In order to collect data on team activities, we first identified teams of interest from 2018. From the iGem 2018 awards page, we found, downloaded, and merged the json files containing data on all the teams, medals they received, and awards won. Using a simple python script with the pandas package, all the data was cleaned and merged into a single file, with teams of interest being gold winners. Then, a web scraper written in python, using the beautifulsoup and selenium packages was deployed to each team’s wiki. The scraper would navigate the website to locate the education and public engagement pages, as well as the integrated human practices pages. It only failed to locate these pages on three sites. While the scraper was able to handle and load javascript, these three sites required specific user interactions that the scraper was unable to provide. Since only three sites had issues, the most efficient solution was just to do these manually. If more sites had issues, then handlers for various generic javascript interactions could be implemented. Once our scraper reached the pages of interest, it would extract the text information from the page, grouping it by section header and section text. We determined this was the most flexible model that worked across the greatest number of sites, since there was no standard format or writing style across all the team wikis. Being able to group text by section was thus the most logical step, as the data would be fed into a trained neural network. However, this model failed on some sites where data was encoded into images, rather than text. Because these sites were few, we read the images manually. However if there were many sites with this issue, a solution would be to train and deploy an optical character recognition algorithm in order to extract meaning from the images. Furthermore, there were also a few pages where the data was not separated very well, due to these pages being contiguous blocks of text, rather than being organized by sections. These sites are not huge issues, as the text is still extracted, and they only suffer from slightly lower accuracy when sent to the text analysis neural network.