Donors Choose Dataset

- ***Data Pre-processing***
- ***Features Encoding***
- ***Export of pre-processed and encoded :: Train, CV and Test matrices***

Notebook Contents

Import_required_libraries

In [1]:

```
import logging
logging.basicConfig(filename="Donors_Choose.log",
                    filemode='w',
                    level=logging.INFO,
                    format="%(asctime)s : %(levelname)s : %(message)s")

try :
    logging.info("#### Packages import ####")
    ## Some basic libaries
    import os
    import sys
    import re          # Tutorial about Python regular expressions: https://pymotw.c
    import string
    import shutil
    import warnings
    import pickle
    import sqlite3
    from tqdm import tqdm
    from collections import Counter

    ## Data Pre-processing Libraries
    import numpy as np
    import pandas as pd

    ## Visualization Libraries
    import matplotlib.pyplot as plt
    from matplotlib.colors import ListedColormap

    ### Visualization :: Seaborn
    import seaborn as sns

    ### Visualization :: Plotly
    from chart_studio import plotly
    import plotly.offline as offline
    import plotly.graph_objs as go
    offline.init_notebook_mode()

    ## NLP
    import nltk

    ### NLP :: Stopwords
    from nltk.corpus import stopwords

    ### NLP :: Stemmer and Lemmatizer
    from nltk.stem.porter import PorterStemmer
    from nltk.stem import PorterStemmer
    from nltk.stem.wordnet import WordNetLemmatizer

    ### NLP :: Word2Vec
    from gensim.models import Word2Vec
    from gensim.models import KeyedVectors

    ### NLP :: Text Featurization libraries
    from sklearn.feature_extraction.text import TfidfTransformer
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.feature_extraction.text import CountVectorizer

    ## Features Scalers/Standardizers/Normalizers
    from sklearn.preprocessing import StandardScaler, MinMaxScaler, Normalizer

    ## Cross-Validation and Data Splitting
    from sklearn.model_selection import cross_val_score
    from sklearn.model_selection import train_test_split
```

```
    ## ML Algorithms
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.tree import DecisionTreeClassifier

    ## Performace Metrics
    from sklearn import metrics
    from sklearn.metrics import confusion_matrix, roc_curve, auc, precision_score, r
except ImportError as ie:
    # Output expected ImportErrors
    logging.error(msg=ie.__class__.__name__  + " :: Missing Package --> " + ie.name)
except Exception as exception:
    # Output unexpected Exceptions
    logging.info("#### Exceptions other than ModuleImportError ####")
    logging.log(msg=(exception, False))
    logging.log(msg=exception.__class__.__name__ + " :: " + exception.name)
%matplotlib inline
```

In [2]:
```
dc_train_df = pd.read_csv('Datasets/train_data.csv',index_col=0).reset_index(drop=Tr
dc_res_df = pd.read_csv('Datasets/resources.csv')
```

In [3]:
```
print("Number of data points in train data", (format(dc_train_df.shape[0],',d'),dc_t
print('-'*50)
print("The attributes of train data :", dc_train_df.columns.values)
```

```
Number of data points in train data ('109,248', 16)
--------------------------------------------------
The attributes of train data : ['id' 'teacher_id' 'teacher_prefix' 'school_state'
 'project_submitted_datetime' 'project_grade_category'
 'project_subject_categories' 'project_subject_subcategories'
 'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects' 'project_is_approved']
```

In [4]:
```
dc_train_df.head(4)
```

Out[4]:

	id	teacher_id	teacher_prefix	school_state	project_submitted_datet
0	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	2016-12-05 13:4
1	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	2016-10-25 09:2
2	p182444	3465aaf82da834c0582ebd0ef8040ca0	Ms.	AZ	2016-08-31 12:0
3	p246581	f3cb9bffbba169bef1a77b243e620b60	Mrs.	KY	2016-10-06 21:1

In [5]:
```
print("Number of data points in resource data", (format(dc_res_df.shape[0],',d'),dc_
print('-'*50)
```

```
print("The attributes of resource data :", dc_res_df.columns.values)
```

```
Number of data points in resource data ('1,541,272', 4)
--------------------------------------------------
The attributes of resource data : ['id' 'description' 'quantity' 'price']
```

In [6]:
```
dc_res_df.head()
```

Out[6]:

	id	description	quantity	price
0	p233245	LC652 - Lakeshore Double-Space Mobile Drying Rack	1	149.00
1	p069063	Bouncy Bands for Desks (Blue support pipes)	3	14.95
2	p069063	Cory Stories: A Kid's Book About Living With Adhd	1	8.45
3	p069063	Dixon Ticonderoga Wood-Cased #2 HB Pencils, Bo...	2	13.59
4	p069063	EDUCATIONAL INSIGHTS FLUORESCENT LIGHT FILTERS...	3	24.95

Preprocessing_Categorical_Features

Feature-1

- **project_grade_category**

In [7]:
```
dc_train_df['project_grade_category'].value_counts()
```

Out[7]:
```
Grades PreK-2    44225
Grades 3-5       37137
Grades 6-8       16923
Grades 9-12      10963
Name: project_grade_category, dtype: int64
```

In [8]:
```
dc_train_df['project_grade_category'] = dc_train_df['project_grade_category'].str.re
dc_train_df['project_grade_category'].value_counts()
```

Out[8]:
```
grades_prek_2    44225
grades_3_5       37137
grades_6_8       16923
grades_9_12      10963
Name: project_grade_category, dtype: int64
```

Feature-2

- **project_subject_categories**

In [9]:
```
dc_train_df['project_subject_categories'].value_counts()
```

Out[9]:
```
Literacy & Language                           23655
Math & Science                                17072
Literacy & Language, Math & Science           14636
Health & Sports                               10177
Music & The Arts                               5180
Special Needs                                  4226
Literacy & Language, Special Needs             3961
Applied Learning                               3771
Math & Science, Literacy & Language            2289
Applied Learning, Literacy & Language          2191
History & Civics                               1851
Math & Science, Special Needs                  1840
Literacy & Language, Music & The Arts          1757
Math & Science, Music & The Arts               1642
Applied Learning, Special Needs                1467
History & Civics, Literacy & Language          1421
```

```
Health & Sports, Special Needs                1391
Warmth, Care & Hunger                         1309
Math & Science, Applied Learning              1220
Applied Learning, Math & Science              1052
Literacy & Language, History & Civics          809
Health & Sports, Literacy & Language           803
Applied Learning, Music & The Arts             758
Math & Science, History & Civics               652
Literacy & Language, Applied Learning          636
Applied Learning, Health & Sports              608
Math & Science, Health & Sports                414
History & Civics, Math & Science               322
History & Civics, Music & The Arts             312
Special Needs, Music & The Arts                302
Health & Sports, Math & Science                271
History & Civics, Special Needs                252
Health & Sports, Applied Learning              192
Applied Learning, History & Civics             178
Health & Sports, Music & The Arts              155
Music & The Arts, Special Needs                138
Literacy & Language, Health & Sports            72
Health & Sports, History & Civics               43
History & Civics, Applied Learning              42
Special Needs, Health & Sports                  42
Special Needs, Warmth, Care & Hunger            23
Health & Sports, Warmth, Care & Hunger          23
Music & The Arts, Health & Sports               19
Music & The Arts, History & Civics              18
History & Civics, Health & Sports               13
Math & Science, Warmth, Care & Hunger           11
Music & The Arts, Applied Learning              10
Applied Learning, Warmth, Care & Hunger         10
Literacy & Language, Warmth, Care & Hunger       9
Music & The Arts, Warmth, Care & Hunger          2
History & Civics, Warmth, Care & Hunger          1
Name: project_subject_categories, dtype: int64
```

In [10]:
```
dc_train_df['project_subject_categories'] = dc_train_df['project_subject_categories'].
apply(lambda val: str(re.sub('_+','__',str(re.sub('[^a-zA-Z]','_',val).replace('The'
```

In [11]:
```
dc_train_df['project_subject_categories'].value_counts()
```

Out[11]:
```
literacy__language                          23655
math__science                               17072
literacy__language__math__science           14636
health__sports                              10177
music__arts                                  5180
special__needs                               4226
literacy__language__special__needs           3961
applied__learning                            3771
math__science__literacy__language            2289
applied__learning__literacy__language        2191
history__civics                              1851
math__science__special__needs                1840
literacy__language__music__arts              1757
math__science__music__arts                   1642
applied__learning__special__needs            1467
history__civics__literacy__language          1421
health__sports__special__needs               1391
warmth__care__hunger                         1309
math__science__applied__learning             1220
applied__learning__math__science             1052
literacy__language__history__civics           809
health__sports__literacy__language            803
applied__learning__music__arts                758
math__science__history__civics                652
literacy__language__applied__learning          636
applied__learning__health__sports              608
```

```
math__science__health__sports                 414
history__civics__math__science                322
history__civics__music__arts                  312
special__needs__music__arts                   302
health__sports__math__science                 271
history__civics__special__needs               252
health__sports__applied__learning             192
applied__learning__history__civics            178
health__sports__music__arts                   155
music__arts__special__needs                   138
literacy__language__health__sports             72
health__sports__history__civics                43
history__civics__applied__learning             42
special__needs__health__sports                 42
health__sports__warmth__care__hunger           23
special__needs__warmth__care__hunger           23
music__arts__health__sports                    19
music__arts__history__civics                   18
history__civics__health__sports                13
math__science__warmth__care__hunger            11
applied__learning__warmth__care__hunger        10
music__arts__applied__learning                 10
literacy__language__warmth__care__hunger        9
music__arts__warmth__care__hunger               2
history__civics__warmth__care__hunger           1
Name: project_subject_categories, dtype: int64
```

Feature-3

- **project_subject_subcategories**

In [12]:
```
dc_train_df['project_subject_subcategories'].value_counts().head(50)
```

Out[12]:
```
Literacy                                      9486
Literacy, Mathematics                         8325
Literature & Writing, Mathematics             5923
Literacy, Literature & Writing                5571
Mathematics                                   5379
Literature & Writing                          4501
Special Needs                                 4226
Health & Wellness                             3583
Applied Sciences, Mathematics                 3399
Applied Sciences                              2492
Literacy, Special Needs                       2440
Gym & Fitness, Health & Wellness              2264
ESL, Literacy                                 2234
Visual Arts                                   2217
Music                                         1472
Warmth, Care & Hunger                         1309
Literature & Writing, Special Needs           1306
Gym & Fitness                                 1195
Health & Wellness, Special Needs              1189
Mathematics, Special Needs                    1187
Environmental Science                         1079
Team Sports                                   1061
Applied Sciences, Environmental Science        984
Environmental Science, Health & Life Science   964
Music, Performing Arts                         948
Early Development                              905
Environmental Science, Mathematics             838
Other                                          831
Health & Life Science                          827
Health & Wellness, Nutrition Education         797
Early Development, Special Needs               779
ESL, Literature & Writing                      731
Early Development, Literacy                    717
Literature & Writing, Visual Arts              684
Applied Sciences, Visual Arts                  623
```

```
History & Geography, Literature & Writing        596
Gym & Fitness, Team Sports                       588
Applied Sciences, Health & Life Science          573
Applied Sciences, Literacy                       549
Literacy, Visual Arts                            545
History & Geography                              540
Health & Life Science, Mathematics               537
History & Geography, Literacy                    533
Mathematics, Visual Arts                         489
Health & Wellness, Literacy                      465
Environmental Science, Literacy                  444
ESL                                              421
College & Career Prep                            421
Applied Sciences, Literature & Writing           420
Applied Sciences, College & Career Prep          405
Name: project_subject_subcategories, dtype: int64
```

In [13]:
```
dc_train_df['project_subject_subcategories'] = dc_train_df['project_subject_subcateg
apply(lambda val: str(re.sub('_+','__',str(re.sub('[^a-zA-Z]','_',val).replace('The'
```

In [14]:
```
dc_train_df['project_subject_subcategories'].value_counts().head(50)
```

Out[14]:
```
literacy                                         9486
literacy__mathematics                            8325
literature__writing__mathematics                 5923
literacy__literature__writing                    5571
mathematics                                      5379
literature__writing                              4501
special__needs                                   4226
health__wellness                                 3583
applied__sciences__mathematics                   3399
applied__sciences                                2492
literacy__special__needs                         2440
gym__fitness__health__wellness                   2264
esl__literacy                                    2234
visual__arts                                     2217
music                                            1472
warmth__care__hunger                             1309
literature__writing__special__needs              1306
gym__fitness                                     1195
health__wellness__special__needs                 1189
mathematics__special__needs                      1187
environmental__science                           1079
team__sports                                     1061
applied__sciences__environmental__science         984
environmental__science__health__life__science     964
music__performing__arts                           948
early__development                                905
environmental__science__mathematics               838
other                                             831
health__life__science                             827
health__wellness__nutrition__education            797
early__development__special__needs                779
esl__literature__writing                          731
early__development__literacy                      717
literature__writing__visual__arts                 684
applied__sciences__visual__arts                   623
history__geography__literature__writing           596
gym__fitness__team__sports                        588
applied__sciences__health__life__science          573
applied__sciences__literacy                       549
literacy__visual__arts                            545
history__geography                                540
health__life__science__mathematics                537
history__geography__literacy                      533
mathematics__visual__arts                         489
health__wellness__literacy                        465
environmental__science__literacy                  444
```

```
college__career__prep                              421
esl                                                421
applied__sciences__literature__writing             420
applied__sciences__college__career__prep           405
Name: project_subject_subcategories, dtype: int64
```

Feature-4

- **teacher_prefix**

In [15]:
```
dc_train_df['teacher_prefix'].value_counts()
```

Out[15]:
```
Mrs.       57269
Ms.        38955
Mr.        10648
Teacher     2360
Dr.           13
Name: teacher_prefix, dtype: int64
```

In [16]:
```
dc_train_df['teacher_prefix'] = dc_train_df['teacher_prefix'].str.replace('.','').st
```

In [17]:
```
dc_train_df['teacher_prefix'].value_counts()
```

Out[17]:
```
mrs        57269
ms         38955
mr         10648
teacher     2360
dr            13
Name: teacher_prefix, dtype: int64
```

In [18]:
```
## only for 3 records
dc_train_df['teacher_prefix'].fillna(value='mrs',inplace=True)
```

In [19]:
```
dc_train_df['teacher_prefix'].value_counts()
```

Out[19]:
```
mrs        57272
ms         38955
mr         10648
teacher     2360
dr            13
Name: teacher_prefix, dtype: int64
```

Feature-5

- **school_state**

In [20]:
```
dc_train_df['school_state'].value_counts().head(50)
```

Out[20]:
```
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
Name: school_state, dtype: int64
```

In [21]:
```
dc_train_df['school_state'] = dc_train_df['school_state'].str.lower()
```

In [22]:
```
dc_train_df['school_state'].value_counts().head(50)
```

Out[22]:
```
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
Name: school_state, dtype: int64
```

Feature-6

- **project_title**

In [23]:
```
pd.set_option('display.max_colwidth',500)
```

In [24]:
```
dc_train_df['project_title'].head(50)
```

Out[24]:
```
0                                                                  Educational Support for English Learners at Home
1                                                                      Wanted: Projector for Hungry Learners
2                                                          Soccer Equipment for AWESOME Middle School Students
3                                                                                  Techie Kindergarteners
4                                                                                  Interactive Math Tools
5                                                  Flexible Seating for Mrs. Jarvis' Terrific Third Graders!!
6                                                           Chromebooks for Special Education Reading Program
7                                                                                   It's the 21st Century
8                                                                              Targeting More Success in Class
9                                                            Just For the Love of Reading--\r\nPure Pleasure
10                                                                                 Reading Changes Lives
11                                                 Elevating Academics and Parent Rapports Through Technology
12                                                                          Building Life Science Experiences
13                                                                             Everyone deserves to be heard!
14                                                                           TABLETS CAN SHOW US THE WORLD
15                                                                                    Making Recess Active
16                                                                         Making Great LEAP's With Leapfrog!
17                                                              Technology Teaches Tomorrow's Talents Today
18                                                                                                 Test Time
19                                                                              Wiggling Our Way to Success
20                                                                          Magic Carpet Ride in Our Library
21                                                                 From Sitting to Standing in the Classroom
22                                                                          Books for Budding Intellectuals
23                                                                       Instrumental Power: Conquering STEAM!
24                                                S.T.E.A.M. Challenges(Science Technology Engineering ART Math)
25                                                                                          Math Masters!
26                                                                                         Techy Teaching
27                                                                      4th Grade French Immersion Class Ipads
28                                                                              Hands-On Language and Literacy
29                                                                          Basic Classroom Supplies Needed
30                                                              2nd Grade Explores the World of Charlotte's Web
31                                                                          An All Inclusive Learning Space
32                                                                              Learning Facts From Fiction
33                                                           Computing Our Way to Financial Literacy! Part 2
34                                                                                    \"Have A Ball!!!\"
35                                                                                       Put Me In Coach
36                                                           Inquiry Based Discovery Through Laptop Learning
37                                                                    Target Our Kids With A Printer And Ink
```

```
38                        Kinders Inspired to be on Target in Fitness Part One
39                                     Engaging Students through Technology
40                                      Leveling Books in a Multi-Age Class
41                           A Twist for Writing Traits for My First Graders
42                                                      We Need Non-Fiction!
43                                                        All Hands on Tech!
44                                    Pressing on to Mastery After the Flood
45    Chromebooks Create Intrigue And Motivation While Filling In The Gaps!!
46                                                          All Out of Paper!
47                                                       Keep Our Closet Open
48                                                      Chromebook's Are Gold
49                                                      Rainy Day Run Around!
Name: project_title, dtype: object
```

In [25]:
```
pd.set_option('display.max_colwidth',50)
```

In [26]:
```
eng_stopwords = stopwords.words('english')
```

In [27]:
```
eng_stopwords.remove('no')
eng_stopwords.remove('nor')
eng_stopwords.remove('not')
```

In [28]:
```
print(eng_stopwords)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "yo
u've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'h
is', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itse
lf', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'who
m', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were',
'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing',
'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of',
'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'durin
g', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'o
n', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'wh
en', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'ot
her', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't',
'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm',
'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "did
n't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'i
sn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 's
han', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won',
"won't", 'wouldn', "wouldn't"]
```

In [29]:
```
rand_proj_ttls_idx = np.random.randint(0,109248,20)
print("**** printing some random project titles ****\n")
for idx in rand_proj_ttls_idx:
    print(idx,'::',dc_train_df['project_title'].values[idx])
```

```
**** printing some random project titles ****

97481 :: Technology Makes Learning Fun!
96479 :: Write Music Through Technology
48024 :: Listen to Read and Learn
50685 :: Temple Blocks go tick tock! (Music Education)
6371 :: Supplying The Tank
55246 :: Scholastic News in First Grade!
78079 :: Color me happy!
74713 :: Power Up with Solar Energy!
101101 :: A Classroom Without Desks, Oh My!
65072 :: Flexible Seating for a Student-Centered Classroom
57261 :: Kindergartners Need More Paint!
57880 :: Back to the Basics!
42113 :: Connecting Content and Culture through Reading
29983 :: Need pencils, audio, and supplies.
24576 :: Chevron: iRead with iPads!
4325 :: Take the Learning Home
67446 :: A Chromebook to Motivate! Part 5
```

```
94572 :: Spanish Classroom Resources: Flexible Seating
12127 :: On Task with Task Cards
42799 :: Individualized Technological Learning
```

In [30]:
```
# https://stackoverflow.com/a/47091490/4084039
def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)

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

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

In [31]:
```
preprocessed_titles = preprocess_text(dc_train_df['project_title'].values)
```

```
100%|████████████████████████████████████████████████████████████████████████| 109248/109248 [00:09<00:00, 12043.15it/s]
```

In [32]:
```
rand_proj_ttls_idx = np.random.randint(0,109248,20)
print("**** printing some random project titles ****\n")
for idx in rand_proj_ttls_idx:
    print(idx,'::',preprocessed_titles[idx])
print(68453,'::',dc_train_df['project_title'].iloc[68453])
print(68453,'::',preprocessed_titles[68453])
```

```
**** printing some random project titles ****

12082 :: literacy technology
74558 :: clear instruction
89759 :: rockin liberty
35511 :: life essentials
62992 :: flexible seating kindergarten learning lounge
34078 :: help students autism physical education equipment
39033 :: boogie boards boogie writing
67690 :: successful seating learning lead
64262 :: project art
7429 :: super charged
39721 :: creating cozy reading spot
41699 :: launching lunar satellites help eagles land moon
49556 :: bridging learning gap technology
56326 :: bookcases needed part 2
6824 :: galore math games
68861 :: community built rug
66743 :: let build word
```

```
105207 :: purposeful movement kindergarten
87258 :: fidget wobble succeed
70625 :: timeless tablets
68453 :: Can You Here Me Now?
68453 ::
```

In [33]:
```
dc_train_df['pp_titles'] = preprocessed_titles
```

In [34]:
```
dc_train_df.head(4)
```

Out[34]:

	id	teacher_id	teacher_prefix	school_state	project_submitted_datet
0	p253737	c90749f5d961ff158d4b4d1e7dc665fc	mrs	in	2016-12-05 13:4
1	p258326	897464ce9ddc600bced1151f324dd63a	mr	fl	2016-10-25 09:2
2	p182444	3465aaf82da834c0582ebd0ef8040ca0	ms	az	2016-08-31 12:0
3	p246581	f3cb9bffbba169bef1a77b243e620b60	mrs	ky	2016-10-06 21:1

Feature-7

- **project_essays**

In [35]:
```
# merge two column text dataframe:
dc_train_df["essay"] = dc_train_df["project_essay_1"].map(str) +\
                       dc_train_df["project_essay_2"].map(str) + \
                       dc_train_df["project_essay_3"].map(str) + \
                       dc_train_df["project_essay_4"].map(str)
```

In [36]:
```
rand_proj_ttls_idx = np.random.randint(0,109248,5)
print("**** printing some random essay ****\n")
for idx in rand_proj_ttls_idx:
    print(idx,'::',dc_train_df['essay'].values[idx])
    print('--'*50)
```

```
**** printing some random essay ****

107705 :: My students this year are a very lively bunch. They have a great sense of
classroom community and work hard to live up to their potential! We are a very diver
se group of learners, with 50% being English Language Learners speaking 6 different
languages. Students in my class span 8 ethnicities, varieties of living arrangements
from shelters to apartments and from rehab centers to houses. 80% of these students
also qualify for free/reduced lunch, having low income/high poverty households. Desp
ite the barriers placed upon these students, they show up every day with big smiles
on their faces. However, they don't always have what they need to be successful all
day. Any contribution to our classroom will greatly help these students thrive acade
mically and socially. They will get reinforcement that they are important, that they
do matter to others.My students are in need of level A-C books because I have severa
l readers who continue to struggle with accessing texts. While we are fortunate enou
```

gh to have a classroom library with many district supplied, donated, and scavenged b
ooks, the most struggling readers are not able to access the text in the majority of
these books. \r\nThese students have already read all of the A -C books in our libra
ry and need fresh content to motivate their learning. Providing these students with
books that they can read independently and confidently helps them start to see thems
elves as readers. This is essential to their growth, both in reading and in self con
cept.nannan

40073 :: My students are third graders from a multitude of different ethnic and econ
omic backgrounds. All of them have their own way of learning. Some are quiet, some
active, some shy, some attentive, some inattentive but all amazing in their own righ
t! I have, like all teachers, students who work better standing or laying on a carp
er or in a nook all on their own. I have students who need to bounce, wiggle, sway
or just sit still. \r\nMy students spend 6.5 hours a day in school, which is the ma
jority of their waking day. Flexible seating fosters independence. They can work a
t their highest potential when they feel \"at home\" and have freedom to choose wher
e they learn best.\r\nToday's student is asked to do much more than 20 years, even 1
0 years ago. The curriculum is much more rigorous and strives to get all students c
ollege and career ready. However, students on the elementary level tend to lack the
self regulation skills needed to sustain the level of attention needed. It is the t
eachers job to provide tools for student success. \r\n By providing flexible seating
in my classroom, I am allowing students to learn however they are comfortable. Stan
ding, wiggling, bouncing and rocking provide students with the movement their young
bodies need while attending to lessons and work. My students will become self awar
e, independent individuals with these donated materials.nannan
---

90228 :: I am so excited to start this school year off. We have an amazing school bo
dy that deserves something new and special this year. So what better way to start th
e year, then stepping out of ones own comfort zone.\r\n\r\nWe want to hit the ground
with both feet and running full speed ahead with engaging and real-world experiences
for our students.\r\n\r\nOur model school receives federal funding for programs that
provide free meals to every student and Title I Reading and Mathematics services to
economically disadvantaged boys and girls. We are a diverse school serving gifted, E
SL, exceptional, and tech savvy students. Our school body is very energetic and alwa
ys striving for excellence. Even with parental support, we would love to say that ou
r students have everything that they need to be successful, unfortunately that is no
t always possible. As a result, we continue to work with the community to prepare ou
r students for college, the workforce, and the world beyond.\r\nEvery year our stude
nts that part in a year long program that helps them to build their problem-solving
skills and work through word problems. But in the past I have found out that some st
udents need a little more help comprehending those word problems. So this year, I wo
uld like to take advantage of their love of technology and request an iPad Mini with
AppleCare and protective case to use for developing their problem-solving ability.
\r\n\r\nIt is not always easy to find something to grab students' attention, but I h
ave found several Apps that will help be to spark their interest in Mathematics. Wit
h you support of this project, I feel that we will have a hit on our hands and more
names displayed on Math Whiz Blvd.nannan

63040 :: How can we close the achievement gap without quality classroom books? Thou
gh many of my students come from supportive homes, there are some students who are b
used in from low-income neighborhoods and their only exposure to books is at school.
\r\n\r\nMy diverse 4th graders enjoy read-alouds, learning about new books and meeti
ng in their book clubs. Nobody asks the basketball coach to buy basketballs, but th
e classroom teacher is responsible for supplying classroom books. It is imperative
that my students are exposed to more books in our class.Basketball coaches are not a
sked to buy basketballs for PE, yet classroom teachers are not provided with quality
classroom libraries. Did you know that classroom libraries should include 600-1000
books per class? My students love to read and providing them with new books for the
ir book clubs, read alouds and independent learning is key to their success.\r\n\r\n
I want my students to get lost in books and discover new friendships, new places and
new ways of life. I can help lead them in the right direction by introducing them t
o great books that include life and social skills, diversity and more. Please consi
der helping our class read our way through 4th grade!nannan

98485 :: I have been teaching for going on eleven years at a low-income/high poverty

```
school. My students are faced with many challenges in and outside of school. When th
ey enter my classroom, I want them to be able to leave their problems at the door an
d learn in a safe, comfortable setting that meets all their needs.  \r\n\r\nMy stude
nts are our future leaders, citizens and neighbors! I want to expose them to as much
useful resources as possible that will help them become productive, helpful, and suc
cessful leaders in the future.  \r\n\r\nOur class has been practicing mindfulness te
chniques to deal with stress and anxiety that sadly affects many of my students dail
y. \r\n\r\nYoga has been proven to naturally release stress and anxiety which makes
learning so much easier!  I want my students to feel calm and happy while they are a
t school.  Practicing yoga techniques can help them learn calming moves, and breathi
ng techniques to deal with complex emotions throughout the day. \r\n\r\nWe will use
the yoga mats after recess or between specific transitions during the day so student
s are calm, focused, and ready to learn.  My goal is for my students to love school
and get all their needs met.  They can't focus on academics if they feel stress insi
de. \r\nnannan
--------------------------------------------------------------------------------
----------
```

In [37]:
```
preprocessed_essays = preprocess_text(dc_train_df['essay'].values)
```

```
100%|
| 109248/109248 [03:23<00:00, 538.13it/s]
```

In [38]:
```
rand_proj_ttls_idx = np.random.randint(0,109248,3)
print("**** printing some random project titles ****\n")
for idx in rand_proj_ttls_idx:
    print(idx,'::',preprocessed_essays[idx])
    print('--'*50)
```

```
**** printing some random project titles ****

44382 :: students ethnically diverse naturally talented say students caribbean islan
ds africa central america south america academically behind average work hard core s
ubjects school building 100 years old outside newark nj not auditorium students jump
chance perform students enter room ready release energy vocally physically provide w
arm environment even shy students willing participate students new country english l
anguage become comfortable almost immediately non verbal exercises knowledgable staf
f helpful classmates students soon excel aspects class students rehearsing upcoming
production cinderella almost ready need costumes younger students older students pla
ying step sisters king knights need racks hanging costumes divider using onstage set
piece stage hide actors students passionate drama cannot wait show community culmina
tion hard work not able use supplies stage play year use years come
--------------------------------------------------------------------------------
----------
43123 :: students curious adventurous part positive learning community second grader
s urban public school district title one school one hundred percent free lunch stude
nts love learn grow together want give every opportunity goal teacher help evolve su
cceed aspects lives learning confident reader able solve problems friends courage tr
y something new students empowered make best decisions throughout education lives tw
o classes twenty four students class spends half day learning read write spell discu
ss literature well explore science social studies passionate educator trying foster
classroom passionate learners 2nd grade students learning importance writing process
learning pre write draft edit revise publish writing final draft students would grea
tly benefit writing portfolio save published writing throughout year instead final d
rafts put folder somewhere hung bulletin board weeks proudly publish adding writing
portfolio students feel like published authors proudly display hard work writing por
tfolios portfolio divided different sections including informational writing narrati
ves opinion writing students opportunities share portfolios peers parents watch writ
ing greatly improves school year project provide class 42 binders sets dividers stud
ent create writing porfolio project also provide classroom enough pencils last entir
e year would surprised many pencils 7 year old go
--------------------------------------------------------------------------------
----------
97567 :: school located camden nj school large hispanic population contains large en
glish learner population majority students qualify free breakfast lunch school class
energetic creative respectful make every day adventure year class heavy focus develo
ping literacy spend many minutes day becoming better readers writers many students n
ot gained love reading hope helps develop new passion reading realize common majorit
y students new united states mostly come south america taking first steps learning e
```

```
nglish language students need many things including uniform shirts school students n
eed biggest distraction student also embarrassment come school dirty shirt students
embarrassed unable focus lessons day self conscious embarrassed circumstances would
like change making sure students classroom access clean shirt need use students woul
d know shirts access feel pride comfort appearance
```

In [39]:
```
dc_train_df['pp_essays'] = preprocessed_essays
```

In [40]:
```
dc_train_df.head(4)
```

Out[40]:

	id	teacher_id	teacher_prefix	school_state	project_submitted_datet
0	p253737	c90749f5d961ff158d4b4d1e7dc665fc	mrs	in	2016-12-05 13:4
1	p258326	897464ce9ddc600bced1151f324dd63a	mr	fl	2016-10-25 09:2
2	p182444	3465aaf82da834c0582ebd0ef8040ca0	ms	az	2016-08-31 12:0
3	p246581	f3cb9bffbba169bef1a77b243e620b60	mrs	ky	2016-10-06 21:1

Feature-8

- **project_costs**

In [41]:
```
price_data = dc_res_df.groupby('id').agg({'price':'sum', 'quantity':'sum'}).reset_in
price_data.head()
```

Out[41]:

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

In [42]:
```
project_data = pd.merge(dc_train_df, price_data, on='id', how='left')
```

In [43]:
```
project_data['price'].head()
```

Out[43]:
```
0    154.60
1    299.00
2    516.85
3    232.90
4     67.98
Name: price, dtype: float64
```

In [44]:
```
project_data.head(4)
```

Out[44]:

	id	teacher_id	teacher_prefix	school_state	project_submitted_datet
0	p253737	c90749f5d961ff158d4b4d1e7dc665fc	mrs	in	2016-12-05 13:4
1	p258326	897464ce9ddc600bced1151f324dd63a	mr	fl	2016-10-25 09:2
2	p182444	3465aaf82da834c0582ebd0ef8040ca0	ms	az	2016-08-31 12:0
3	p246581	f3cb9bffbba169bef1a77b243e620b60	mrs	ky	2016-10-06 21:1

4 rows × 21 columns

In [45]:
```
project_data[project_data['pp_titles']==''].shape[0]
```

Out[45]:
```
43
```

In [46]:
```
project_data[project_data['pp_titles']=='']['project_title']
```

Out[46]:
```
880                                        I CAN do it!
3517                     Who, What, When, Where, How?
7503                                          Who I Am
9019                Who? What? Where? When? Why? How?
9518                       This IS How You Do It!
12390                        This Is How We Do It
15617                              I Can Do It!!
17652                      I Can Do It By Myself!
17798                         Who is?! What is?!
18915                       I Do. We Do. You Do.
19122                I do it. We do it. You do it.
19731                          Me, Myself and I!
19772                                    I Won!
```

```
21484                                        We Can Do It!
24024                        Who Was...Where Was...What Is...
25089                                   I Can do it Myself!
29208     WHO Was That? WHAT Was That? WHERE Is That? WHY?
32089                              OVER UNDER: You can do it!
32608                                       WE CAN DO IT TOO!
34091                                         Can It Do That?
41656                               *I Do, We Do, You Do!!**
41797                                              Who Is It?
45844                                       We Can Do This!
50004                                         We Can Do It!
50526                                          I Can Do It!
54464                                       Are You With Me?
54781                                      We Can Do It Too!
56857                                         All By Myself!
57432                                    I Can Do It Myself!
59766                                           We Can Do It
63032                  From There to Here and Here to There
68453                                   Can You Here Me Now?
68821                                          Out and About
73774                                             We Did It!
74264                                      To Be Is Up to Me
77851                                   We Can, and We Will!
80909                                         Over and Under
81581                                       This Just In!!!
93213                               Just Between You and Me!
94369                                         What Was That?
101144                                 Under Over Under Over
102430                                                I Can
106760                                       Where Are You?
Name: project_title, dtype: object
```

In [47]:
```
project_data[project_data['pp_titles']=='']['project_is_approved'].value_counts()
```

Out[47]:
```
1    26
0    17
Name: project_is_approved, dtype: int64
```

Feature-9

- **project_resource_summary**

In [48]:
```
rand_proj_ttls_idx = np.random.randint(0,109248,12)
print("**** printing some random resource summary ****\n")
for idx in rand_proj_ttls_idx:
    print(idx,'::',project_data['project_resource_summary'].values[idx])
    print('-'*50)
```

```
**** printing some random resource summary ****

27332 :: My students need vocal booth boxes to assist them with their technology pro
jects.
--------------------------------------------------
2498 :: My students need two Alphabetter Stand-Up Desks in order to improve their le
arning.
--------------------------------------------------
83275 :: My students need a subscription to Scholastic News Magazine. This magazine
has wonderfully engaging articles that inspire students to learn more about their wo
rld.
--------------------------------------------------
44882 :: My students need whiteboard markers, pencils and sharpeners, copy paper, fl
ip chart paper in order to Empower my Students through Art.
--------------------------------------------------
83151 :: My students need ipads in the classroom to enhance and challenge their lear
ning in all subjects.
--------------------------------------------------
25923 :: My students need alternative seating arrangement areas that are comfortabl
e.  Learning is easier when you are comfortable!
```

```
--------------------------------------------------
99298 :: My students need paper to create poster-size images of their original artwo
rk. These students are part of my Advanced Placement Studio Art year-long class.
--------------------------------------------------
908 :: My students need to have many different ways to learn. Each student has some
type of sensory deficit due to their disability of Autism.
--------------------------------------------------
67778 :: My students need guided reading sets in our classroom to build our library
for teaching and learning. This will help me support students as they move up levels
by having engaging, authentic texts at our fingertips!
--------------------------------------------------
9268 :: My students need adjustable height basketball goals so all grades kindergart
en through 5th grade can benefit from this equipment.
--------------------------------------------------
50402 :: My students need hands-on models to improve their comprehension and memoriz
ation of human anatomy and physiology.
--------------------------------------------------
19661 :: My students need a class sets of books, an owl pellet Study Classroom kit t
o guide in the research of owls and environmental issues they may be facing, and pri
nter ink .
--------------------------------------------------
```

In [49]:
```
preprocessed_res_smry = preprocess_text(project_data['project_resource_summary'].val
```

```
100%|████████████████████████████████████████████████████████████████████████████| 109248/109248 [00:23<00:00, 4563.56it/s]
```

In [50]:
```
rand_proj_ttls_idx = np.random.randint(0,109248,13)
print("**** printing some random project titles ****\n")
for idx in rand_proj_ttls_idx:
    print(idx,'::',preprocessed_res_smry[idx])
    print('-'*50)
```

```
**** printing some random project titles ****

2793 :: students need tablets access personalized engaging math literacy programs li
ke lexia raz kids st math
--------------------------------------------------
86459 :: students need books selected variety books students lexile levels students
read abilities
--------------------------------------------------
38809 :: students need dependable device capable exploring world around us within co
nfines classroom walls
--------------------------------------------------
63726 :: students need beakers lab measure capacity liquids
--------------------------------------------------
93140 :: students need building blocks strengthen gross fine motor skills
--------------------------------------------------
18313 :: students need first aid soccer kit athletic tape goalie gloves team set zip
soccer warm ups equipment protect youth soccer team common soccer injuries
--------------------------------------------------
93968 :: students need 6 ecr4kids 14 stack chairs green matching legs
--------------------------------------------------
66986 :: students need cooking materials order able follow cooking curriculum
--------------------------------------------------
89470 :: students need several art supplies projects class books use resources mater
ial
--------------------------------------------------
84650 :: students need bright paper glue sticks labels storage container
--------------------------------------------------
87251 :: students need electronic tablets things make successful 21st century learni
ng environment
--------------------------------------------------
107589 :: students need large drying rack better equipped handle amount work produce
--------------------------------------------------
29114 :: students need blue dress shirts khaki pants black shoes meet school uniform
requirements
--------------------------------------------------
```

In [51]:
```
project_data['pp_res_smrys'] = preprocessed_res_smry
```

In [52]:
```
project_data.head(4)
```

Out[52]:

	id	teacher_id	teacher_prefix	school_state	project_submitted_datet
0	p253737	c90749f5d961ff158d4b4d1e7dc665fc	mrs	in	2016-12-05 13:4
1	p258326	897464ce9ddc600bced1151f324dd63a	mr	fl	2016-10-25 09:2
2	p182444	3465aaf82da834c0582ebd0ef8040ca0	ms	az	2016-08-31 12:0
3	p246581	f3cb9bffbba169bef1a77b243e620b60	mrs	ky	2016-10-06 21:1

4 rows × 22 columns

In [53]:
```
project_data[project_data['pp_res_smrys']==''].shape[0]
```

Out[53]:
```
0
```

In [54]:
```
project_data.columns
```

Out[54]:
```
Index(['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',
       'pp_titles', 'essay', 'pp_essays', 'price', 'quantity', 'pp_res_smrys'],
      dtype='object')
```

In [55]:
```
pp_data = project_data[['teacher_prefix','school_state','project_grade_category',
                        'project_subject_categories', 'project_subject_subcategories
                        'pp_titles', 'pp_essays', 'price', 'quantity', 'pp_res_smrys
```

In [56]:
```
pp_data.shape
```

Out[56]:
```
(109248, 12)
```

In [57]:
```
pp_data.head(4)
```

Out[57]:

	teacher_prefix	school_state	project_grade_category	project_subject_categories	project_su
0	mrs	in	grades_prek_2	literacy__language	
1	mr	fl	grades_6_8	history__civics__health__sports	civics__goverr
2	ms	az	grades_6_8	health__sports	health__we
3	mrs	ky	grades_prek_2	literacy__language__math__science	li

Splitting_Data

In [58]:
```
limited_pp_data = pp_data.iloc[0:60000,:].copy(deep=True)
```

In [59]:
```
y = limited_pp_data['project_is_approved'].values
X = limited_pp_data.drop(['project_is_approved'], axis=1)
print("X shape -->",X.shape,"y shape -->",y.shape)
X.head(3)
```

```
X shape --> (60000, 11) y shape --> (60000,)
```

Out[59]:

	teacher_prefix	school_state	project_grade_category	project_subject_categories	project_subject
0	mrs	in	grades_prek_2	literacy__language	
1	mr	fl	grades_6_8	history__civics__health__sports	civics__governmer

	teacher_prefix	school_state	project_grade_category	project_subject_categories	project_subject
2	ms	az	grades_6_8	health__sports	health__wellnes

Stratified_Sampling

- **Splitting data into Train and cross validation(or test)**

In [60]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, stratify=y
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.10, st
```

In [61]:
```
print("X_train shape --> ",X_train.shape)
print("y_train shape --> ",y_train.shape)
print('-'*20)
print("X_cv shape --> ",X_cv.shape)
print("y_cv shape --> ",y_cv.shape)
print('-'*20)
print("X_test shape --> ",X_test.shape)
print("y_test shape --> ",y_test.shape)
```

```
X_train shape -->  (40500, 11)
y_train shape -->  (40500,)
--------------------
X_cv shape -->  (4500, 11)
y_cv shape -->  (4500,)
--------------------
X_test shape -->  (15000, 11)
y_test shape -->  (15000,)
```

In [62]:
```
np.savetxt("y_train.csv", y_train, delimiter=",")
np.savetxt("y_cv.csv", y_cv, delimiter=",")
np.savetxt("y_test.csv", y_test, delimiter=",")
```

Making_Data_Model_Ready

- **Encoding project title, summary and essay**

In [63]:
```
limited_pp_data.columns
```

Out[63]:
```
Index(['teacher_prefix', 'school_state', 'project_grade_category',
       'project_subject_categories', 'project_subject_subcategories',
       'teacher_number_of_previously_posted_projects', 'pp_titles',
       'pp_essays', 'price', 'quantity', 'pp_res_smrys',
       'project_is_approved'],
      dtype='object')
```

In [64]:
```
X_test.columns
```

Out[64]:
```
Index(['teacher_prefix', 'school_state', 'project_grade_category',
       'project_subject_categories', 'project_subject_subcategories',
       'teacher_number_of_previously_posted_projects', 'pp_titles',
       'pp_essays', 'price', 'quantity', 'pp_res_smrys'],
      dtype='object')
```

Encoding-->Essays

Essay_BOW

In [65]:
```
vectorizer = CountVectorizer(min_df=20,ngram_range=(1,4),max_features=3000)
vectorizer.fit(X_train['pp_essays'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_essay_bow = vectorizer.transform(X_train['pp_essays'].values)
X_cv_essay_bow = vectorizer.transform(X_cv['pp_essays'].values)
X_test_essay_bow = vectorizer.transform(X_test['pp_essays'].values)

print("*** After vectorizations ***\n")
print("X-train Essay BOW {} and y-train {}".format(X_train_essay_bow.shape, y_train.
print("="*20)
print("X-cv Essay BOW {} and y-cv {}".format(X_cv_essay_bow.shape, y_cv.shape))
print("="*20)
print("X-test Essay BOW {} and y-test {}".format(X_test_essay_bow.shape, y_test.shap
print("="*20)
```

```
*** After vectorizations ***

X-train Essay BOW (40500, 3000) and y-train (40500,)
====================
X-cv Essay BOW (4500, 3000) and y-cv (4500,)
====================
X-test Essay BOW (15000, 3000) and y-test (15000,)
====================
```

Essay_TF-IDF

In [66]:
```
essay_tfidfvectorizer = TfidfVectorizer(min_df=20,ngram_range=(1,4),max_features=300
essay_tfidfvectorizer.fit(X_train['pp_essays'].values) # fit has to happen only on t

# we use the fitted CountVectorizer to convert the text to vector
X_train_essay_tfidf_vect = essay_tfidfvectorizer.transform(X_train['pp_essays'].valu
X_cv_essay_tfidf_vect = essay_tfidfvectorizer.transform(X_cv['pp_essays'].values)
X_test_essay_tfidf_vect = essay_tfidfvectorizer.transform(X_test['pp_essays'].values

print("*** After vectorizations ***\n")
print("X-train Essay Tf-Idf Vect {} and y-train {}".format(X_train_essay_tfidf_vect.
print("="*20)
print("X-cv Essay Tf-Idf Vect {} and y-cv {}".format(X_cv_essay_tfidf_vect.shape, y_
print("="*20)
print("X-test Essay Tf-Idf Vect {} and y-test {}".format(X_test_essay_tfidf_vect.sha
print("="*20)
```

```
*** After vectorizations ***

X-train Essay Tf-Idf Vect (40500, 3000) and y-train (40500,)
====================
X-cv Essay Tf-Idf Vect (4500, 3000) and y-cv (4500,)
====================
X-test Essay Tf-Idf Vect (15000, 3000) and y-test (15000,)
====================
```

Encoding-->Titles

Title_BOW

In [67]:
```
title_vectorizer = CountVectorizer(min_df=20,ngram_range=(1,4),max_features=200)
title_vectorizer.fit(X_train['pp_titles'].values) # fit has to happen only on train

# we use the fitted CountVectorizer to convert the text to vector
X_train_title_bow = title_vectorizer.transform(X_train['pp_titles'].values)
X_cv_title_bow = title_vectorizer.transform(X_cv['pp_titles'].values)
X_test_title_bow = title_vectorizer.transform(X_test['pp_titles'].values)
```

```
print("*** After vectorizations ***\n")
print("X-train Titles BOW {} and y-train {}".format(X_train_title_bow.shape, y_train
print("="*20)
print("X-cv Titles BOW {} and y-cv {}".format(X_cv_title_bow.shape, y_cv.shape))
print("="*20)
print("X-test Titles BOW {} and y-test {}".format(X_test_title_bow.shape, y_test.sha
print("="*20)
```

```
*** After vectorizations ***

X-train Titles BOW (40500, 200) and y-train (40500,)
====================
X-cv Titles BOW (4500, 200) and y-cv (4500,)
====================
X-test Titles BOW (15000, 200) and y-test (15000,)
====================
```

Title_TF-IDF

In [68]:
```
titles_tfidfvectorizer = TfidfVectorizer(min_df=20,ngram_range=(1,4),max_features=20
titles_tfidfvectorizer.fit(X_train['pp_titles'].values) # fit has to happen only on

# we use the fitted CountVectorizer to convert the text to vector
X_train_title_tfidf_vect = titles_tfidfvectorizer.transform(X_train['pp_titles'].val
X_cv_title_tfidf_vect = titles_tfidfvectorizer.transform(X_cv['pp_titles'].values)
X_test_title_tfidf_vect = titles_tfidfvectorizer.transform(X_test['pp_titles'].value

print("*** After vectorizations ***\n")
print("X-train Titles Tf-Idf Vect {} and y-train {}".format(X_train_title_tfidf_vect
print("="*20)
print("X-cv Titles Tf-Idf Vect {} and y-cv {}".format(X_cv_title_tfidf_vect.shape, y
print("="*20)
print("X-test Titles Tf-Idf Vect {} and y-test {}".format(X_test_title_tfidf_vect.sh
print("="*20)
```

```
*** After vectorizations ***

X-train Titles Tf-Idf Vect (40500, 200) and y-train (40500,)
====================
X-cv Titles Tf-Idf Vect (4500, 200) and y-cv (4500,)
====================
X-test Titles Tf-Idf Vect (15000, 200) and y-test (15000,)
====================
```

Encoding-->Resource_Summaries

Summary_BOW

In [69]:
```
summary_vectorizer = CountVectorizer(min_df=20,ngram_range=(1,4),max_features=800)
summary_vectorizer.fit(X_train['pp_res_smrys'].values) # fit has to happen only on t

# we use the fitted CountVectorizer to convert the text to vector
X_train_summary_bow = summary_vectorizer.transform(X_train['pp_res_smrys'].values)
X_cv_summary_bow = summary_vectorizer.transform(X_cv['pp_res_smrys'].values)
X_test_summary_bow = summary_vectorizer.transform(X_test['pp_res_smrys'].values)

print("*** After vectorizations ***\n")
print("X-train Summaries BOW {} and y-train {}".format(X_train_summary_bow.shape, y_
print("="*20)
print("X-cv Summaries BOW {} and y-cv {}".format(X_cv_summary_bow.shape, y_cv.shape)
print("="*20)
print("X-test Summaries BOW {} and y-test {}".format(X_test_summary_bow.shape, y_tes
print("="*20)
```

```
*** After vectorizations ***

X-train Summaries BOW (40500, 800) and y-train (40500,)
```

```
====================
X-cv Summaries BOW (4500, 800) and y-cv (4500,)
====================
X-test Summaries BOW (15000, 800) and y-test (15000,)
====================
```

Summary_TF-IDF

In [70]:
```
summary_tfidfvectorizer = TfidfVectorizer(min_df=20,ngram_range=(1,4),max_features=8
summary_tfidfvectorizer.fit(X_train['pp_res_smrys'].values) # fit has to happen only

# we use the fitted CountVectorizer to convert the text to vector
X_train_summary_tfidf_vect = summary_tfidfvectorizer.transform(X_train['pp_res_smrys
X_cv_summary_tfidf_vect = summary_tfidfvectorizer.transform(X_cv['pp_res_smrys'].val
X_test_summary_tfidf_vect = summary_tfidfvectorizer.transform(X_test['pp_res_smrys']

print("*** After vectorizations ***\n")
print("X-train Summary Tf-Idf Vect {} and y-train {}".format(X_train_summary_tfidf_v
print("="*20)
print("X-cv Summary Tf-Idf Vect {} and y-cv {}".format(X_cv_summary_tfidf_vect.shape
print("="*20)
print("X-test Summary Tf-Idf Vect {} and y-test {}".format(X_test_summary_tfidf_vect
print("="*20)
```

```
*** After vectorizations ***

X-train Summary Tf-Idf Vect (40500, 800) and y-train (40500,)
====================
X-cv Summary Tf-Idf Vect (4500, 800) and y-cv (4500,)
====================
X-test Summary Tf-Idf Vect (15000, 800) and y-test (15000,)
====================
```

Encoding-->Teacher_Prefix

In [71]:
```
t_prefix_vectorizer = CountVectorizer()
t_prefix_vectorizer.fit(X_train['teacher_prefix'].values) # fit has to happen only o

# we use the fitted CountVectorizer to convert the text to vector
X_train_teacher_prefix_ohe = t_prefix_vectorizer.transform(X_train['teacher_prefix']
X_cv_teacher_prefix_ohe = t_prefix_vectorizer.transform(X_cv['teacher_prefix'].value
X_test_teacher_prefix_ohe = t_prefix_vectorizer.transform(X_test['teacher_prefix'].v

print("*** After vectorizations ***\n")
print("X-train teacher prefix BOW {} and y-train {}".format(X_train_teacher_prefix_o
print("="*20)
print("X-cv teacher prefix BOW {} and y-cv {}".format(X_cv_teacher_prefix_ohe.shape,
print("="*20)
print("X-test teacher prefix BOW {} and y-test {}".format(X_test_teacher_prefix_ohe.
print("="*20)
print(t_prefix_vectorizer.get_feature_names())
print("="*20)
```

```
*** After vectorizations ***

X-train teacher prefix BOW (40500, 5) and y-train (40500,)
====================
X-cv teacher prefix BOW (4500, 5) and y-cv (4500,)
====================
X-test teacher prefix BOW (15000, 5) and y-test (15000,)
====================
['dr', 'mr', 'mrs', 'ms', 'teacher']
====================
```

Encoding-->School_State

In [72]:
```
ss_vectorizer = CountVectorizer()
ss_vectorizer.fit(X_train['school_state'].values) # fit has to happen only on train

# we use the fitted CountVectorizer to convert the text to vector
X_train_state_ohe = ss_vectorizer.transform(X_train['school_state'].values)
X_cv_state_ohe = ss_vectorizer.transform(X_cv['school_state'].values)
X_test_state_ohe = ss_vectorizer.transform(X_test['school_state'].values)

print("*** After vectorizations ***\n")
print("X-train School State BOW {} and y-train {}".format(X_train_state_ohe.shape, y
print("="*20)
print("X-cv School State BOW {} and y-cv {}".format(X_cv_state_ohe.shape, y_cv.shape
print("="*20)
print("X-test School State BOW {} and y-test {}".format(X_test_state_ohe.shape, y_te
print("="*20)
print(ss_vectorizer.get_feature_names())
print("="*20)
```

```
*** After vectorizations ***

X-train School State BOW (40500, 51) and y-train (40500,)
====================
X-cv School State BOW (4500, 51) and y-cv (4500,)
====================
X-test School State BOW (15000, 51) and y-test (15000,)
====================
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga', 'hi', 'ia', 'id',
'il', 'in', 'ks', 'ky', 'la', 'ma', 'md', 'me', 'mi', 'mn', 'mo', 'ms', 'mt', 'nc',
'nd', 'ne', 'nh', 'nj', 'nm', 'nv', 'ny', 'oh', 'ok', 'or', 'pa', 'ri', 'sc', 'sd',
'tn', 'tx', 'ut', 'va', 'vt', 'wa', 'wi', 'wv', 'wy']
====================
```

Encoding-->Project_Grade_Category

In [73]:
```
p_grade_cat_vectorizer = CountVectorizer()
p_grade_cat_vectorizer.fit(X_train['project_grade_category'].values) # fit has to ha

# we use the fitted CountVectorizer to convert the text to vector
X_train_p_grade_cat_ohe = p_grade_cat_vectorizer.transform(X_train['project_grade_ca
X_cv_p_grade_cat_ohe = p_grade_cat_vectorizer.transform(X_cv['project_grade_category
X_test_p_grade_cat_ohe = p_grade_cat_vectorizer.transform(X_test['project_grade_cate

print("*** After vectorizations ***\n")
print("X-train Project_Grade_Category BOW {} and y-train {}".format(X_train_p_grade_
print("="*20)
print("X-cv Project_Grade_Category BOW {} and y-cv {}".format(X_cv_p_grade_cat_ohe.s
print("="*20)
print("X-test Project_Grade_Category BOW {} and y-test {}".format(X_test_p_grade_cat
print("="*20)
print(p_grade_cat_vectorizer.get_feature_names())
print("="*20)
```

```
*** After vectorizations ***

X-train Project_Grade_Category BOW (40500, 4) and y-train (40500,)
====================
X-cv Project_Grade_Category BOW (4500, 4) and y-cv (4500,)
====================
X-test Project_Grade_Category BOW (15000, 4) and y-test (15000,)
====================
['grades_3_5', 'grades_6_8', 'grades_9_12', 'grades_prek_2']
====================
```

Encoding-->Project_Subject_Categories

In [74]:
```
p_subj_cat_vectorizer = CountVectorizer()
p_subj_cat_vectorizer.fit(X_train['project_subject_categories'].values) # fit has to

# we use the fitted CountVectorizer to convert the text to vector
X_train_p_subj_cat_ohe = p_subj_cat_vectorizer.transform(X_train['project_subject_ca
X_cv_p_subj_cat_ohe = p_subj_cat_vectorizer.transform(X_cv['project_subject_categori
X_test_p_subj_cat_ohe = p_subj_cat_vectorizer.transform(X_test['project_subject_cate

print("*** After vectorizations ***\n")
print("X-train Project_Subject_Categories BOW {} and y-train {}".format(X_train_p_su
print("="*20)
print("X-cv Project_Subject_Categories BOW {} and y-cv {}".format(X_cv_p_subj_cat_oh
print("="*20)
print("X-test Project_Subject_Categories BOW {} and y-test {}".format(X_test_p_subj_
print("="*20)
print(p_subj_cat_vectorizer.get_feature_names())
print("="*20)
```

```
*** After vectorizations ***

X-train Project_Subject_Categories BOW (40500, 49) and y-train (40500,)
====================
X-cv Project_Subject_Categories BOW (4500, 49) and y-cv (4500,)
====================
X-test Project_Subject_Categories BOW (15000, 49) and y-test (15000,)
====================
['applied__learning', 'applied__learning__health__sports', 'applied__learning__histo
ry__civics', 'applied__learning__literacy__language', 'applied__learning__math__scie
nce', 'applied__learning__music__arts', 'applied__learning__special__needs', 'applie
d__learning__warmth__care__hunger', 'health__sports', 'health__sports__applied__lear
ning', 'health__sports__history__civics', 'health__sports__literacy__language', 'hea
lth__sports__math__science', 'health__sports__music__arts', 'health__sports__special
__needs', 'health__sports__warmth__care__hunger', 'history__civics', 'history__civic
s__applied__learning', 'history__civics__health__sports', 'history__civics__literacy
__language', 'history__civics__math__science', 'history__civics__music__arts', 'hist
ory__civics__special__needs', 'literacy__language', 'literacy__language__applied__le
arning', 'literacy__language__health__sports', 'literacy__language__history__civic
s', 'literacy__language__math__science', 'literacy__language__music__arts', 'literac
y__language__special__needs', 'literacy__language__warmth__care__hunger', 'math__sci
ence', 'math__science__applied__learning', 'math__science__health__sports', 'math__s
cience__history__civics', 'math__science__literacy__language', 'math__science__music
__arts', 'math__science__special__needs', 'math__science__warmth__care__hunger', 'mu
sic__arts', 'music__arts__applied__learning', 'music__arts__health__sports', 'music_
_arts__history__civics', 'music__arts__special__needs', 'special__needs', 'special__
needs__health__sports', 'special__needs__music__arts', 'special__needs__warmth__care
__hunger', 'warmth__care__hunger']
====================
```

Encoding-->Project_Subject_Subcategories

In [75]:
```
p_subj_subcat_vectorizer = CountVectorizer()
p_subj_subcat_vectorizer.fit(X_train['project_subject_subcategories'].values) # fit

# we use the fitted CountVectorizer to convert the text to vector
X_train_p_subj_subcat_ohe = p_subj_subcat_vectorizer.transform(X_train['project_subj
X_cv_p_subj_subcat_ohe = p_subj_subcat_vectorizer.transform(X_cv['project_subject_su
X_test_p_subj_subcat_ohe = p_subj_subcat_vectorizer.transform(X_test['project_subjec

print("*** After vectorizations ***\n")
print("X-train Project_Subject_Subcategories BOW {} and y-train {}".format(X_train_p
print("="*20)
print("X-cv Project_Subject_Subcategories BOW {} and y-cv {}".format(X_cv_p_subj_sub
print("="*20)
print("X-test Project_Subject_Subcategories BOW {} and y-test {}".format(X_test_p_su
print("="*20)
```

```
# print(p_subj_subcat_vectorizer.get_feature_names())
# print("="*20)
```

```
*** After vectorizations ***

X-train Project_Subject_Subcategories BOW (40500, 373) and y-train (40500,)
====================
X-cv Project_Subject_Subcategories BOW (4500, 373) and y-cv (4500,)
====================
X-test Project_Subject_Subcategories BOW (15000, 373) and y-test (15000,)
====================
```

Encoding-->Price

In [76]:
```
price_normalizer = Normalizer()
price_normalizer.fit(X_train[['price']])

X_train_price_norm = price_normalizer.transform(X_train[['price']])
X_cv_price_norm = price_normalizer.transform(X_cv[['price']])
X_test_price_norm = price_normalizer.transform(X_test[['price']])

print("*** After vectorizations ***\n")
print("X-train Price {} and y-train {}".format(X_train_price_norm.shape, y_train.sha
print("="*20)
print("X-cv Price {} and y-cv {}".format(X_cv_price_norm.shape, y_cv.shape))
print("="*20)
print("X-test Price {} and y-test {}".format(X_test_price_norm.shape, y_test.shape))
print("="*20)
```

```
*** After vectorizations ***

X-train Price (40500, 1) and y-train (40500,)
====================
X-cv Price (4500, 1) and y-cv (4500,)
====================
X-test Price (15000, 1) and y-test (15000,)
====================
```

Encoding-->Quantity

In [77]:
```
quant_normalizer = Normalizer()
quant_normalizer.fit(X_train[['quantity']])

X_train_quant_norm = quant_normalizer.transform(X_train[['quantity']])
X_cv_quant_norm = quant_normalizer.transform(X_cv[['quantity']])
X_test_quant_norm = quant_normalizer.transform(X_test[['quantity']])

print("*** After vectorizations ***\n")
print("X-train Quantity {} and y-train {}".format(X_train_quant_norm.shape, y_train.
print("="*20)
print("X-cv Quantity {} and y-cv {}".format(X_cv_quant_norm.shape, y_cv.shape))
print("="*20)
print("X-test Quantity {} and y-test {}".format(X_test_quant_norm.shape, y_test.shap
print("="*20)
```

```
*** After vectorizations ***

X-train Quantity (40500, 1) and y-train (40500,)
====================
X-cv Quantity (4500, 1) and y-cv (4500,)
====================
X-test Quantity (15000, 1) and y-test (15000,)
====================
```

Concatinating_Encoded_Features

SET-1

- **Encoded Categorical Features + Encoded Numerical Features + Pre-processes `BOWs` of Titles, Essays and Summaries**

In [78]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr_bow = hstack((X_train_teacher_prefix_ohe,
                X_train_state_ohe,
                X_train_p_grade_cat_ohe,
                X_train_p_subj_cat_ohe,
                X_train_p_subj_subcat_ohe,
                X_train_title_bow,
                X_train_essay_bow,
                X_train_price_norm,
                X_train_quant_norm,
                X_train_summary_bow)).tocsr()

X_cv_bow = hstack((X_cv_teacher_prefix_ohe,
                X_cv_state_ohe,
                X_cv_p_grade_cat_ohe,
                X_cv_p_subj_cat_ohe,
                X_cv_p_subj_subcat_ohe,
                X_cv_title_bow,
                X_cv_essay_bow,
                X_cv_price_norm,
                X_cv_quant_norm,
                X_cv_summary_bow)).tocsr()

X_te_bow = hstack((X_test_teacher_prefix_ohe,
                X_test_state_ohe,
                X_test_p_grade_cat_ohe,
                X_test_p_subj_cat_ohe,
                X_test_p_subj_subcat_ohe,
                X_test_title_bow,
                X_test_essay_bow,
                X_test_price_norm,
                X_test_quant_norm,
                X_test_summary_bow)).tocsr()

print("*** Final Data matrix (BOW) ***\n")
print("Final X-train BOW {} and y-train {}".format(X_tr_bow.shape, y_train.shape))
print("="*20)
print("Final X-cv BOW {} and y-cv {}".format(X_cv_bow.shape, y_cv.shape))
print("="*20)
print("Final X-test BOW {} and y-test {}".format(X_te_bow.shape, y_test.shape))
print("="*20)
```

```
*** Final Data matrix (BOW) ***

Final X-train BOW (40500, 4484) and y-train (40500,)
====================
Final X-cv BOW (4500, 4484) and y-cv (4500,)
====================
Final X-test BOW (15000, 4484) and y-test (15000,)
====================
```

SET-2

- **Encoded Categorical Features + Encoded Numerical Features + Pre-processes `Tf-IDFs` of Titles, Essays and Summaries**

In [79]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
```

```
X_tr_tfidf = hstack((X_train_teacher_prefix_ohe,
                X_train_state_ohe,
                X_train_p_grade_cat_ohe,
                X_train_p_subj_cat_ohe,
                X_train_p_subj_subcat_ohe,
                X_train_title_tfidf_vect,
                X_train_essay_tfidf_vect,
                X_train_price_norm,
                X_train_quant_norm,
                X_train_summary_tfidf_vect)).tocsr()

X_cv_tfidf = hstack((X_cv_teacher_prefix_ohe,
                X_cv_state_ohe,
                X_cv_p_grade_cat_ohe,
                X_cv_p_subj_cat_ohe,
                X_cv_p_subj_subcat_ohe,
                X_cv_title_tfidf_vect,
                X_cv_essay_tfidf_vect,
                X_cv_price_norm,
                X_cv_quant_norm,
                X_cv_summary_tfidf_vect)).tocsr()

X_te_tfidf = hstack((X_test_teacher_prefix_ohe,
                X_test_state_ohe,
                X_test_p_grade_cat_ohe,
                X_test_p_subj_cat_ohe,
                X_test_p_subj_subcat_ohe,
                X_test_title_tfidf_vect,
                X_test_essay_tfidf_vect,
                X_test_price_norm,
                X_test_quant_norm,
                X_test_summary_tfidf_vect)).tocsr()

print("*** Final Data matrix (Tf-IDF) ***\n")
print("Final X-train Tf-IDF {} and y-train {}".format(X_tr_tfidf.shape, y_train.shap
print("="*20)
print("Final X-cv Tf-IDF {} and y-cv {}".format(X_cv_tfidf.shape, y_cv.shape))
print("="*20)
print("Final X-test Tf-IDF {} and y-test {}".format(X_te_tfidf.shape, y_test.shape))
print("="*20)
```

```
*** Final Data matrix (Tf-IDF) ***

Final X-train Tf-IDF (40500, 4484) and y-train (40500,)
====================
Final X-cv Tf-IDF (4500, 4484) and y-cv (4500,)
====================
Final X-test Tf-IDF (15000, 4484) and y-test (15000,)
====================
```

In [80]:
```
X_train.columns
```

Out[80]:
```
Index(['teacher_prefix', 'school_state', 'project_grade_category',
       'project_subject_categories', 'project_subject_subcategories',
       'teacher_number_of_previously_posted_projects', 'pp_titles',
       'pp_essays', 'price', 'quantity', 'pp_res_smrys'],
      dtype='object')
```

In [81]:
```
X_tr_bow
```

Out[81]:
```
<40500x4484 sparse matrix of type '<class 'numpy.float64'>'
        with 5236476 stored elements in Compressed Sparse Row format>
```

In [82]:
```
X_tr_tfidf
```

Out[82]:
```
<40500x4484 sparse matrix of type '<class 'numpy.float64'>'
```

```
with 5236476 stored elements in Compressed Sparse Row format>
```

Export_Matrices

- **Exporting pre-processed and encoded TRAIN, CV and TEST matrices**

In [83]:
```
from scipy import sparse
```

In [84]:
```
sparse.save_npz("X_tr_bow.npz", X_tr_bow)
sparse.save_npz("X_cv_bow.npz", X_cv_bow)
sparse.save_npz("X_te_bow.npz", X_te_bow)
sparse.save_npz("X_tr_tfidf.npz", X_tr_tfidf)
sparse.save_npz("X_cv_tfidf.npz", X_cv_tfidf)
sparse.save_npz("X_te_tfidf.npz", X_te_tfidf)
```

Export_Feature_Names

In [89]:
```
tfidf_features_names = []
```

In [90]:
```
for t_prefix_feature in t_prefix_vectorizer.get_feature_names():
    tfidf_features_names.append(t_prefix_feature)

for ss_feature in ss_vectorizer.get_feature_names():
    tfidf_features_names.append(ss_feature)

for p_grade_cat_feature in p_grade_cat_vectorizer.get_feature_names():
    tfidf_features_names.append(p_grade_cat_feature)

for p_subj_cat_feature in p_subj_cat_vectorizer.get_feature_names():
    tfidf_features_names.append(p_subj_cat_feature)

for p_subj_subcat_feature in p_subj_subcat_vectorizer.get_feature_names():
    tfidf_features_names.append(p_subj_subcat_feature)

for titles_feature in titles_tfidfvectorizer.get_feature_names():
    tfidf_features_names.append(titles_feature)

for essay_feature in essay_tfidfvectorizer.get_feature_names():
    tfidf_features_names.append(essay_feature)

tfidf_features_names.append("price")
tfidf_features_names.append("quantity")

for summary_feature in summary_tfidfvectorizer.get_feature_names():
    tfidf_features_names.append(summary_feature)
```

In [94]:
```
len(tfidf_features_names)
```

Out[94]: 4484

In [95]:
```
import pickle

with open('tfidf_feature_names', 'wb') as fp:
    pickle.dump(tfidf_features_names, fp)
```

Exported all the features in a pickle file.