

Auto-tagging Global Giving Projects

By: Melissa Barona

1. Problem Statement

1.1 Introduction


The mission of GlobalGiving is “to transform aid and philanthropy to accelerate community-led change.” To do this, GlobalGiving provides a web-based fundraising platform that supports nonprofits by connecting them with donors and companies. A nonprofit organization can use this platform to pitch their charitable projects on the GlobalGiving website to potential donors. Donors can browse on the website and select from a variety of projects to contribute to. Since 2002, more than 1.1M donors on GlobalGiving have donated more than \$530 million to support more than 28,000 projects in 170 countries.

The nonprofit organizations pitch their development projects by describing the mission and long-term impact of the project in the project description. Based on the nature of the project, GlobalGiving organizes the projects by themes such as education, gender equality, child protection, economic growth, among others. Currently, there are 14,720 active projects organized under 28 different themes. It can be tedious, expensive, and time-consuming to manually assign each project to a theme. Therefore, the goal of this capstone project is to use machine learning to assign a project to a theme and streamline the content development process. Such auto-tagging system can also be used in a metadata management system that offers options and solutions on how to best tag a project such that the nonprofit attracts and retains the most ideal donors.

1.2 Product Vision

Exhibit A is a screenshot of GlobalGiving’s website showing three different projects posted on GlobalGiving. All three could be classified as educational, since all three titles contain words related to education. For example, the first title contains the word “student”, the second title contains the word “educate”, and the third title contains the word “schools”. A simple rule-based algorithm may classify all three projects under education. A more complex rule-based algorithm will require more rules to distinguish amongst these projects. However, each project has a different mission. The first project’s goal is to help medical students become better doctors (education), the second project’s goal is to

empower women to improve a country (gender equality), and the third project's goal is to protect the local environment and improve the livelihood of young children in Moroccan schools and villages (child protection). An appropriate tag of a project to a theme is essential for attracting and retaining the ideal donor that will continue to support the mission of the project.




EDUCATION | INDIA

Helping medical students become better doctors

by QMed Knowledge Foundation

In India, medical/health sciences students waste precious hours as they are not taught structured online searching & referencing skills. Th... [read more](#)

\$59,763 raised of \$75,000 goal




GENDER EQUALITY | SIERRA LEONE

Educate a Girl, Educate a Nation - Sierra Leone

by Develop Africa, Inc.

Across the region, 9 million girls between the ages of about 6 and 11 will never go to school, compared to 6 million boys, according to UIS ... [read more](#)

\$261,886 raised of \$265,000 goal



CHILD PROTECTION | MOROCCO

Improve Rural Moroccan Schools: Sami's Project

by High Atlas Foundation

Sami's Project will plant 5,000 fruit and nut trees with young children in 96 participating Moroccan schools and villages. HAF's staff along... [read more](#)

\$52,531 raised of \$100,000 goal

Exhibit A: Screenshot of GlobalGiving's website showing three different projects posted on GlobalGiving.

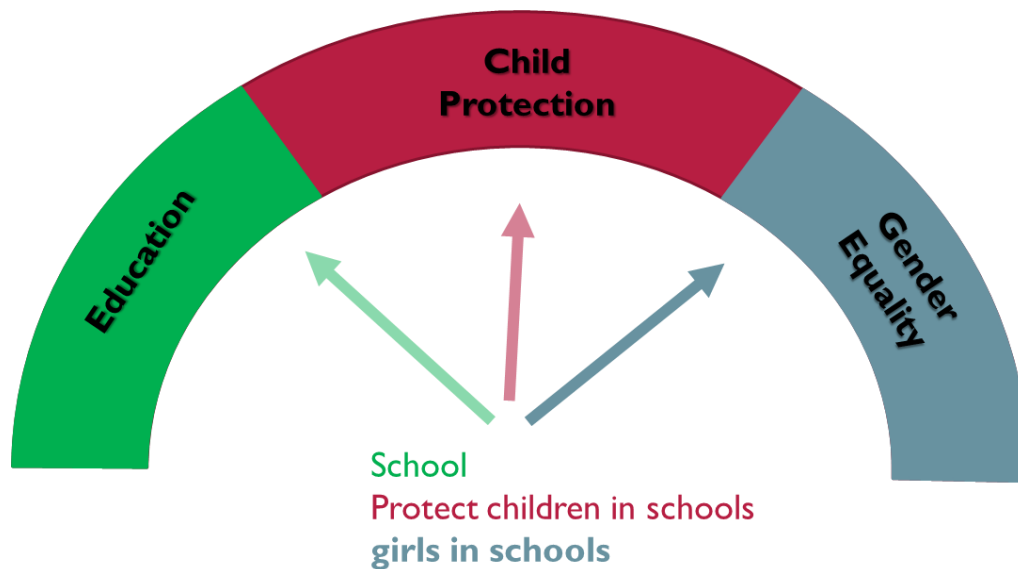


Exhibit B: The product vision is an auto-tagger that properly tags a project based on the context of its description

2. Objective

In this project I will build a topic classification model that classifies a project under one of the four most common themes on GlobalGiving – education, physical health, gender equality, and economic growth. The features will be extracted from the project description, which contains a summary of the project, the challenge it is tackling, and the long-term impact. To do this, I will use natural language processing (NLP) methods to convert the text into numerical feature vectors.

The two ML algorithms that I will use are Naïve Bayes and Support Vector Machine, and the models will be evaluated based on the percentage of projects that were assigned to the correct theme.

3. Data Source and Data Cleaning

The themes with the number of projects currently on the Global Giving website are the following:

- Education (2,479)
- Physical Health (1,422)
- Gender Equality (1,255)
- Child Protection (916)
- Economic Growth (868)
- Justice and Human Rights (715)
- Food Security (671)
- COVID-19 (624)
- Mental Health (558)
- Climate Action (544)
- Disaster Response (521)
- Safe Housing (496)
- Ending Abuse (487)
- Ecosystem Restoration (304)
- Refugee Rights (302)
- Sustainable Agriculture (299)
- Wildlife Conservation (281)
- Clean Water (279)
- Disability Rights (274)
- Animal Welfare (262)
- Reproductive Health (261)
- Arts and Culture (218)
- Digital Literacy (195)
- Ending Human Trafficking (169)
- Sport (121)
- Peace and Reconciliation (109)
- Racial Justice (54)
- LGBTQIA+ Equality (36)

The dataset for this project was collected from the Global Giving website using their API. The texts were gathered by collecting the title and the descriptions for the activities, challenge, solution, and long-term impact for each project. Each text was cleaned by removing numbers, hyphens, stopwords, and punctuation. Lemmatization was used to reduce words to their lemma, or root word. Lastly, the data containing 38,188 projects and their descriptions was saved in a .csv file.

4. Preprocessing and Modeling

To convert the text into numerical feature vectors, I used the bag-to-words, the term frequency-inverse document frequency (TF-IDF), and the word2vec word embedding methods. Applying the word embedding methods resulted in a very large matrix that my personal computer could not handle. Therefore, I subsampled 2,500 projects from the 'Education', 'Economic Growth', 'Gender Equality', and 'Physical Health' themes for training and testing. Figure 1 shows the most common words in the documents categorized under 'Education', 'Economic Growth', 'Gender Equality', and 'Physical Health'.

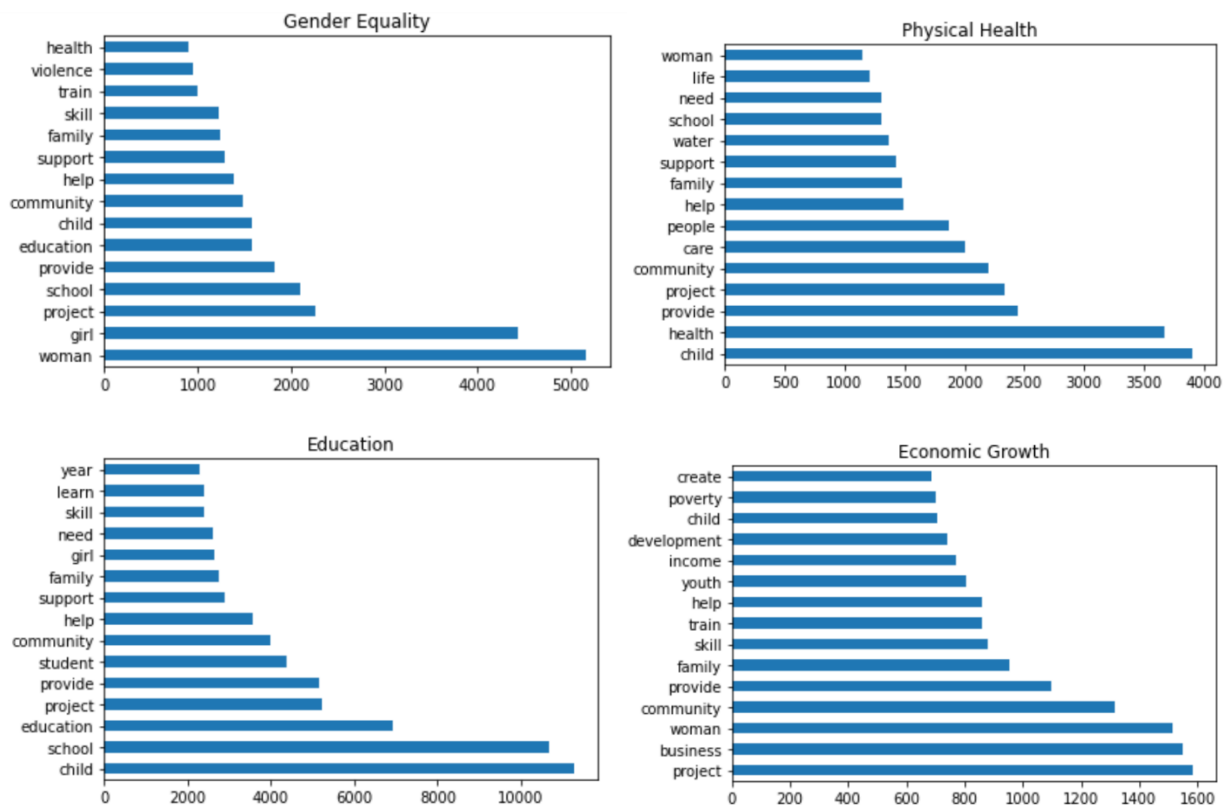


Figure 1: Most frequent words in documents categorized as 'Education', 'Gender Equality', 'Physical Health', and 'Economic Growth'.

The four categories have several words in common, such as 'project', 'provide', 'help', 'child', 'community', and 'family'. 'School', 'education', 'woman', and 'girl' appear often in 'Gender Equality' and 'Education' texts which can lead to some texts being misclassified under these themes. 'Water' and 'health' are most frequent in 'Physical Health' and 'development' and 'income' are most frequent in 'Economic Growth' texts.

While the bag-of-words method counts the number of times a word appears in the text, The TF-IDF method takes into account the importance of the word in the document. A word will be less important if it is common and appears in all texts. Therefore, words such as 'project', 'provide', and help', which are the most frequent words in all categories, will be assigned a lower importance and a lower TF-IDF value. In this manner, the TF-IDF method helps identify the words that are most important for text classification. For these reasons, I used the TF-IDF method first to convert the text for each project into feature vectors prior to training the Naïve Bayes and SVM models. This resulted in a sparse matrix of 10,000 by 26,677 in size.

4.1 TF-IDF/Naïve Bayes and TF-IDF/SVM models

To optimize the Naïve Bayes model, I used GridSearchCV to tune the variance smoothing parameter of the Gaussian Naïve Bayes algorithm. Figure 2 shows that the accuracy is improved by 12% when the variance smoothing parameter is increased from the default value of 10^{-9} to 10^{-2} . Figures 3 and 4 show the confusion matrix when the fitted Naïve Bayes model is applied on the training and testing datasets, respectively. When the model is applied on the test set, 'Education' projects are misclassified the most, and 20% of them get classified under 'Gender Equality'. Average accuracy is 0.76, which is 17% less accurate compared to when the model is applied on the training set. This suggests that the Naïve Bayes algorithm is overfitting the training set.

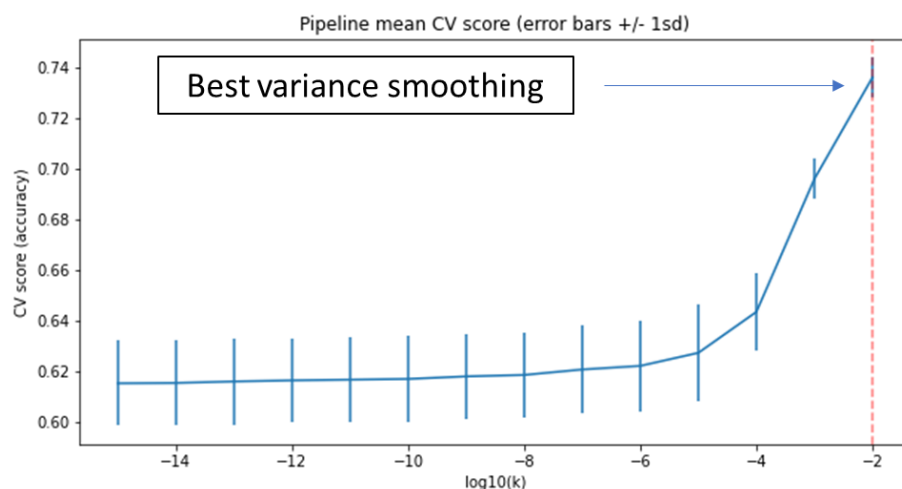


Figure 2: Hyperparameter tuning of the variance smoothing parameter in Gaussian Naïve Bayes algorithm

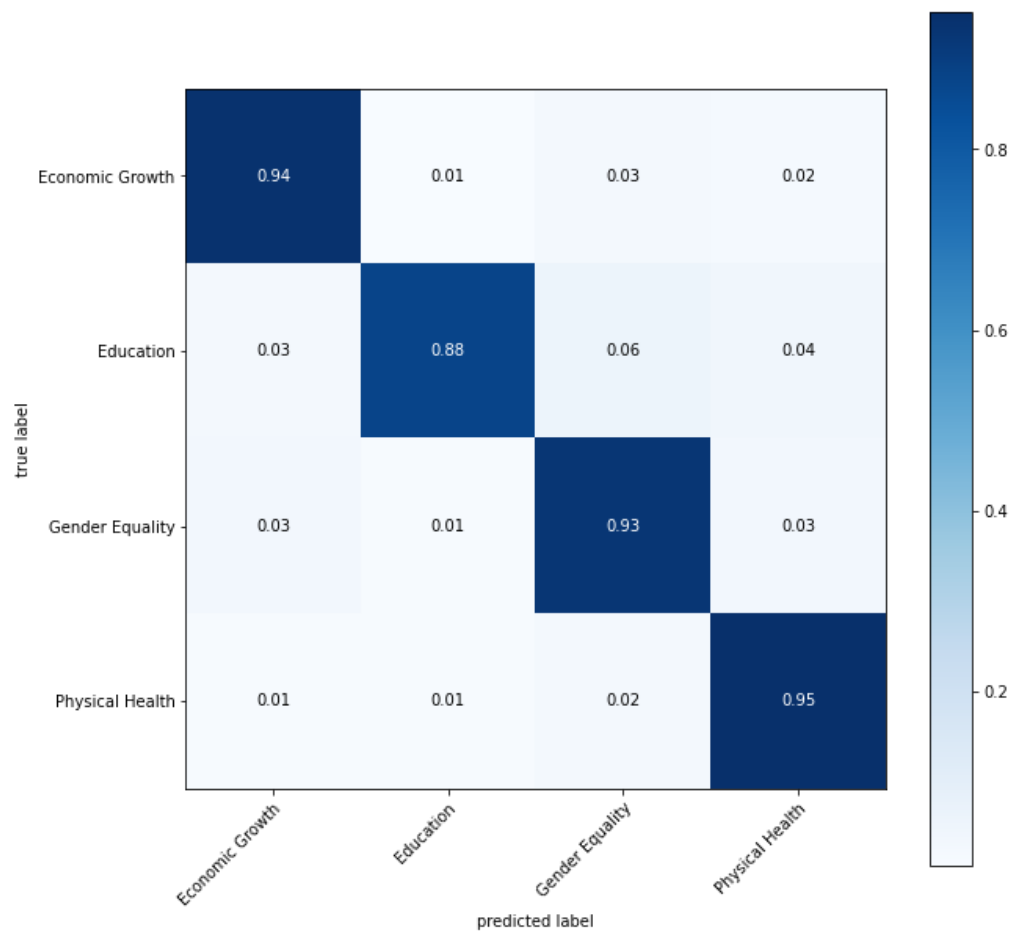


Figure 3: Confusion matrix for training dataset with the Gaussian Naïve Bayes model.
Average accuracy is 0.93

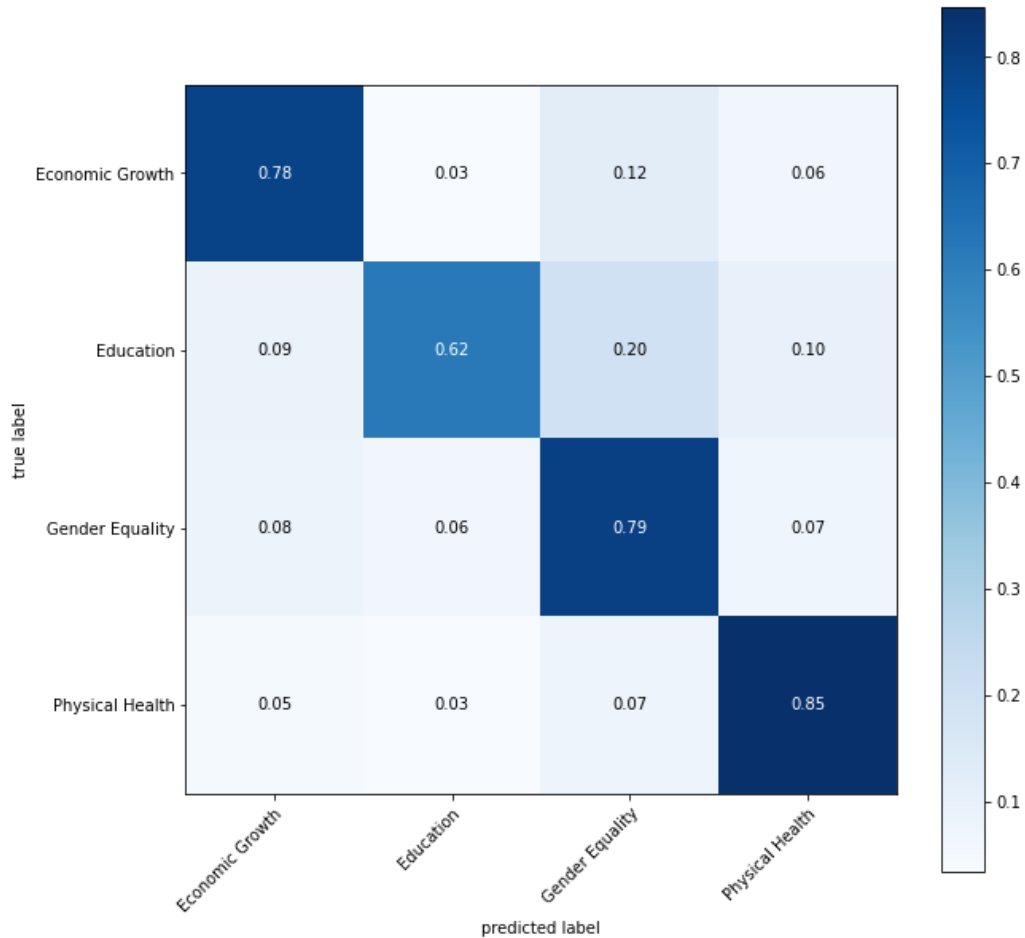


Figure 4: Confusion matrix for testing dataset with the Gaussian Naïve Bayes model. Average accuracy is 0.76.

While fitting the Naïve Bayes model took about an hour with hyperparameter tuning, fitting a linear SVM model took about 8 hours with no hyperparameter tuning. Figure 5 shows the confusion matrix when the linear SVM model with default parameter values is applied to test set. The average accuracy is 0.84, which is 8% higher than the best Naïve Bayes model previously discussed. The SVM model classifies the 'Education' projects better than the Naïve Bayes model by 20%.

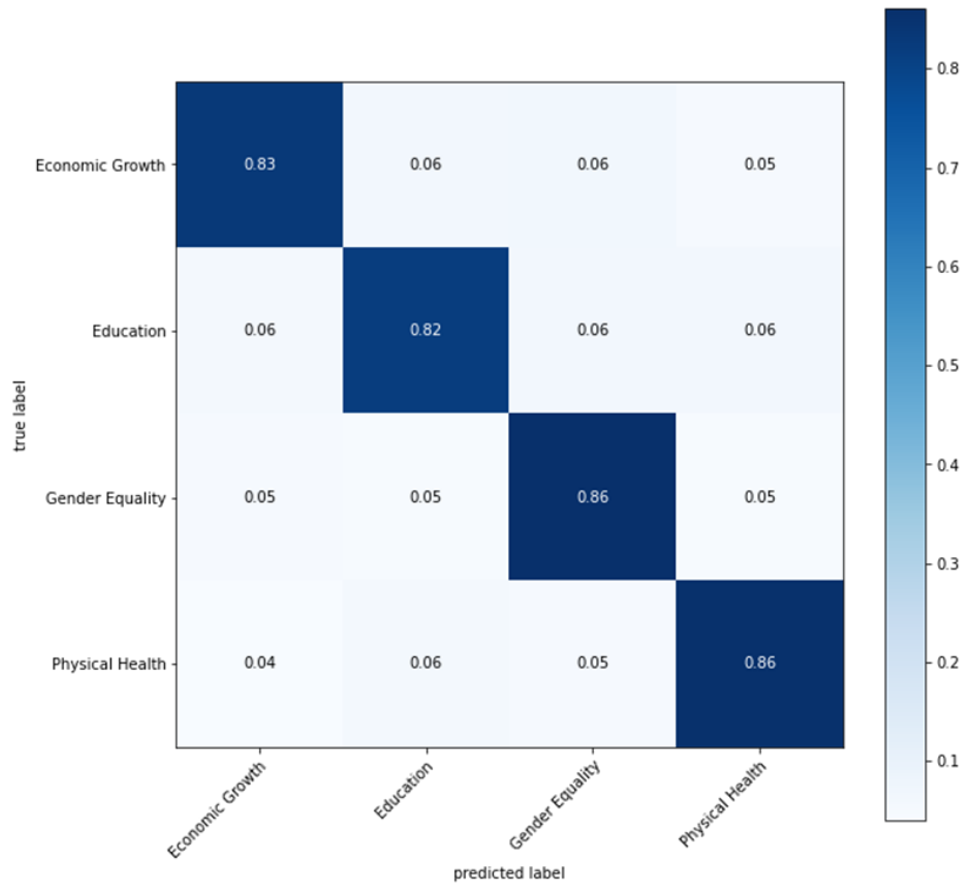


Figure 5: Confusion matrix for the testing dataset with the linear SVM model. Average accuracy is 0.84.

4.2 Word2Vec/Naïve Bayes and Word2Vec/SVM models

While both Bag-of-Words and TF-IDF are the most popular word-embedding methods, there are a number of drawbacks: (1) They don't capture the context of the words and their similarity, (2) if the new sentences contain new words, then our vocabulary size would increase and thereby, the length of the vectors would increase too, making the dataset very sparse. This affects the efficiency of the training. To improve the efficiency of the model and better capture the context of words and their similarity, I used the word2vec word-embedding method, which allows me to choose the size of the word vector. This allowed me to reduce the dataset to a matrix of 10,000 by 300 in size.

Figures 6 and 7 show the confusion matrix for the performance of the Naïve Bayes and SVM models, respectively. Here we find that the SVM model is 13% more accurate than

the Naïve Bayes model. The reduced size of dataset also allowed me to optimize the SVM model.

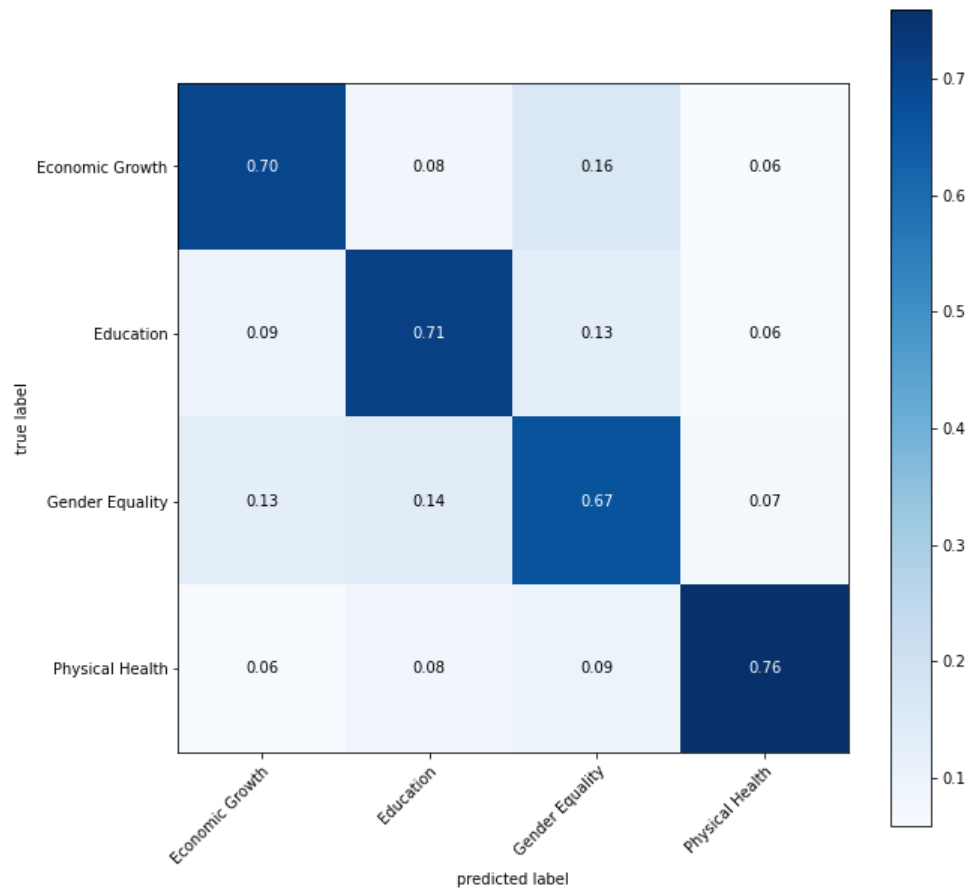


Figure 6: Confusion matrix for the testing dataset with the Naïve Bayes model. Average accuracy is 0.71.

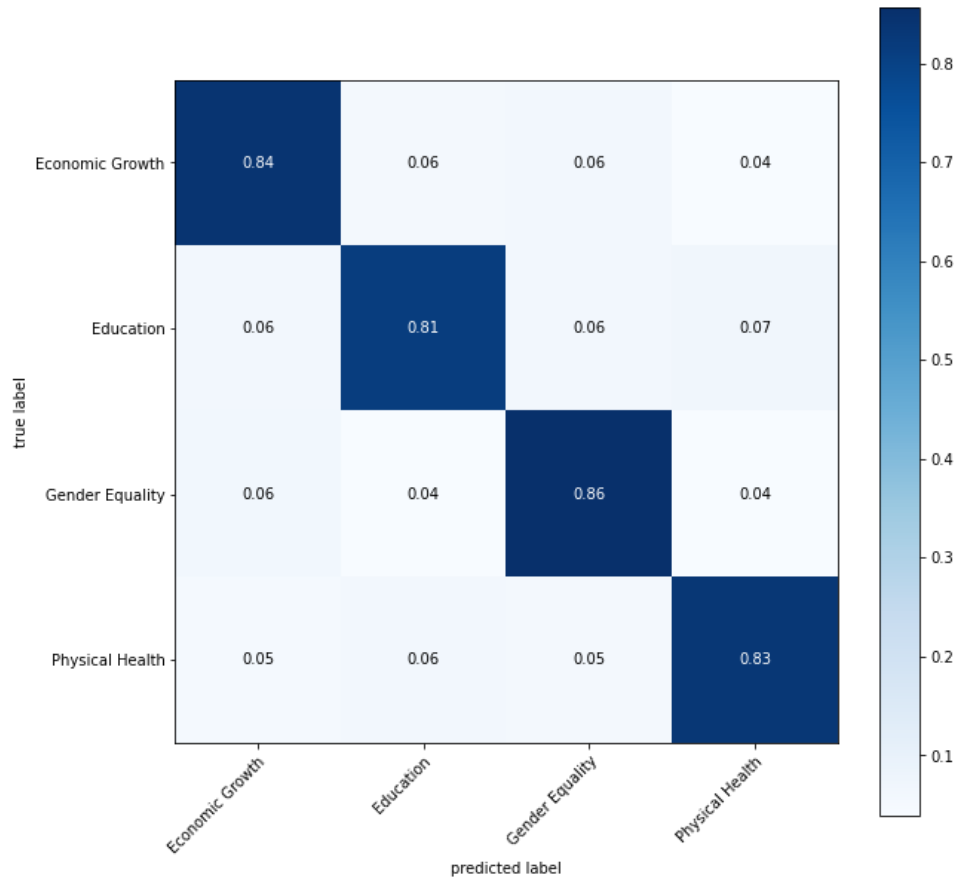


Figure 7: Confusion matrix for the testing dataset with the word2vec/SVM model. Average accuracy is 0.84.

Table 1: Summary of the overall accuracy of each model

Model	Overall Accuracy	
	Training Set	Testing Set
TF-IDF/Naïve Bayes	0.94	0.92
TF-IDF/Linear SVM	0.94	0.84
Word2vec/Naïve Bayes	0.70	0.71
Word2vec/RDF SVM	0.99	0.84

8. Future Improvements

In this project we found that a combination of the word2vec word-embedding method and an SVM model classifies best the 'Education', 'Gender Equality', 'Economic Growth', and 'Physical Health' projects. Because the word2vec method results in a less sparse dataset, the next step would be to apply the word2vec word-embedding method to the entire dataset of containing all 38,811 projects and fit a radial SVM model. Depending on how accurate this model is, the challenges of this next step will continue to be distinguishing among projects with similar language (e.g., Exhibit A) and identifying a metric that can assess how correctly the model classifies these projects, but also gives merit to partially correct classification.