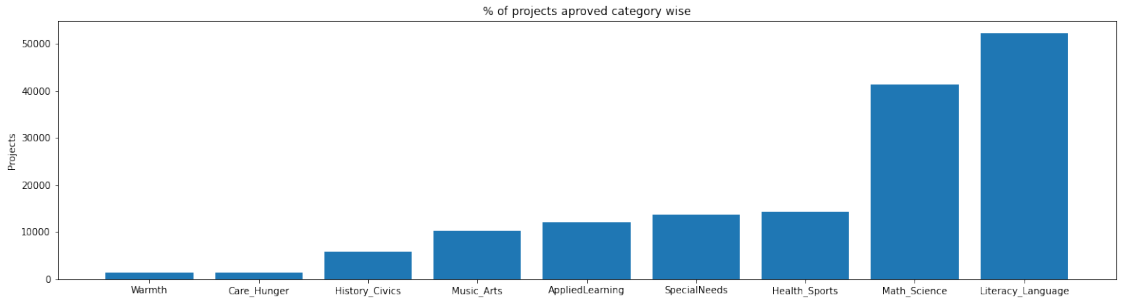
```
univariate_barplots(project_data, 'clean_categories',
'project_is_approved', top=20)
```




```
                   clean_categories  project_is_approved  total
Avg
24                Literacy_Language                20520  23655
0.867470
32                     Math_Science                13991  17072
0.819529
28  Literacy_Language Math_Science                12725  14636
0.869432
8                    Health_Sports                 8640  10177
0.848973
40                      Music_Arts                 4429   5180
0.855019
==================================================
                     clean_categories  project_is_approved  total
Avg
19  History_Civics Literacy_Language                 1271   1421
0.894441
14        Health_Sports SpecialNeeds                 1215   1391
0.873472
50                Warmth Care_Hunger                 1212   1309
0.925898
33      Math_Science AppliedLearning                 1019   1220
0.835246
4       AppliedLearning Math_Science                  855   1052
0.812738
```

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

```
# dict sort by value python:
https://stackoverflow.com/a/613218/4084039
cat_dict = dict(my_counter)
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))


ind = np.arange(len(sorted_cat_dict))
plt.figure(figsize=(20,5))
p1 = plt.bar(ind, list(sorted_cat_dict.values()))

plt.ylabel('Projects')
plt.title('% of projects aproved category wise')
plt.xticks(ind, list(sorted_cat_dict.keys()))
plt.show()
```




```
for i, j in sorted_cat_dict.items():
    print("{:20} :{:10}".format(i,j))
```

```
Warmth               :      1388
Care_Hunger          :      1388
History_Civics       :      5914
Music_Arts           :     10293
AppliedLearning      :     12135
SpecialNeeds         :     13642
Health_Sports        :     14223
Math_Science         :     41421
Literacy_Language    :     52239
```

1.2.5 Univariate Analysis: project_subject_subcategories

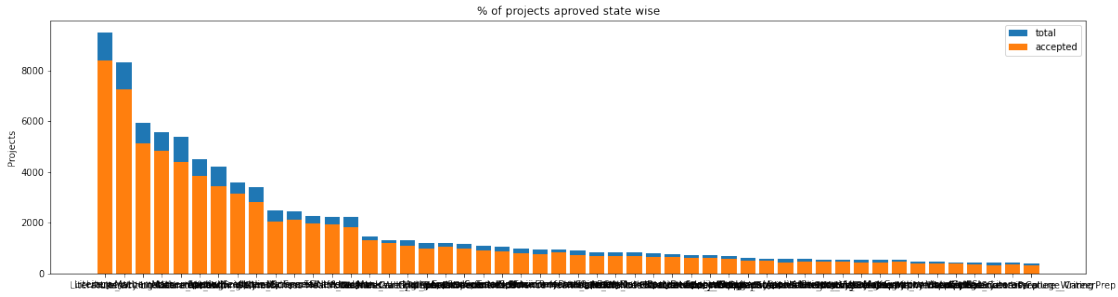
```
sub_catogories =
list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python:
https://stackoverflow.com/a/47301924/4084039

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

sub_cat_list = []
```

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

```
project_data['clean_subcategories'] = sub_cat_list
project_data.drop(['project_subject_subcategories'], axis=1,
inplace=True)
project_data.head(2)
```

```
   Unnamed: 0       id                         teacher_id
teacher_prefix  \
0      160221  p253737  c90749f5d961ff158d4b4d1e7dc665fc
Mrs.
1      140945  p258326  897464ce9ddc600bced1151f324dd63a
Mr.

  school_state project_submitted_datetime project_grade_category  \
0           IN       2016-12-05 13:43:57          Grades PreK-2
1           FL       2016-10-25 09:22:10             Grades 6-8

                                     project_title  \
0  Educational Support for English Learners at Home
1            Wanted: Projector for Hungry Learners

                                    project_essay_1  \
0  My students are English learners that are work...
1  Our students arrive to our school eager to lea...

                                    project_essay_2
project_essay_3  \
0  \"The limits of your language are the limits o...             NaN

1  The projector we need for our school is very c...             NaN


  project_essay_4
project_resource_summary  \
```

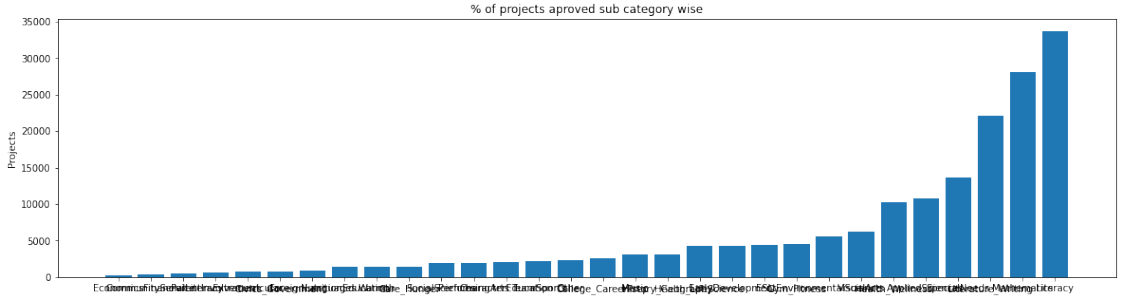
```
0             NaN  My students need opportunities to practice beg...

1             NaN  My students need a projector to help with view...


   teacher_number_of_previously_posted_projects
project_is_approved  \
0                                             0                    0

1                                             7                    1


              clean_categories           clean_subcategories
0             Literacy_Language                  ESL Literacy
1  History_Civics Health_Sports  Civics_Government TeamSports

univariate_barplots(project_data, 'clean_subcategories',
'project_is_approved', top=50)
```




```
                clean_subcategories  project_is_approved  total
Avg
317                         Literacy                 8371   9486
0.882458
319            Literacy Mathematics                 7260   8325
0.872072
331  Literature_Writing Mathematics                 5140   5923
0.867803
318     Literacy Literature_Writing                 4823   5571
0.865733
342                     Mathematics                 4385   5379
0.815207
==================================================
                   clean_subcategories  project_is_approved  total
Avg
196       EnvironmentalScience Literacy                  389    444
0.876126
127                                ESL                  349    421
0.828979
79                  College_CareerPrep                  343    421
0.814727
17   AppliedSciences Literature_Writing                  361    420
```

```
0.859524
3    AppliedSciences College_CareerPrep                    330    405
0.814815
```

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

# dict sort by value python:
https://stackoverflow.com/a/613218/4084039
sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv:
kv[1]))


ind = np.arange(len(sorted_sub_cat_dict))
plt.figure(figsize=(20,5))
p1 = plt.bar(ind, list(sorted_sub_cat_dict.values()))

plt.ylabel('Projects')
plt.title('% of projects aproved sub category wise')
plt.xticks(ind, list(sorted_sub_cat_dict.keys()))
plt.show()
```



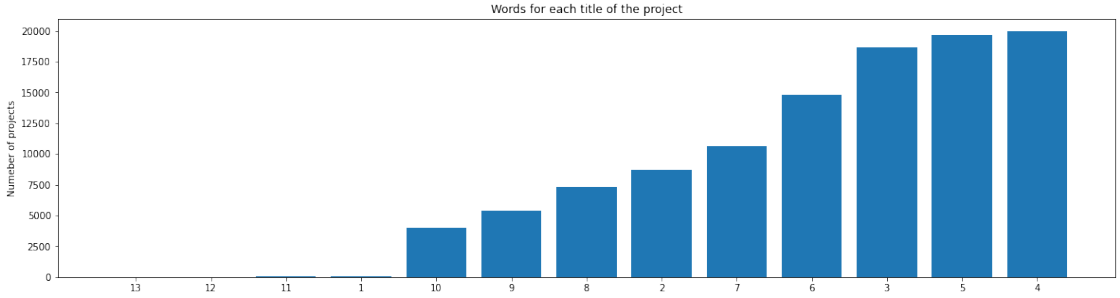

```
for i, j in sorted_sub_cat_dict.items():
    print("{:20} :{:10}".format(i,j))
```

```
Economics            :       269
CommunityService     :       441
FinancialLiteracy    :       568
ParentInvolvement    :       677
Extracurricular      :       810
Civics_Government    :       815
ForeignLanguages     :       890
NutritionEducation   :      1355
Warmth               :      1388
Care_Hunger          :      1388
SocialSciences       :      1920
```

```
PerformingArts       :      1961
CharacterEducation   :      2065
TeamSports           :      2192
Other                :      2372
College_CareerPrep   :      2568
Music                :      3145
History_Geography    :      3171
Health_LifeScience   :      4235
EarlyDevelopment     :      4254
ESL                  :      4367
Gym_Fitness          :      4509
EnvironmentalScience :      5591
VisualArts           :      6278
Health_Wellness      :     10234
AppliedSciences      :     10816
SpecialNeeds         :     13642
Literature_Writing   :     22179
Mathematics          :     28074
Literacy             :     33700
```

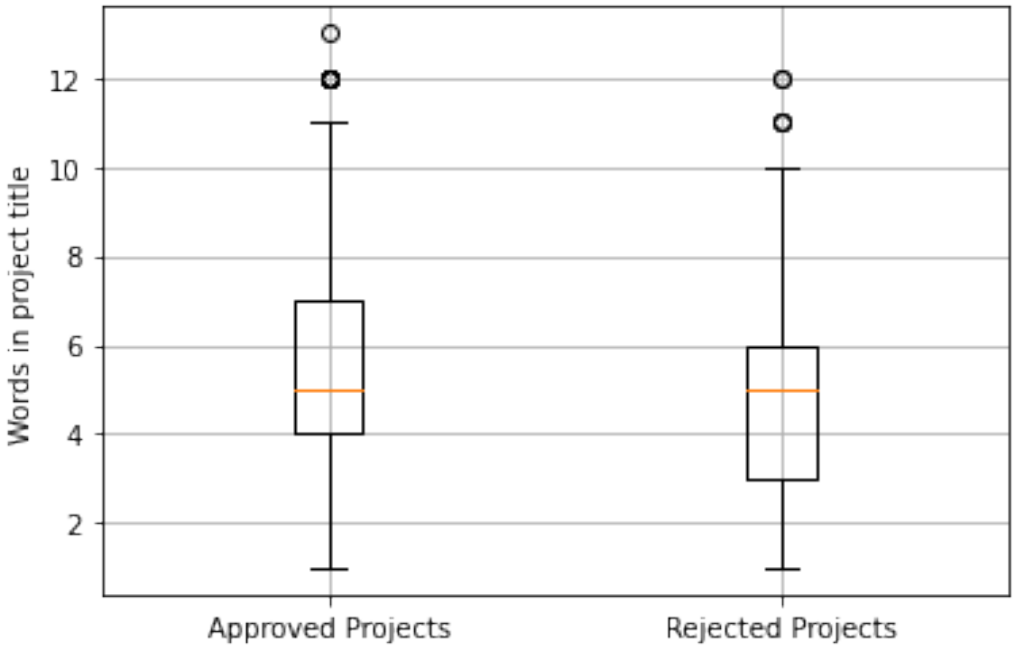
1.2.6 Univariate Analysis: Text features (Title)

```
#How to calculate number of words in a string in DataFrame:
https://stackoverflow.com/a/37483537/4084039
word_count =
project_data['project_title'].str.split().apply(len).value_counts()
word_dict = dict(word_count)
word_dict = dict(sorted(word_dict.items(), key=lambda kv: kv[1]))


ind = np.arange(len(word_dict))
plt.figure(figsize=(20,5))
p1 = plt.bar(ind, list(word_dict.values()))

plt.ylabel('Numeber of projects')
plt.title('Words for each title of the project')
plt.xticks(ind, list(word_dict.keys()))
plt.show()
```

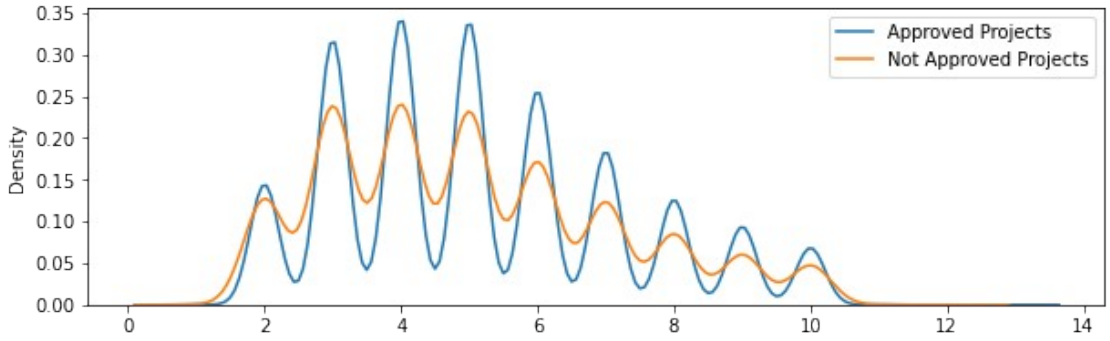



```
approved_word_count =
project_data[project_data['project_is_approved']==1]
['project_title'].str.split().apply(len)
```

```
approved_word_count = approved_word_count.values

rejected_word_count =
project_data[project_data['project_is_approved']==0]
['project_title'].str.split().apply(len)
rejected_word_count = rejected_word_count.values

# https://glowingpython.blogspot.com/2012/09/boxplot-with-
matplotlib.html
plt.boxplot([approved_word_count, rejected_word_count])
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Words in project title')
plt.grid()
plt.show()
```

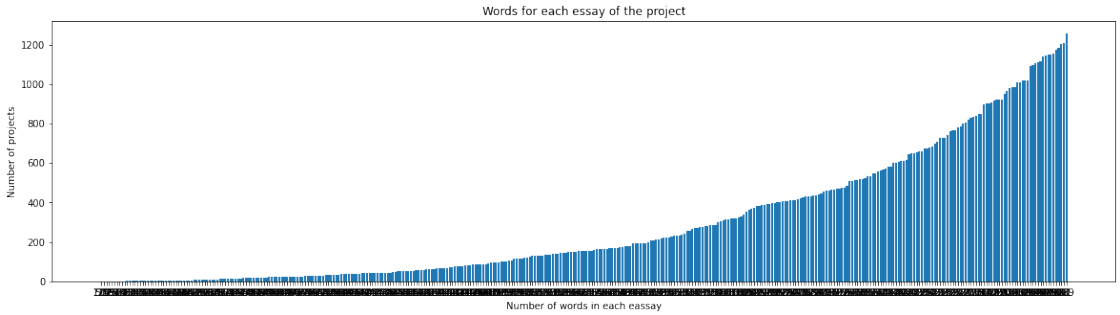



```
plt.figure(figsize=(10,3))
sns.distplot(approved_word_count, hist=False, label="Approved
Projects")
sns.distplot(rejected_word_count, hist=False, label="Not Approved
Projects")
plt.legend()
plt.show()
```

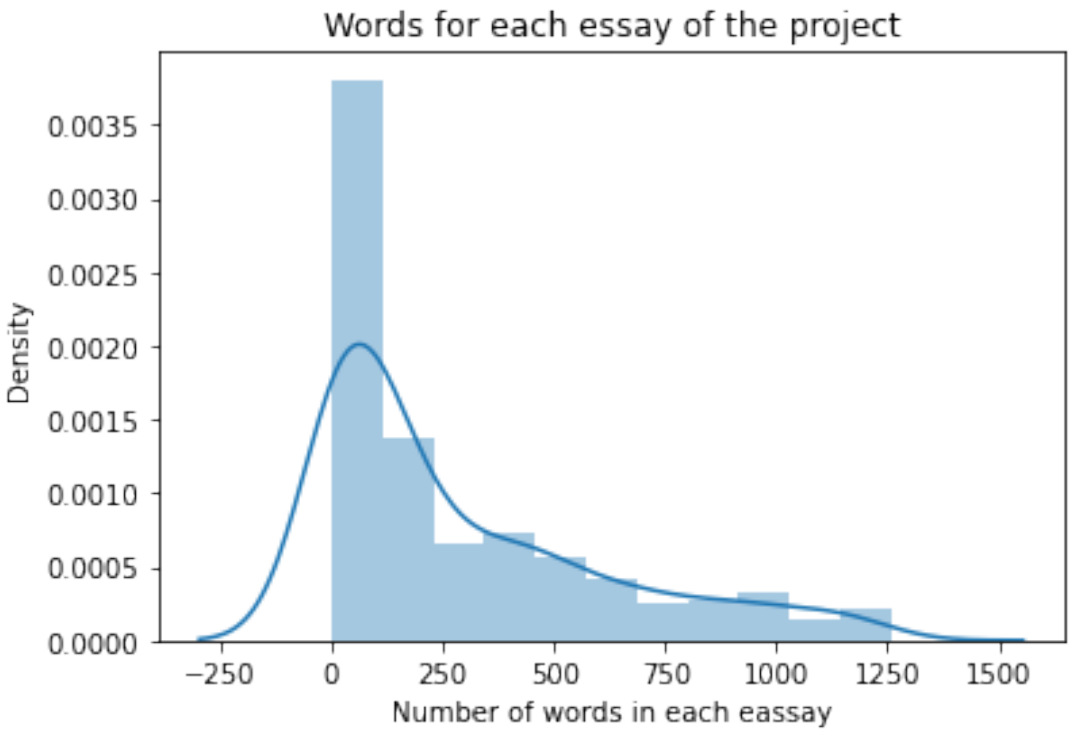


1.2.7 Univariate Analysis: Text features (Project Essay's)

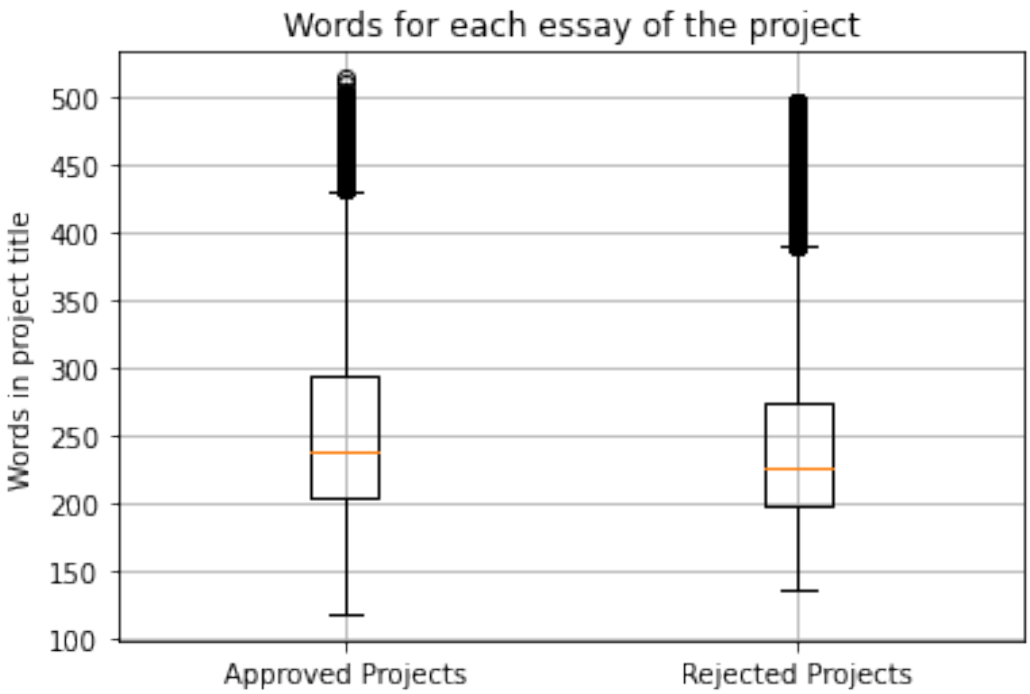
```
# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) +\
                        project_data["project_essay_2"].map(str) + \
                        project_data["project_essay_3"].map(str) + \
                        project_data["project_essay_4"].map(str)

#How to calculate number of words in a string in DataFrame:
https://stackoverflow.com/a/37483537/4084039
word_count =
project_data['essay'].str.split().apply(len).value_counts()
word_dict = dict(word_count)
word_dict = dict(sorted(word_dict.items(), key=lambda kv: kv[1]))


ind = np.arange(len(word_dict))
plt.figure(figsize=(20,5))
p1 = plt.bar(ind, list(word_dict.values()))

plt.ylabel('Number of projects')
plt.xlabel('Number of words in each eassay')
plt.title('Words for each essay of the project')
plt.xticks(ind, list(word_dict.keys()))
plt.show()
```



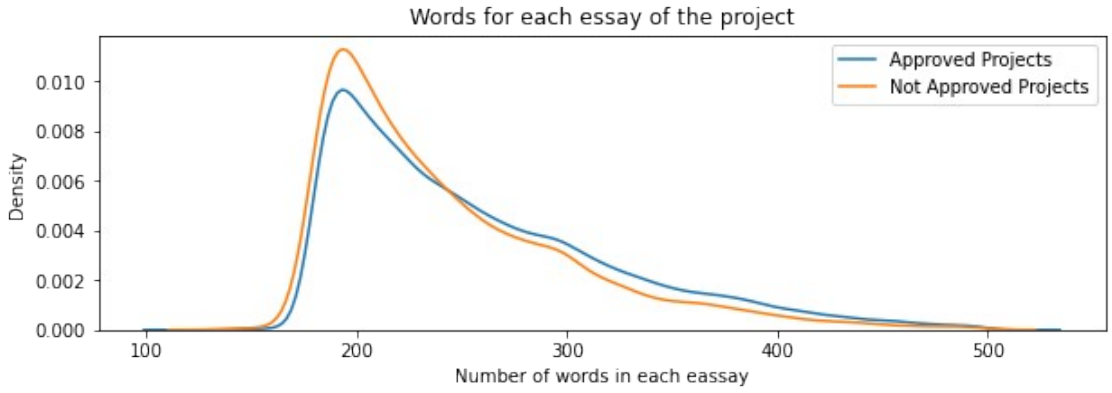

```
sns.distplot(word_count.values)
plt.title('Words for each essay of the project')
plt.xlabel('Number of words in each eassay')
plt.show()
```




```
approved_word_count =
project_data[project_data['project_is_approved']==1]
['essay'].str.split().apply(len)
approved_word_count = approved_word_count.values

rejected_word_count =
project_data[project_data['project_is_approved']==0]
['essay'].str.split().apply(len)
rejected_word_count = rejected_word_count.values

# https://glowingpython.blogspot.com/2012/09/boxplot-with-
matplotlib.html
plt.boxplot([approved_word_count, rejected_word_count])
plt.title('Words for each essay of the project')
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Words in project title')
plt.grid()
plt.show()
```




```
plt.figure(figsize=(10,3))
sns.distplot(approved_word_count, hist=False, label="Approved
Projects")
sns.distplot(rejected_word_count, hist=False, label="Not Approved
Projects")
plt.title('Words for each essay of the project')
plt.xlabel('Number of words in each eassay')
plt.legend()
plt.show()
```



1.2.8 Univariate Analysis: Cost per project

```
# we get the cost of the project using resource.csv file
resource_data.head(2)
```

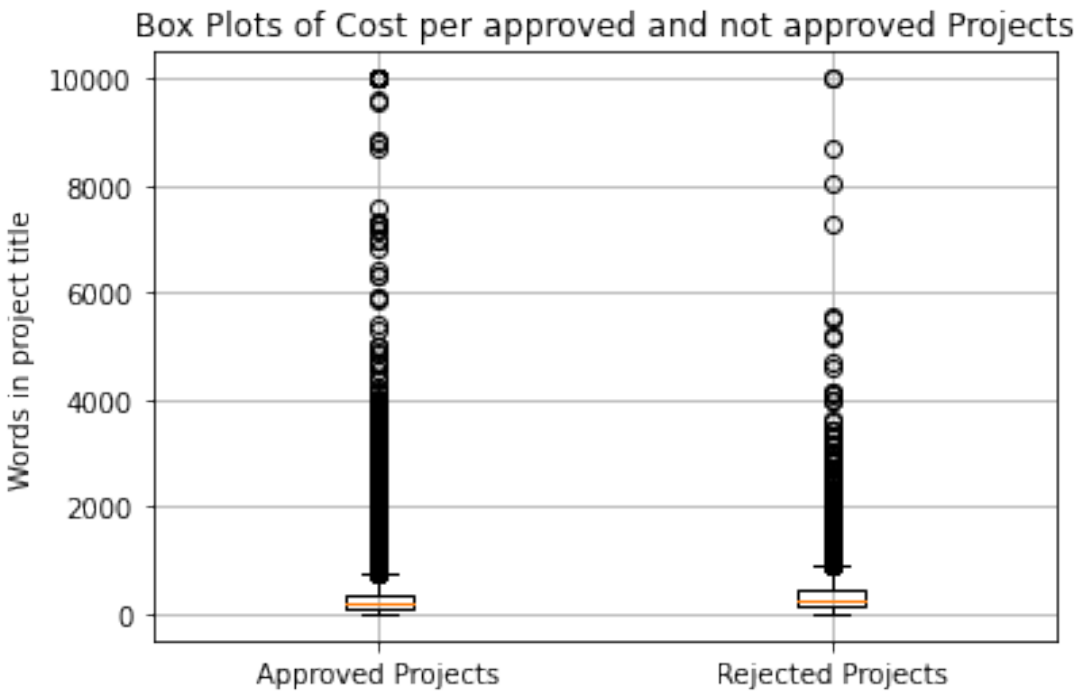
```
        id                                         description
quantity  \
```

```
0  p233245  LC652 - Lakeshore Double-Space Mobile Drying Rack
1
1  p069063         Bouncy Bands for Desks (Blue support pipes)
3


    price
0  149.00
1   14.95
```

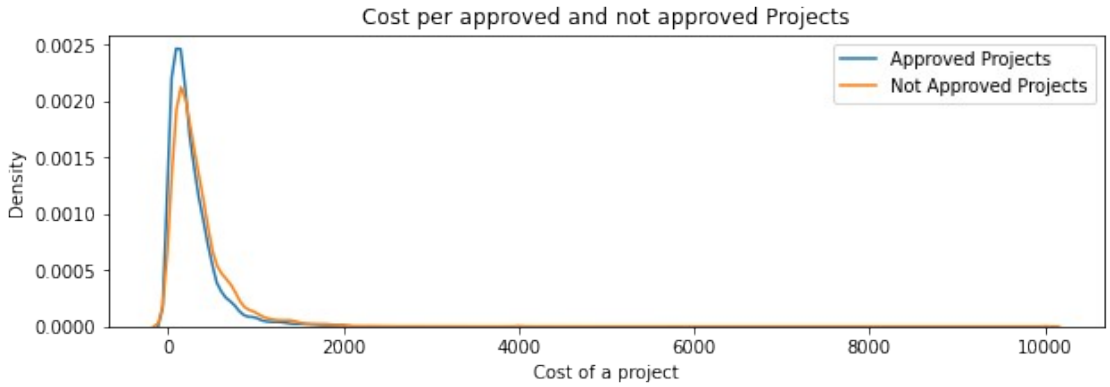
```
# https://stackoverflow.com/questions/22407798/how-to-reset-a-
dataframes-indexes-for-all-groups-in-one-step
price_data = resource_data.groupby('id').agg({'price':'sum',
'quantity':'sum'}).reset_index()
price_data.head(2)
```

```
        id   price  quantity
0  p000001  459.56         7
1  p000002  515.89        21
```

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

approved_price = project_data[project_data['project_is_approved']==1]
['price'].values

rejected_price = project_data[project_data['project_is_approved']==0]
['price'].values

# https://glowingpython.blogspot.com/2012/09/boxplot-with-
matplotlib.html
plt.boxplot([approved_price, rejected_price])
plt.title('Box Plots of Cost per approved and not approved Projects')
plt.xticks([1,2],('Approved Projects','Rejected Projects'))
plt.ylabel('Words in project title')
plt.grid()
plt.show()
```




```
plt.figure(figsize=(10,3))
sns.distplot(approved_price, hist=False, label="Approved Projects")
sns.distplot(rejected_price, hist=False, label="Not Approved
Projects")
plt.title('Cost per approved and not approved Projects')
plt.xlabel('Cost of a project')
plt.legend()
plt.show()
```




```
# http://zetcode.com/python/prettytable/
from prettytable import PrettyTable

table = PrettyTable()
table.field_names = ["Percentile", "Approved Projects", "Not Approved
Projects"]
```

```
for i in range(0,101,5):
    table.add_row([i,np.round(np.percentile(approved_price,i), 3),
np.round(np.percentile(rejected_price,i), 3)])
print(table)
```

```
+------------+-------------------+-----------------------+
| Percentile | Approved Projects | Not Approved Projects |
+------------+-------------------+-----------------------+
|     0      |        0.66       |          1.97         |
|     5      |       13.59       |          41.9         |
|     10     |       33.88       |         73.67         |
|     15     |        58.0       |         99.109        |
|     20     |       77.38       |         118.56        |
|     25     |       99.95       |        140.892        |
|     30     |       116.68      |         162.23        |
|     35     |      137.232      |        184.014        |
|     40     |       157.0       |        208.632        |
|     45     |      178.265      |        235.106        |
|     50     |       198.99      |        263.145        |
|     55     |       223.99      |         292.61        |
|     60     |       255.63      |        325.144        |
|     65     |      285.412      |         362.39        |
|     70     |      321.225      |         399.99        |
|     75     |      366.075      |        449.945        |
|     80     |       411.67      |        519.282        |
|     85     |       479.0       |        618.276        |
|     90     |       593.11      |        739.356        |
|     95     |      801.598      |        992.486        |
|    100     |       9999.0      |         9999.0        |
+------------+-------------------+-----------------------+
```