SEIS 764-02 Artificial Intelligence

Spring 2020

Kiel Auer
Monique Dargis
Jeff Fillipi
Erik Hutchinson
Derek Synan

I. Dataset Introduction and Problem

The dataset for this project was obtained from Kaggle,[1] a website that provides open datasets for use in data science projects. Kaggle regularly hosts competitions in which data scientists can compete to provide the best solution given a problem and a dataset. The dataset used for this project was originally part of a Playground Prediction Competition held by Kaggle in April 2018.

The data originates from DonorsChoose.org, a website that allows public school teachers to request funding for classroom projects. The DonorsChoose.org funding model asks teachers to submit proposals outlining the goal of their project, resources needed, and cost of the project. Donors can then fund all or part of a project via crowd funding by donating toward funds for the purchase of specific resources. This Kaggle competition was centered around helping DonorsChoose develop a model to "prediction 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." Such a model would allow DonorsChoose staff to auto-approve some project applications, reducing the need for manual review.

The original dataset contained four zipped files – resources.csv (124 MB), sample_submission.csv (296 KB), test.csv (140 MB) and train.csv (328 MB). Competitors were to use the train.csv (joined with resources.csv) to train a model, and then predict the likelihood that each project in test.csv would get approved using the area under the ROC curve. However, although train.csv contains a target variable for training purposes, test.csv does not. For this reason, we did not use test.csv and instead both trained and tested our model using only the train.csv dataset. We did not use sample_submission.csv as this file is only useful if we were to submit to the Kaggle competition (which has been closed for two years).

II. Data Preparation

The properties of the features in the two files we used, train.csv and resources.csv, can be found in Appendix A. Descriptions taken from Kaggle data page[2] and statistics generated via pandas.

Null values

When exploring the dataset, we discovered that there are 6,734 records in train.csv that have values for essays 3 and 4, while the other 175,706 records are null in these fields. DonorsChoose changed their essay criteria on May 17, 2016, so the 6,734 records were submitted before that date. Because these are the oldest records (so, least likely to reflect current DonorsChoose practices) and there are comparatively few, we removed these

[1] https://www.kaggle.com/c/donorschoose-application-screening

[2] https://www.kaggle.com/c/donorschoose-application-screening/data

records from the dataset. We did not remove any other records due to Null values as it is possible that a proposal that is missing values is more (or less) likely to be approved.

Data transformation

Many of the fields in the dataset needed to be transformed before they could be used in our models. The train.csv file was organized by project, while the resources.csv file was organized by individual resource request, often with multiple records that belonged to the same project. We determined that the best way to handle this mismatch in data granularity was to find the total cost of each project. To do this, we grouped resources.csv by project id, aggregated the total cost (sum of per-item costs times volume requested), and joined this new "resources_agg" column with train.csv

Next, we dropped the columns for essays 3 and 4 (since we removed all records that had values in those fields) and concatenated essays 1 and 2 into one field, project_essays, to prepare for sentiment analysis.

Although we knew we needed to one hot encode the subject category and subcategory columns, we discovered that some columns had one entry while others had two entries per column. So, for example, a project could have two categories in one column and two subcategories in another column. We split these into separate columns so that each column had only one or zero category or subcategory names and then were able to one hot encode the resulting columns.

We also one hot encoded the target variable (project approval) and grade categories and scaled numeric data points (project cost, number of previously approved proposals) using standardscaler.

Data augmentation

The final change to our dataset involved augmenting the school location data. The only data given for each school's location was its state, and there were 51 possible values in this field (including DC). We did not wish to end up with 51 features from one hot encoding this variable alone, so we divided the states into 'subregions' taken from the US census.[3] There were only 9 subregions, ["Pacific", "Mountain", "West North Central", "West South Central", "East North Central", "East South Central", "Middle Atlantic", "South Atlantic", "New England"], which seemed like a more appropriate number of categories. We added this data using a data dictionary and then one hot encoded the resulting subregions.

III. Modeling

[3] https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

IV. Conclusion

Appendix A: Data Dictionary for Original Dataset

train.csv – 182,080 records				
Feature	**Description**	**Data type**	**Required?**	**Data properties and statistics**
id	Unique ID of the project application	String	Yes	'p' plus six digits 0-9. PRIMARY KEY
teacher_id	ID of the teacher submitting the application	String	Yes	32 alphanumeric characters, not case-sensitive
teacher_prefix	Teacher's title	String	No	4 NULL values; Others choose one from ["Ms.", "Mrs.", "Mr.", "Teacher", "Dr."]
school_state	US state of the teacher's school	String	Yes	One from ["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"]
project_submitted_datetime	Application submission timestamp	Datetime	Yes	YYYY-MM-DD HH:MM:SS
project_grade_category	School grade levels	String	Yes	["Grades PreK-2", "Grades 3-5", "Grades 6-8", "Grades 9-12"]
project_subject_categories	Category of the project	String	Yes	One or two options from the following: ["Applied Learning", "Health & Sports", "History & Civics", "Literacy & Language","Math & Science", "Music & The Arts", "Special Needs", "Warmth, Care & Hunger"]
project_subject_subcategories	Sub-category of the project	String	Yes	One or two options from the following: ["Applied Sciences", "Character Education", "Civics & Government", "College & Career Prep", "Community Service", "Early Development", "Economics", "Environmental Science", "ESL", "Extracurricular", "Financial Literacy", "Foreign Languages", "Gym & Fitness", "Health & Life Science", "Health & Wellness", "History & Geography", "Literacy", "Literature & Writing", "Mathematics", "Music", "Nutrition Education", "Other", "Parent Involvement", "Performing Arts", "Social Sciences", "Special Needs", "Team Sports", "Visual Arts", "Warmth, Care & Hunger"]
project_title	Title of the project	String	Yes	alphanumeric and punctuation character range = [4,141]; median = 30; mode = 21

train.csv				
Feature	**Description**	**Data type**	**Required?**	**Data properties and statistics**
project_essay_1	*See notes below	String	Yes	alphanumeric and punctuation character range = [50,2760]; median = 605; mode = 505
project_essay_2	*See notes below	String	Yes	alphanumeric and punctuation character range = [248, 5224]; median = 731; mode = 638
project_essay_3	*See notes below	String	No	175,706 NULL values; 6,734 non-NULL with alphanumeric and punctuation character range = [250,1675]; median = 509; mode = [998, 999, 1000]
project_essay_4	*See notes below	String	No	175,706 NULL values; 6,734 non-NULL with alphanumeric and punctuation character range =[180, 1178]; median = 351; mode = 500
project_resource_summary	Descriptive listing of items in resources.csv	String	Yes	alphanumeric and punctuation characters
teacher_number_of_previously_posted_projects	Number of previously poster applications by the submitting teacher	Integer	Yes	range = [0,451]; median = 2; mode = 0
project_is_approved	Whether the DonorsChoose proposal was accepted	Binary	Yes	154,346 projects approved; 27,734 projects not approved

*Essay prompts changed on May 17, 2016. For the 6,734 applications submitted with four essays before May 17, 2016, the prompts are as follows:

project_essay_1: “Introduce us to your classroom”
project_essay_2: “Tell us more about your students”
project_essay_3: “Describe how your students will use the materials you’re requesting”
project_essay_4: “Close by sharing why your project will make a difference”

For the 175,706 applications submitted on or after May 17, 2016, only two essays were submitted. The prompts are as follows:

project_essay_1: “Describe your students: What makes your student special” Specific details about their background, your neighborhood, and your school are all helpful.”
project_essay_2: “About your project: How will these materials make a difference in your students’ learning and improve their school lives?”

resources.csv – 1,541,272 records				
Feature	**Description**	**Data type**	**Required?**	**Data properties and statistics**
id	Unique ID of the project application	String	Yes	'p' plus six digits 0-9. Corresponds with 'id' in train.csv
description	description of the resources requested	String	No	292 NULL; otherwise, alphanumeric and punctuation characters
quantity	quantity of resources requested	Integer	Yes	Range = [1,800]; median = 1; mode = 1. When organized by project, range = [1,930]; median = 9; mode = 1
price	price of resources requested	Float	Yes	Up to two decimal places. Range = [0,9999]; median = 14.99; mode = 29.99. 4,528 items with value 0 (2059 of these have label 'Standard Shipping') and 32 items with value 9999 (All are 'Google Expeditions Kit' or 'Google Expeditions Kit - (30 Students)'). When organized by project, range = [0.66,9999]; median = 206.02; mode = 479.

Data Cleaning

- Imported train.csv and resources.csv and assigned datatypes
- From train.csv:
    - Removed all project proposals submitted before May 17, 2016 as these submissions had different essay prompts.
        - Total of 6,734 records removed, with 175,706 records remaining.
        - Dropped project_essay_3 and project_essay_4 columns
    - Added school_region and school_subregion columns and mapped values based on project state.
        - Total of four regions and nine subregions included.
    - Subject category cleaning
        - Can have one or two categories, but they are both included in the same column project_subject_categories with comma separation
            - "Warmth, Care, and Hunger" is only category label that also includes a comma
            - Split on comma, then clean up "Warmth, Care, and Hunger" – results in two columns project_subject_category1 and project_subject_category2
        - One-hot-encode by hand then drop project_subject_category1 and project_subject_category2
        - Total of eight possible categories
    - Subject subcategory cleaning
        - Can have one or two categories, but they are both included in the same column project_subject_subcategories with comma separation
            - "Warmth, Care, and Hunger" is only subcategory that also includes a comma
            - Split on comma, then clean up "Warmth, Care, and Hunger" – results in two columns project_subject_subcategory1 and project_subject_subcategory2
        - One-hot-encode by hand then drop project_subject_subcategory1 and project_subject_subcategory2
        - Total of 29 possibly categories
- From resources.csv:
    - Aggregated data so that quantity of resources requested and total cost of resource requested are summed by project ID
    - Left join train.csv data with resources.csv aggregation