MACHINE LEARNING ENGINEER NANODEGREE

CAPSTONE PROPOSAL

Domain Background

Refer to https://www.kaggle.com/donorschoose/io/home

DonorsChoose.org acts as a channel which links teachers requiring funding for their classrooms projects & the donors who are willing to sponsor those projects. Even though 3 million people & their partners have funded 1.1 million donorsChoose.org projects, the teachers continue to spend $1 billion of their own money on the classroom projects. It imperative that efforts should be made to simplify donors ability to sponsor projects.

The classroom project is usually defined by a teacher includes fields like project title, project essay, funding & resources requirements etc.
The donor can search by topics, teacher, schools & by city, state and zip codes.

The relationship between the projects and donors is many-to-many which means that a donor can sponsor multiple projects and a project can be sponsored by multiple donors.

The problem on hand is to identify the projects with their related projects. The donors are passionate about specific projects and if we can associate the projects with all the related projects, the donor can be presented with all the related projects. Given this information, the donor can make informed choices quickly.

I have learnt various disciplines of machine learning such as supervised, unsupervised, reinforcement & deep learning at Udacity. I came across document similarity using SVM (Support Vector Machine) using Linear Kernel. I am making an effort to utilize this algorithm in this project.

Problem Statement

The "project essay" is the primary field to define the purpose of the project. There are other subsidiary fields such as category & sub-category & title.

The problem to be solved is.

1) Employ document similarity algorithm to identify a project with its related projects. If a donor is interested in a set of projects, the organization should be able to present all the related projects to the donor. This will help the donor to choose the projects he will be willing to sponsor quickly & effortlessly.

Datasets and Inputs

The public data is available for download at https://www.kaggle.com/donorschoose/io

It consists of the following

1) Donations.csv: This dataset lists all the donation made by the donors. It has many-to-one relationship with the projects and the donors. It has about 4.69 million records.
2) Donor.csv: This lists all unique donors for this organization. It has about 2.12 million records.
3) Projects.csv: This lists all the projects. It has about 5.02 million records.

I will be primarily using Projects dataset alone. The other two inputs provides a context as how a project is related to Donors and their donations. Refer to the ER diagram.

Solution Statement

The problem to be solved is identify a project with other similar projects. The similarity can be established by comparing the “project essay”: field with projects dataset. So this is essentially a document similarity problem. I plan to use Supervised Machine learning using SVM (Support Vector Machine) for document classification.

Benchmark Model

I plan to employ Support Vector Machine (SVM) in conjunction with Linear Kernel for classifying the documents.

The following are generally acceptable reasons for using Linear kernel

1) Text is usually linearly separable.
2) Text has a lot of features.
    - The linear kernel is good when there is a lot of features
3) Linear kernel is faster as compared to other kernels.

Evaluation Metrics

TF-IDF (Term Frequency - Inverse Document Frequency) is a concept defined as “A term that occurs in a few documents is likely to be a better discriminator than a term that appears in most or all documents.”

1. A document is similar to another document in terms of cosine similarity score which varies from 0 (not similar) to 1 (most similar). The comparison score is relative to other documents involved in the comparison.

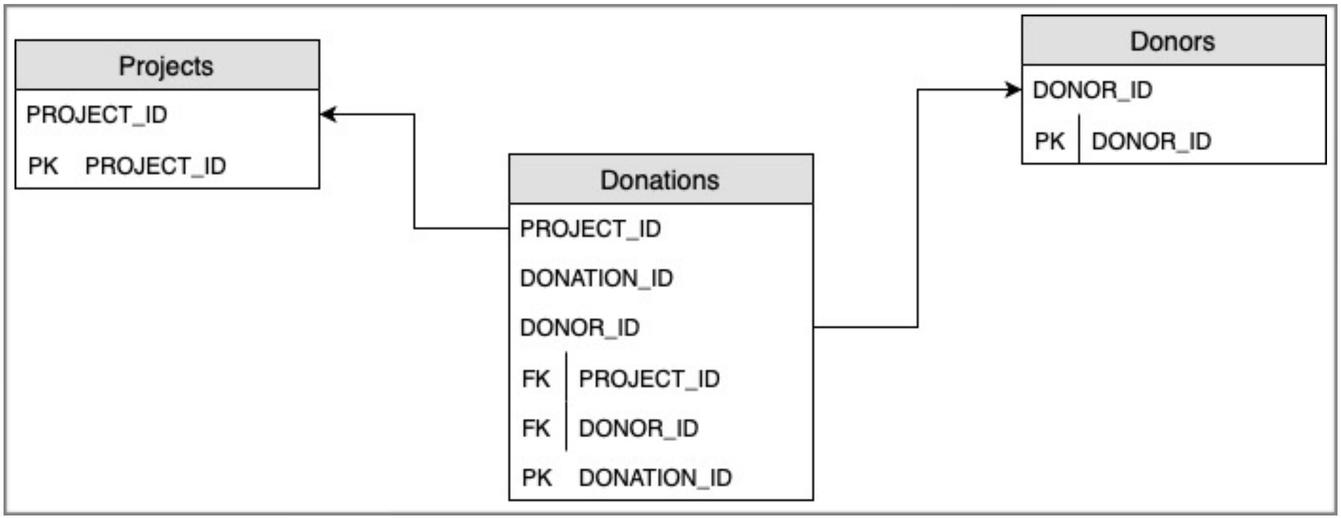
2. The results of the computation should include the related projects with their id and the cosine similarity score as a tuple for any given project. This will be included a separate column in the projects dataset. If the cosine similarity score is 0 for the related projects, the related projects value should be null.

3. It should be possible to query the related documents with a criteria such as "Show top 20 related documents where the cosine similarity score is greater than 0.5".

Project Design

The ER diagram of the data is shown below. There is 1-to-many relationship between the Donor & Donations and the Project & Donations. Similarly a Donor can sponsor multiple projects and a project can have multiple Donors.

Note: Only selected attributes are shown in the diagram for the purpose of brevity.



The attribute “Project Essay” describes the purpose of the project and this content of this field will be used as a document for the project. Documents of several projects will be compared to determine if there is a similarity between them.

Here is the pseudo code for implementation (This will be subject to change during actual implementation)

1) Read input file projects dataset (projects_df)

2) Perform cleanup of the data

3) Fetch the project essay from the above dataset into an array (project_text)

4) Create an instance of TfidfVectorizer

```
from sklearn.feature_extraction.text import TfidfVectorizer
 vectorizer = TfidfVectorizer(
                 strip_accents='unicode',
                 analyzer='word',
                 lowercase=True, # Convert all uppercase to lowercase
                 stop_words='english', # Remove common English words
                 max_df = 0.9)
tfidf_matrix = vectorizer.fit_transform(project_text)
```

5) Create a function to return a list of tuples (project id, cosine similarity score).

Here I will be using SVM (Support Vector Machine) linear kernel function to classify the documents.

```
from sklearn.metrics.pairwise import linear_kernel

 def find_similar_docs(index, top_similar = 20):
    cos_similar = linear_kernel(tfidf_matrix[index:index+1], tfidf_matrix).flatten()
```

```
# identify the related documents excluding the current index
# using list comprehension
# return top_similar array a tuple of (index, cosine similarity value)
```

6) Associate a project to its related projects.

Loop thru the project_df and pass on the row index and top projects parameters to the function shown at 5)

Convert the list of tuples returned by the function into a dictionary of project id and similarity value.

This will help us to filter this dictionary for any similarity score such as "Fetch all the projects which have similarity score greater than 0.6"

Add this dictionary as a column with dataset project_df

7) Verify the results obtained. If necessary fine tune the code till consistency is established.

References:

Document similarity :

https://sites.temple.edu/tudsc/2017/03/30/measuring-similarity-between-texts-in-python/

https://courses.cs.washington.edu/courses/cse573/12sp/lectures/17-ir.pdf

https://www.kaggle.com/ranliu/donor-project-matching-with-recommender-systems

http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

https://markhneedham.com/blog/2016/07/27/scitkit-learn-tfidf-and-cosine-similarity-for-computer-science-papers/

https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/