DonorsChoose

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

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

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

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

About the DonorsChoose Data Set

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

Feature	Desc
`project_id`	A unique identifier for the proposed project. **Example:** `p0`
`project_title`	Title of the project. **Exa** • `Art Will Make You H` • `First Grad`
`project_grade_category`	Grade level of students for which the project is targeted. One of the fo enumerated v • `Grades P` • `Grade` • `Grade` • `Grades`
`project_subject_categories`	One or more (comma-separated) subject categories for the project fr following enumerated list of v • `Applied Lea` • `Care & H` • `Health & S` • `History & C` • `Literacy & Lan` • `Math & Sc` • `Music & The` • `Special` • `W` **Exar** • `Music & The` • `Literacy & Language, Math & Sc`

Feature	Desc
`school_state`	State where school is located (Two-letter U.S. posta (https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations#Postal_c **Exampl**
`project_subject_subcategories`	One or more (comma-separated) subject subcategories for the **Exar** • `Lit` • `Literature & Writing, Social Sci`
`project_resource_summary`	An explanation of the resources needed for the project. **Exa** • `My students need hands on literacy materials to ma sensory needs!<`
`project_essay_1`	First application
`project_essay_2`	Second application
`project_essay_3`	Third application
`project_essay_4`	Fourth application
`project_submitted_datetime`	Datetime when project application was submitted. **Example:** `2016-0 12:43:5`
`teacher_id`	A unique identifier for the teacher of the proposed project. **Exa** `bdf8baa8fedef6bfeec7ae4ff1c`
`teacher_prefix`	Teacher's title. One of the following enumerated v • • • • • • `Tea`
`teacher_number_of_previously_posted_projects`	Number of project applications previously submitted by the same te **Exam**

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

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

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

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

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

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

Notes on the Essay Data

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

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

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

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

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

In [1]:

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

# Files:
import os

# Data:
import sqlite3
import pandas as pd
import numpy as np
from collections import Counter

# Visuals:
import matplotlib.pyplot as plt
import seaborn as sns
from plotly import plotly
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from prettytable import PrettyTable

# Text:
import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
from nltk.corpus.corpora import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import nltk
from nltk.stem.porter import PorterStemmer
import string
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from gensim.models import Word2Vec
from gensim.models import KeyedVectors
#from sklearn.feature_extraction.text import TfidfTransformer

# Metrics:
from sklearn import metrics
from sklearn.metrics import confusion_matrix, roc_curve, auc

# Preprocessing:
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Misc:
import pickle
from tqdm import tqdm
```

```
c:\users\byron\applications\pythonmaster\lib\site-packages\gensim\utils.py:1
212: UserWarning:

detected Windows; aliasing chunkize to chunkize_serial
```

1. Reading Data

In [2]:

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

In [3]:

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

```
Number of data points in train data (109248, 17)
--------------------------------------------------
The attributes of data : ['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']
```

In [4]:

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

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

Out[4]:

	id	description	quantity	price
0	p233245	LC652 - Lakeshore Double-Space Mobile Drying Rack	1	149.00
1	p069063	Bouncy Bands for Desks (Blue support pipes)	3	14.95

2. Preprocessing Categorical Features: project_grade_category

In [5]:

```
project_data['project_grade_category'].value_counts()
```

Out[5]:

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

we need to remove the spaces, replace the '-' with '_' and convert all the letters to small

In [6]:

```
# https://stackoverflow.com/questions/36383821/pandas-dataframe-apply-function-to-colu
project_data['clean_grade_categories'] = project_data['project_grade_category'].str.re
project_data['clean_grade_categories'] = project_data['clean_grade_categories'].str.re
project_data['clean_grade_categories'] = project_data['clean_grade_categories'].str.lo
project_data['clean_grade_categories'].value_counts()
```

Out[6]:

```
grades_prek_2    44225
grades_3_5       37137
grades_6_8       16923
grades_9_12      10963
Name: clean_grade_categories, dtype: int64
```

In [7]:

```
project_data.drop(labels = ['project_grade_category'],axis=1,inplace=True)
```

3. Preprocessing Categorical Features: project_subject_categories

In [8]:

```
project_data['project_subject_categories'].value_counts()
```

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

remove spaces, 'the'
replace '&' with '_', and ',' with '_'

In [9]:

```
project_data['clean_subject_categories'] = project_data['project_subject_categories'].
project_data['clean_subject_categories'] = project_data['clean_subject_categories'].st
project_data['clean_subject_categories'] = project_data['clean_subject_categories'].st
project_data['clean_subject_categories'] = project_data['clean_subject_categories'].st
project_data['clean_subject_categories'] = project_data['clean_subject_categories'].st
project_data['clean_subject_categories'].value_counts()
```

```
history_civics_math_science          322
history_civics_music_arts            312
specialneeds_music_arts              302
health_sports_math_science           271
history_civics_specialneeds          252
health_sports_appliedlearning        192
appliedlearning_history_civics       178
health_sports_music_arts             155
music_arts_specialneeds              138
literacy_language_health_sports       72
health_sports_history_civics          43
specialneeds_health_sports            42
history_civics_appliedlearning        42
specialneeds_warmth_care_hunger       23
health_sports_warmth_care_hunger      23
music_arts_health_sports              19
music_arts_history_civics             18
history_civics_health_sports          13
math_science_warmth_care_hunger       11
appliedlearning_warmth_care_hunger    10
```

In [10]:

```
project_data.drop(labels = ['project_subject_categories'],axis=1,inplace=True)
```

4. Preprocessing Categorical Features: teacher_prefix

In [11]:

```
project_data['teacher_prefix'].value_counts()
```

Out[11]:

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

In [12]:

```
# check if we have any nan values are there
print(project_data['teacher_prefix'].isnull().values.any())
print("number of nan values",project_data['teacher_prefix'].isnull().values.sum())
```

```
True
number of nan values 3
```

> numebr of missing values are very less in number, we can replace it with Mrs. as most of the projects are submitted by Mrs.

In [13]:

```
project_data['teacher_prefix']=project_data['teacher_prefix'].fillna('Mrs.')
```

In [14]:

```
project_data['teacher_prefix'].value_counts()
```

Out[14]:

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

> Remove '.'
> convert all the chars to small

In [15]:

```
project_data['clean_teacher_prefix'] = project_data['teacher_prefix'].str.replace('.',
project_data['clean_teacher_prefix'] = project_data['clean_teacher_prefix'].str.lower(
project_data['clean_teacher_prefix'].value_counts()
```

Out[15]:

```
mrs        57272
ms         38955
mr         10648
teacher     2360
dr            13
Name: clean_teacher_prefix, dtype: int64
```

In [16]:

```
project_data.drop(labels = ['teacher_prefix'], axis=1, inplace=True)
```

5. Preprocessing Categorical Features: project_subject_subcategories

In [17]:

```
project_data['project_subject_subcategories'].value_counts()
```

```
Environmental Science, Team Sports                   2
Civics & Government, Team Sports                     2
Civics & Government, Health & Wellness               2
Early Development, Economics                         2
Financial Literacy, Health & Wellness                2
Other, Warmth, Care & Hunger                         1
History & Geography, Warmth, Care & Hunger           1
Financial Literacy, Foreign Languages                1
Community Service, Gym & Fitness                     1
Community Service, Financial Literacy                1
Civics & Government, Parent Involvement              1
Gym & Fitness, Warmth, Care & Hunger                 1
Community Service, Music                             1
Economics, Other                                     1
Civics & Government, Nutrition Education             1
Economics, Foreign Languages                         1
Financial Literacy, Performing Arts                  1
Economics, Nutrition Education                       1
Economics, Music                                     1
Gym & Fitness, Parent Involvement                    1
```

same process we did in project_subject_categories

In [18]:

```
project_data['clean_subject_subcategories'] = project_data['project_subject_subcategor
project_data['clean_subject_subcategories'] = project_data['clean_subject_subcategorie
project_data['clean_subject_subcategories'] = project_data['clean_subject_subcategorie
project_data['clean_subject_subcategories'] = project_data['clean_subject_subcategorie
project_data['clean_subject_subcategories'] = project_data['clean_subject_subcategorie
project_data['clean_subject_subcategories'].value_counts()
```

Out[18]:

```
literacy                                   9486
literacy_mathematics                       8325
literature_writing_mathematics             5923
literacy_literature_writing                5571
mathematics                                5379
literature_writing                         4501
specialneeds                               4226
health_wellness                            3583
appliedsciences_mathematics                3399
appliedsciences                            2492
literacy_specialneeds                      2440
gym_fitness_health_wellness                2264
esl_literacy                               2234
visualarts                                 2217
music                                      1472
warmth_care_hunger                         1309
literature_writing_specialneeds            1306
```

In [19]:

```
project_data.drop(labels = ['project_subject_subcategories'], axis=1, inplace=True)
```

6. Preprocessing Categorical Features: school_state

In [20]:

```
project_data['school_state'].value_counts()
```

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
```

convert all of them into small letters

In [21]:

```
project_data['clean_school_state'] = project_data['school_state'].str.lower()
project_data['clean_school_state'].value_counts()
```

Out[21]:

```
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
```

In [22]:

```python
project_data.drop(labels = ['school_state'], axis=1, inplace=True)
```

7. Preprocessing Categorical Features: project_title

In [23]:

```python
# 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
```

In [24]:

```python
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you'r
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him',
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', '
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "t
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'h
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'o
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'an
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 'too
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'n
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't"
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mig
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", '
            'won', "won't", 'wouldn', "wouldn't"]
```

In [25]:

```
project_data['project_title'].head(5)
```

Out[25]:

```
0      Educational Support for English Learners at Home
1                 Wanted: Projector for Hungry Learners
2     Soccer Equipment for AWESOME Middle School Stu...
3                                Techie Kindergarteners
4                                Interactive Math Tools
Name: project_title, dtype: object
```

In [26]:

```
print("printing some random reviews")
print(9, project_data['project_title'].values[9])
print(34, project_data['project_title'].values[34])
print(147, project_data['project_title'].values[147])
```

```
printing some random reviews
9 Just For the Love of Reading--\r\nPure Pleasure
34 \"Have A Ball!!!\"
147 Who needs a Chromebook?\r\nWE DO!!
```

In [27]:

```
# Combining all the above
def preprocess_text(text_data):
    preprocessed_text_list = []
    # 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 = sent.replace('nannan','')
        sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
        # https://gist.github.com/sebleier/554280
        sent = ' '.join(e for e in sent.split() if e.lower() not in stopwords)
        preprocessed_text_list.append(sent.lower().strip())
    return preprocessed_text_list
```

In [28]:

```
preprocessed_titles = preprocess_text(project_data['project_title'].values)
```

```
100%|████████████████████████████████████████████████████████████████████████
| 109248/109248 [00:02<00:00, 44243.82it/s]
```

In [29]:

```
print("printing some random reviews")
print(9, preprocessed_titles[9])
print(34, preprocessed_titles[34])
print(147, preprocessed_titles[147])
```

```
printing some random reviews
9 love reading pure pleasure
34 ball
147 needs chromebook
```

In [30]:

```
project_data['clean_project_title'] = preprocessed_titles
```

In [31]:

```
project_data.drop(labels = ['project_title'], axis=1, inplace=True)
```

8. Preprocessing Categorical Features: project_resource_summary

In [32]:

```
preprocessed_resource_sum = preprocess_text(project_data['project_resource_summary'].v
```

```
100%|████████████████████████████████████████
| 109248/109248 [00:05<00:00, 18955.49it/s]
```

In [33]:

```
project_data['clean_resource_summary'] = preprocessed_resource_sum
```

9. Preprocessing Categorical Features: essay

In [34]:

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

In [35]:

```
print("printing some random essay")
print(9, project_data['essay'].values[9])
print('-'*50)
print(34, project_data['essay'].values[34])
print('-'*50)
print(147, project_data['essay'].values[147])
```

```
printing some random essay
9 Over 95% of my students are on free or reduced lunch.  I have a few who
are homeless, but despite that, they come to school with an eagerness to l
earn.  My students are inquisitive eager learners who  embrace the challen
ge of not having great books and other resources  every day.  Many of them
are not afforded the opportunity to engage with these big colorful pages o
f a book on a regular basis at home and they don't travel to the public li
brary.  \r\nIt is my duty as a teacher to do all I can to provide each stu
dent an opportunity to succeed in every aspect of life. \r\nReading is Fun
damental! My students will read these books over and over again while boos
ting their comprehension skills. These books will be used for read alouds,
partner reading and for Independent reading. \r\nThey will engage in readi
ng to build their \"Love for Reading\" by reading for pure enjoyment. They
will be introduced to some new authors as well as some old favorites. I wa
nt my students to be ready for the 21st Century and know the pleasure of h
olding a good hard back book in hand. There's nothing like a good book to
read!  \r\nMy students will soar in Reading, and more because of your cons
ideration and generous funding contribution. This will help build stamina
and prepare for 3rd grade. Thank you so much for reading our proposal!nann
```

In [36]:

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

```
100%|████████████████████████████████████████████████████████████████████
█| 109248/109248 [00:54<00:00, 1997.23it/s]
```

In [37]:

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

```
printing some random essay
9 95 students free reduced lunch homeless despite come school eagerness le
arn students inquisitive eager learners embrace challenge not great books
resources every day many not afforded opportunity engage big colorful page
s book regular basis home not travel public library duty teacher provide s
tudent opportunity succeed every aspect life reading fundamental students
read books boosting comprehension skills books used read alouds partner re
ading independent reading engage reading build love reading reading pure e
njoyment introduced new authors well old favorites want students ready 21s
t century know pleasure holding good hard back book hand nothing like good
book read students soar reading consideration generous funding contributio
n help build stamina prepare 3rd grade thank much reading proposal
--------------------------------------------------
34 students mainly come extremely low income families majority come homes
parents work full time students school 7 30 6 00 pm 2 30 6 00 pm school pr
ogram receive free reduced meals breakfast lunch want students feel comfor
table classroom home many students take multiple roles home well school so
metimes caretakers younger siblings cooks babysitters academics friends de
veloping going become adults consider essential part job model helping oth
```

In [38]:

```
project_data['clean_essay'] = preprocessed_essays
```

In [39]:

```
project_data.drop(labels = ["essay","project_essay_1","project_essay_2","project_essay_
```

10. Preprocessing Numerical Values: price, quantity and poste_projects

In [40]:

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

Out[40]:

	id	price	quantity
0	p000001	459.56	7
1	p000002	515.89	21

In [41]:

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

In [42]:

```
project_data['price'].head()
```

Out[42]:

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

10.1 applying StandardScaler

In [43]:

```
scaler = StandardScaler()
scaler.fit(project_data['price'].values.reshape(-1, 1))
project_data['std_price']=scaler.transform(project_data['price'].values.reshape(-1, 1)

scaler.fit(project_data['quantity'].values.reshape(-1, 1))
project_data['std_quantity']=scaler.transform(project_data['quantity'].values.reshape(

scaler.fit(project_data['teacher_number_of_previously_posted_projects'].values.reshape
project_data['std_teacher_number_of_previously_posted_projects']=scaler.transform(proj
```

```
c:\users\byron\applications\pythonmaster\lib\site-packages\sklearn\utils\val
idation.py:475: DataConversionWarning:

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

c:\users\byron\applications\pythonmaster\lib\site-packages\sklearn\utils\val
idation.py:475: DataConversionWarning:

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

c:\users\byron\applications\pythonmaster\lib\site-packages\sklearn\utils\val
idation.py:475: DataConversionWarning:

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

c:\users\byron\applications\pythonmaster\lib\site-packages\sklearn\utils\val
idation.py:475: DataConversionWarning:

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

In [44]:

```
project_data['std_price'].head()
```

Out[44]:

```
0   -0.390533
1    0.002396
2    0.595191
3   -0.177469
4   -0.626236
Name: std_price, dtype: float64
```

10.2 applying MinMaxScaler

In [45]:

```
scaler = MinMaxScaler()
scaler.fit(project_data['price'].values.reshape(-1, 1))
project_data['nrm_price']=scaler.transform(project_data['price'].values.reshape(-1, 1)

scaler.fit(project_data['quantity'].values.reshape(-1, 1))
project_data['nrm_quantity']=scaler.transform(project_data['quantity'].values.reshape(

scaler.fit(project_data['teacher_number_of_previously_posted_projects'].values.reshape
project_data['nrm_teacher_number_of_previously_posted_projects']=scaler.transform(proj
```

```
c:\users\byron\applications\pythonmaster\lib\site-packages\sklearn\utils\val
idation.py:475: DataConversionWarning:

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

c:\users\byron\applications\pythonmaster\lib\site-packages\sklearn\utils\val
idation.py:475: DataConversionWarning:

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

In [46]:

```
project_data['nrm_price'].head()
```

Out[46]:

```
0    0.015397
1    0.029839
2    0.051628
3    0.023228
4    0.006733
Name: nrm_price, dtype: float64
```

In [47]:

```
project_data.drop(labels = ['price','quantity','teacher_number_of_previously_posted_pr
```

10.3 Resource summary countains digits

In [48]:

```
def check_numeric(x):
    return_list = list()
    contains_numeric=0
    for sentence in tqdm(x):
        for i in sentence.split():
            if i.isnumeric() == True:
                contains_numeric=1
            else:
                continue
        return_list.append(contains_numeric)
    return return_list
```

In [49]:

```
project_data['resource_summary_contains_numerical_digits'] = check_numeric(project_dat
```

```
100%|██████████████████████████████████████████████████████████████████████████████████|
109248/109248 [00:00<00:00, 354002.15it/s]
```

In [50]:

```
project_data.drop(labels = ['project_resource_summary'], axis=1, inplace=True)
```

11 Final features

In [51]:

```
final_data = project_data.loc[:,['project_submitted_datetime','clean_teacher_prefix','
                                 'clean_subject_subcategories','clean_project_title','c
                                 'resource_summary_contains_numerical_digits',
                                 'std_price','std_quantity','std_teacher_number_of_prev
                                 'nrm_price','nrm_quantity','nrm_teacher_number_of_prev
                                 'project_is_approved']]
```

In [52]:

```
final_data.to_csv('data/final_features.csv',index=False)
```