Business Problem

DonorsChoose.org receives hundreds of thousands of project proposals each year for classroom projects in desk of Donating. Right now it takes them 4 weeks to review these proposals and project approvals before they can be posted on the 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:

1. How to scale current manual procedures and resources to which 500,000 proposals per year they can review and approve as well as effectively as possible
2. How to ensure a consistent and fair process of project vetting across different volunteers to improve the experience for teachers
3. 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, taking into account the teacher's previous proposals, school and district demographics, and school data. DonorsChoose.org can then use the information to identify projects most likely to need further review before approval.

Dataset Information

The TRAIN.CSV data set provided by DonorsChoose contains the following features:

Feature	Description
project_id	A unique identifier for the proposed project. **Example:** `p036502`
project_title	Title of the project. **Examples:** - Art Supplies for Frida Kahlo's Classroom - It's Cold Outside for the Winter
project_grade_category	Grade level of students for which the project is targeted. One of the following enumerated values: - Grades PreK-2 - Grades 3-5 - Grades 6-8 - Grades 9-12
project_subject_categories	One or two (comma-separated) subject categories for the project from the following enumerated list of values: - Applied Sciences - Care & Hygiene - Health & Sports - History & Civics - Literacy & Language - Math & Science - Music & The Arts - Special Needs - Warmth **Examples:** - Music & The Arts - Literacy & Language, Math & Science
school_state	State where school is located (two-letter USPS abbreviation). **Example:** `NJ`
project_subject_subcategories	One or two (comma-separated) subject subcategories for the project. **Examples:** - Literature & Writing, Social Sciences An enumeration of the subcategories related to the project. **Examples:**
project_resource_summary	An explanation of how the project will be used. **Example:** `students need opportunities to practice beginning reading skills to build reading fluency.`
project_essay_1	"Introduce us to your classroom"
project_essay_2	"Tell us more about your students"
project_essay_3	"Your project would impact this population."
project_essay_4	"Describe how your project will be used."
project_submitted_datetime	Datetime when project application was submitted. **Example:** `2016-04-28 12:43:56.245`
teacher_id	A unique identifier for the teacher of the proposed project. **Example:** `bdf8baa8fedef6bfeec7ae4ff8c6fabc`
teacher_prefix	Teacher's title. One of the following enumerated values: - nan - Dr. - Mr. - Mrs. - Ms. - Teacher.
teacher_number_of_previously_posted_projects	Number of previously posted applications previously submitted by this same teacher. **Example:** `2`

See the section **Notes on the Data Fields** 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 for a project:

Feature	Description
id	A project id. **Example:** `p036502`
description	Description of the resource. **Examples:** `Tenor Saxophone Reeds, Boy of 25`
quantity	Quantity of the resource required. **Example:** `3`
price	Price of the resource required. **Example:** `$135.00`

Note: Many projects require multiple resources. The `id` value corresponds to a `project_id` in train.csv, so you can join it to the data described above to examine resources 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.

Load Libraries

```
!pip install -U plotly
!pip install cufflinks
!pip install chart_studio
!jupyter nbextension enable --py widgetsnbextension
!ipkernel install --user --name=tensorflow_gpu
!jupyter nbextension enable --py --sys-prefix qgrid
!jupyter nbextension enable --py --sys-prefix ipyparallel
!jupyter nbextension enable --py --sys-prefix ipywidgets
!jupyter nbextension enable --py --sys-prefix ipyvolume
!jupyter nbextension enable --py --sys-prefix pythreejs
!jupyter nbextension enable --py --sys-prefix ipyleaflet
!jupyter nbextension enable --py --sys-prefix ipygeoscatter
!jupyter nbextension enable --py --sys-prefix ipycanvas
!jupyter nbextension enable --py --sys-prefix ipytwitter
!ipkernel install --user --name=tensorflow_gpu
!jupyter nbextension enable --py --sys-prefix qgrid
!jupyter nbextension enable --py --sys-prefix ipyparallel
!jupyter nbextension enable --py --sys-prefix ipywidgets
!jupyter nbextension enable --py --sys-prefix ipyvolume
!jupyter nbextension enable --py --sys-prefix pythreejs
!jupyter nbextension enable --py --sys-prefix ipyleaflet
!jupyter nbextension enable --py --sys-prefix ipygeoscatter
!jupyter nbextension enable --py --sys-prefix ipycanvas
!jupyter nbextension enable --py --sys-prefix ipytwitter
!jupyter nbextension enable --py --sys-prefix ipywidgets
!jupyter nbextension enable --py --sys-prefix ipyvolume
!jupyter nbextension enable --py --sys-prefix pythreejs
!jupyter nbextension enable --py --sys-prefix ipyleaflet
```

```
In [14]: print("Approved:",approved)
         print("Not approved:",not_approved)
         plt.hist(approved,bins=50,alpha=0.5,color='orange',label='Approved')
         plt.hist(not_approved,bins=50,alpha=0.5,color='red',label='Not approved')
         plt.xlabel('project_essay_length_in_characters')
         plt.ylabel('count')
         plt.legend()
         plt.title('A density plot presents density function for approved and not approved project')
         plt.show()
```

```
Approved: 92706
Not approved: 7327
```