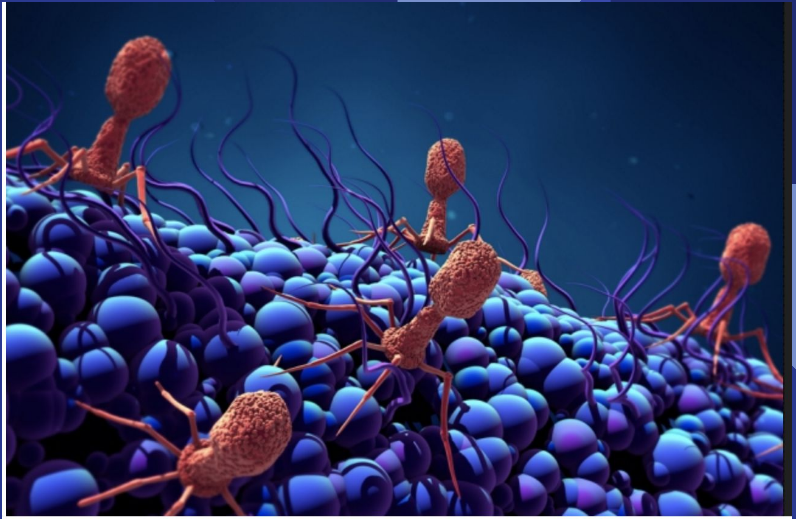
Project 3: NLP Binary Classification Model

Amy Ontiveros-Bear
DSI-10



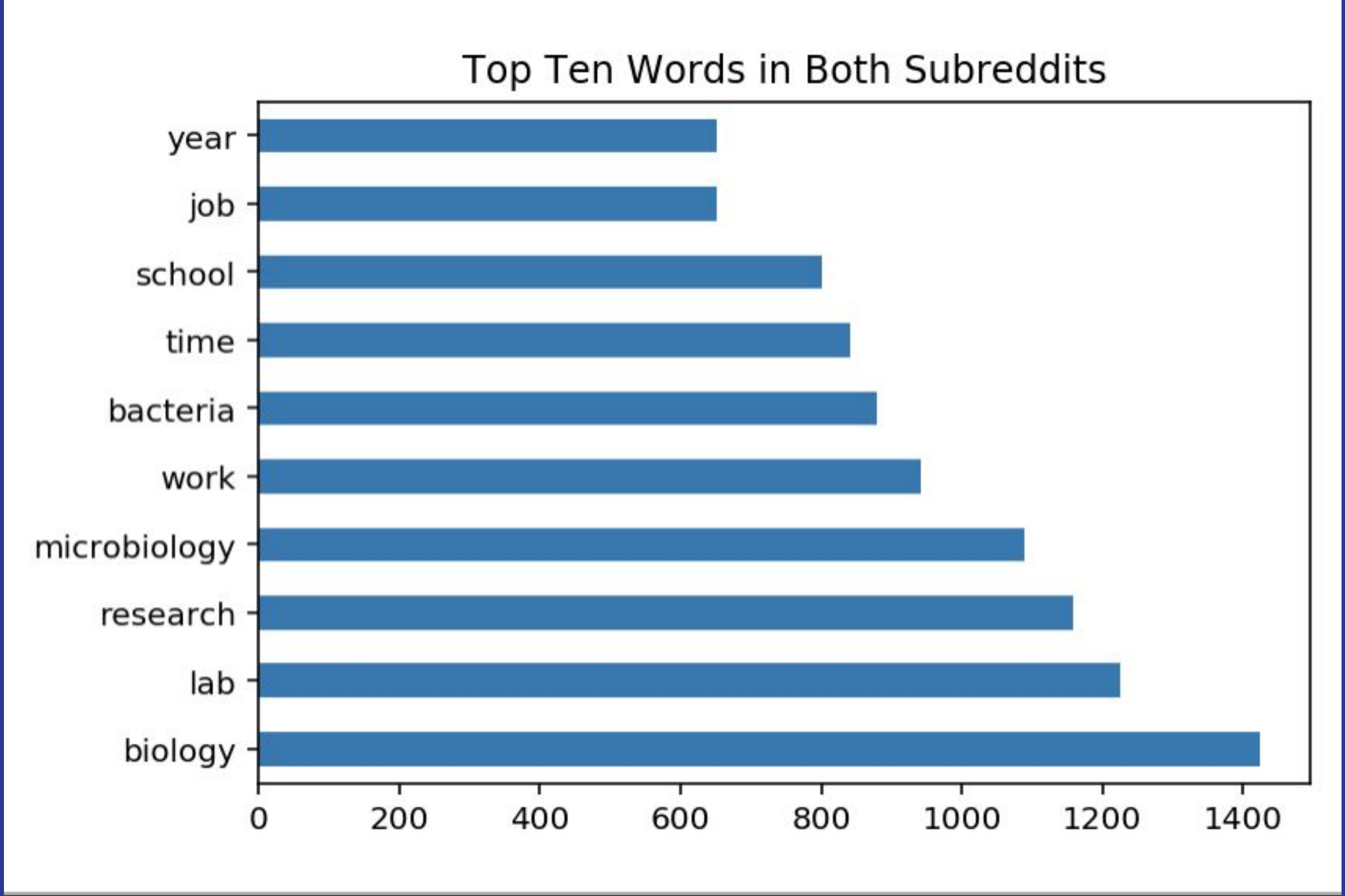
EYE OF SCIENCE/ SPL

Problem Statement:

- I am moderator for r/Biology and r/ Microbiology, someone hacked my subreddits and combined them into one! I need to build a model ASAP that will help me differentiate between the subreddit posts. Once complete, I can begin to organize the mess that the hacker created! Luckily, I have some back up data from both subreddits that I can use to train my models with. Since this is a NLP binary classification problem I will try out Logistic and Naive Bayes models with accuracy as my metric.

Exploratory Data Analysis:

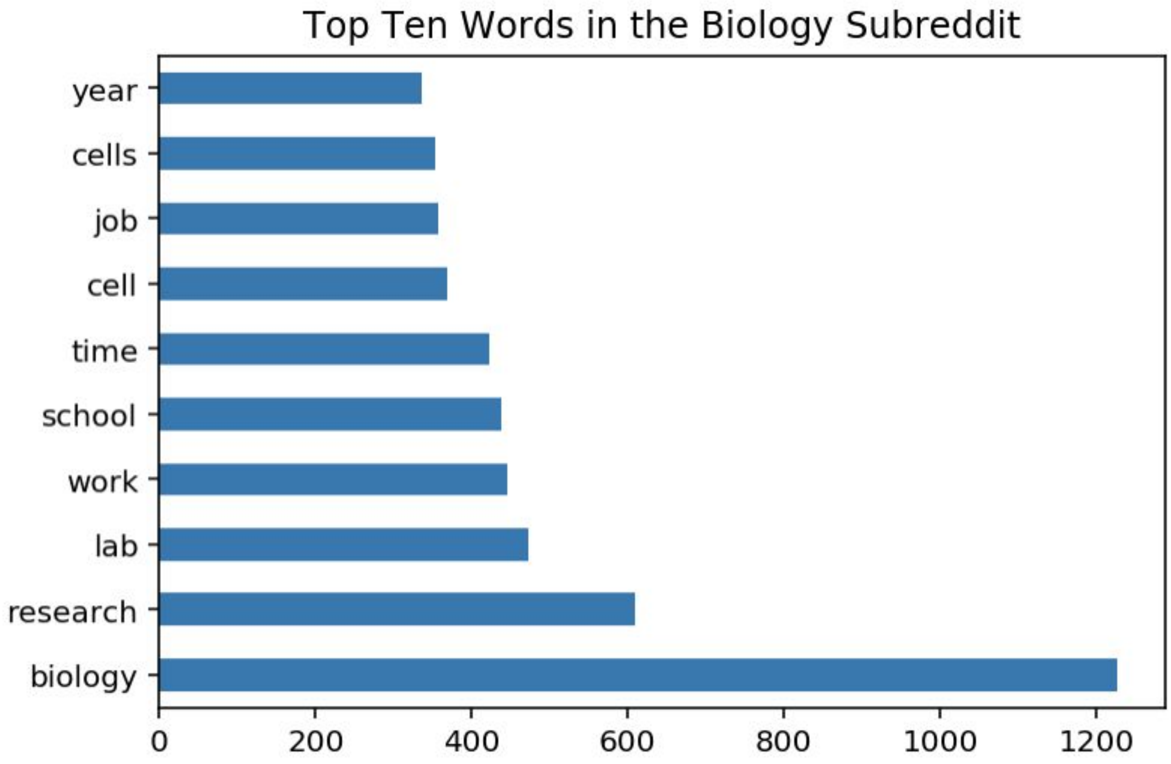
1. Count Vectorized my entire data set.
2. Sorted out the most frequently occuring.
3. Customized Stopwords.
4. Graphed the top ten words from each subreddit.


Top Ten Words in Both Subreddits
year
job
school
time
bacteria
work
microbiology
research
lab
biology
0
200
400
600
800
1000
1200
1400

Observations:

- Since Microbiology is a subfield of Biology I figured they share some common words like lab, research etc…

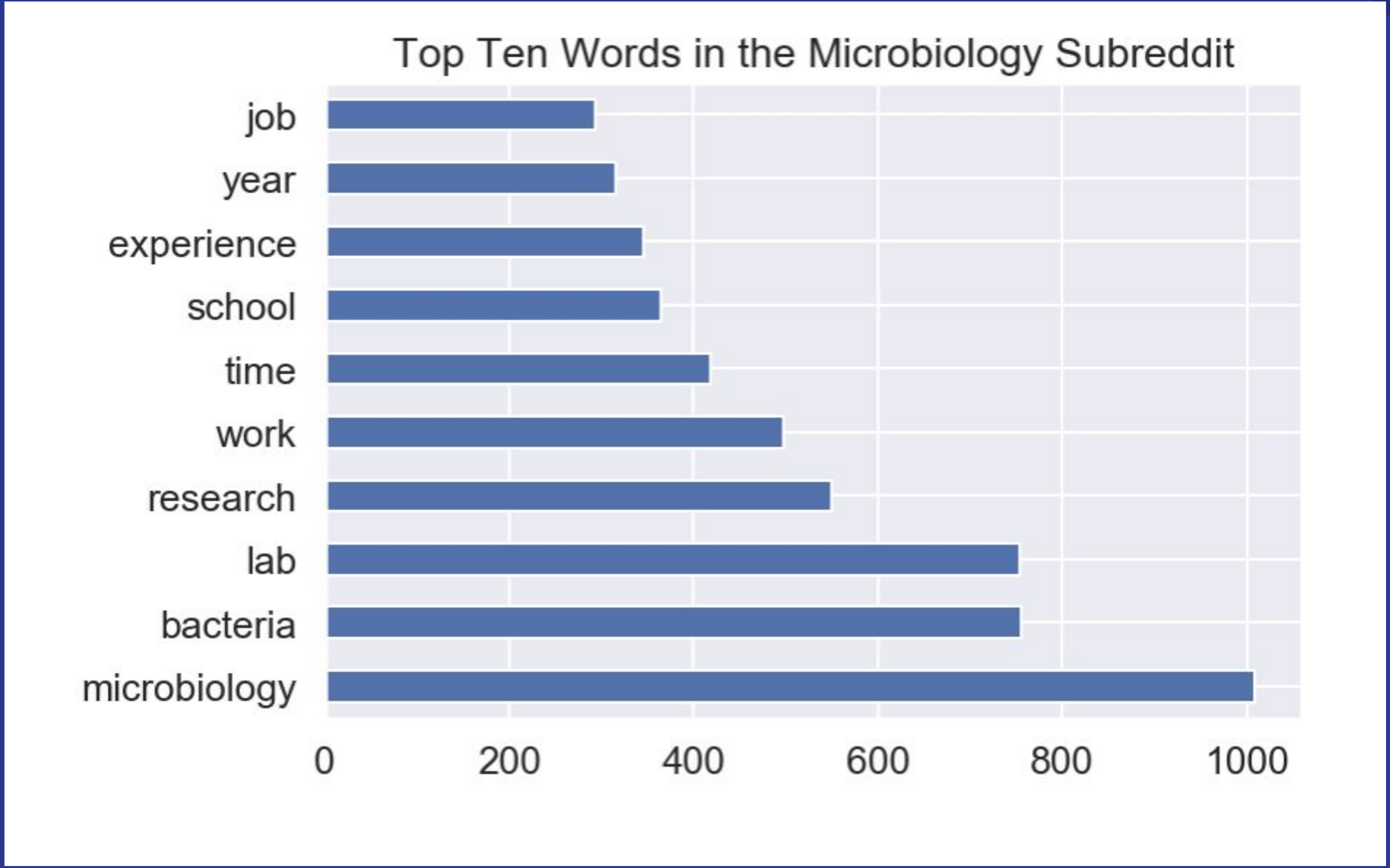
- Reddit is a forum based community we do get a lot of people asking for help and/ or advice in these fields, which would explain the rest of the cross over words.


Top Ten Words in the Biology Subreddit
year
cells
job
cell
time
school
work
lab
research
biology
0
200
400
600
800
1000
1200

Observations:

- It is very interesting that biology took the lead for cell and cells, I would have expected this trophy to go to micro.

- As expected they share common terminology, and biology is the key indicator word for this subreddit's corpus.


Top Ten Words in the Microbiology Subreddit
job
year
experience
school
time
work
research
lab
bacteria
microbiology
0
200
400
600
800
1000

Observations:

- I am surprised that bacteria is the only field specific word that populated in the top ten.

- Microbiology is the key indicator word for this subreddit's corpus.

- Both subreddits shared 70% of the same top words which is more than I expected.

Models:

- TfidfVectorizer & Logistic Regression
- CountVectorizer & Logistic Regression
- Multinomial Naive Bayes & CountVectorizer
- Gaussian Naive Bayes & TfidfVectorizer Model

Model’s Scores:

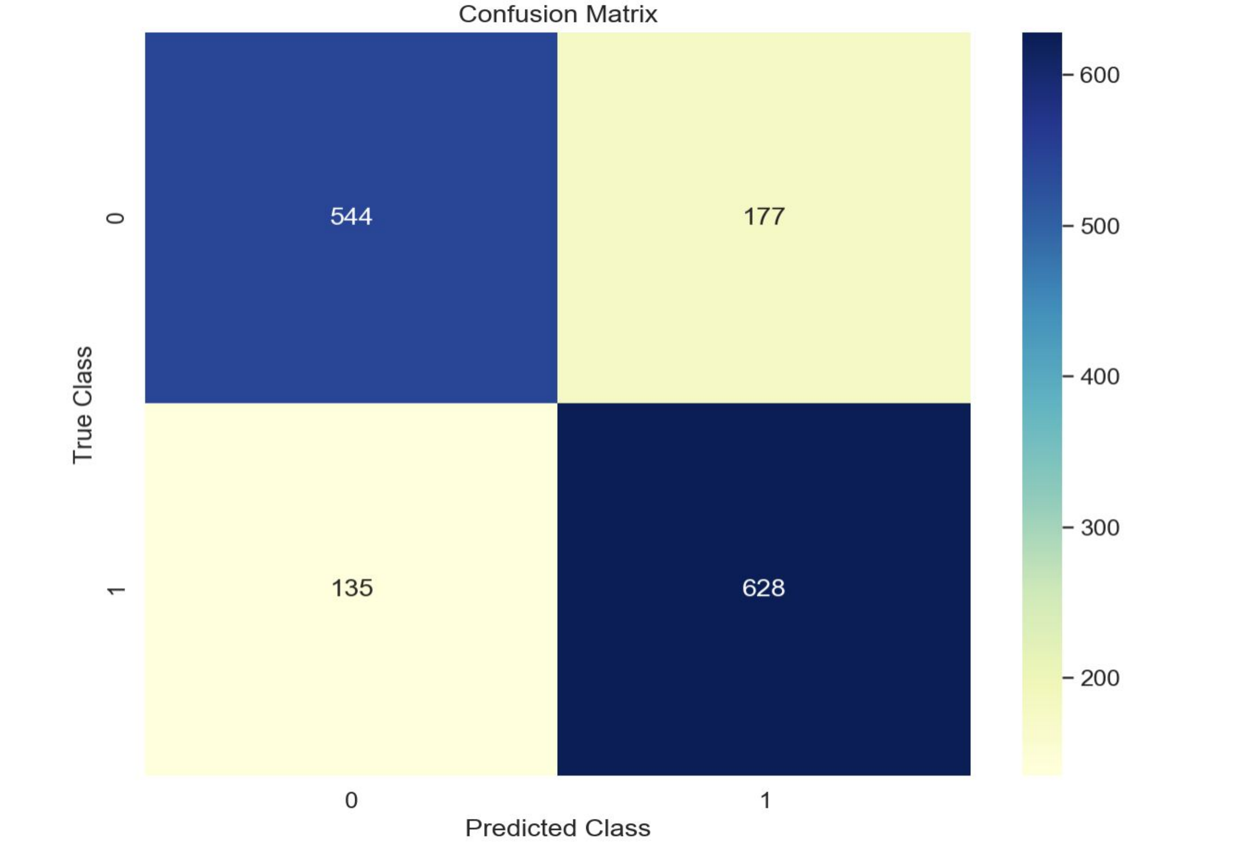
- All of my models performed at least .25 better than the baseline of .51.
- The difference in scoring was not that much, max .04.
- TfidfVectorizer with Logistic Regression gave me the best accuracy scores overall at .80.

Model Selection:

TfidfVectorizer & Logistic Regression

- I picked this combo because Tfidf increases the word's value proportionally to count, but is offset by the frequency of the word in the corpus. This allows you to pick up on the unique identifier words.
- Logistic regression was chosen because this is a binary classification problem and it makes the model's scores easily interpreted when comparing to the baseline.
- It has built in regularization which helps out quite a bit, my model uses Ridge.

Model Evaluation:



Confusion Matrix

True Class \ Predicted Class	0	1
0	544	177
1	135	628

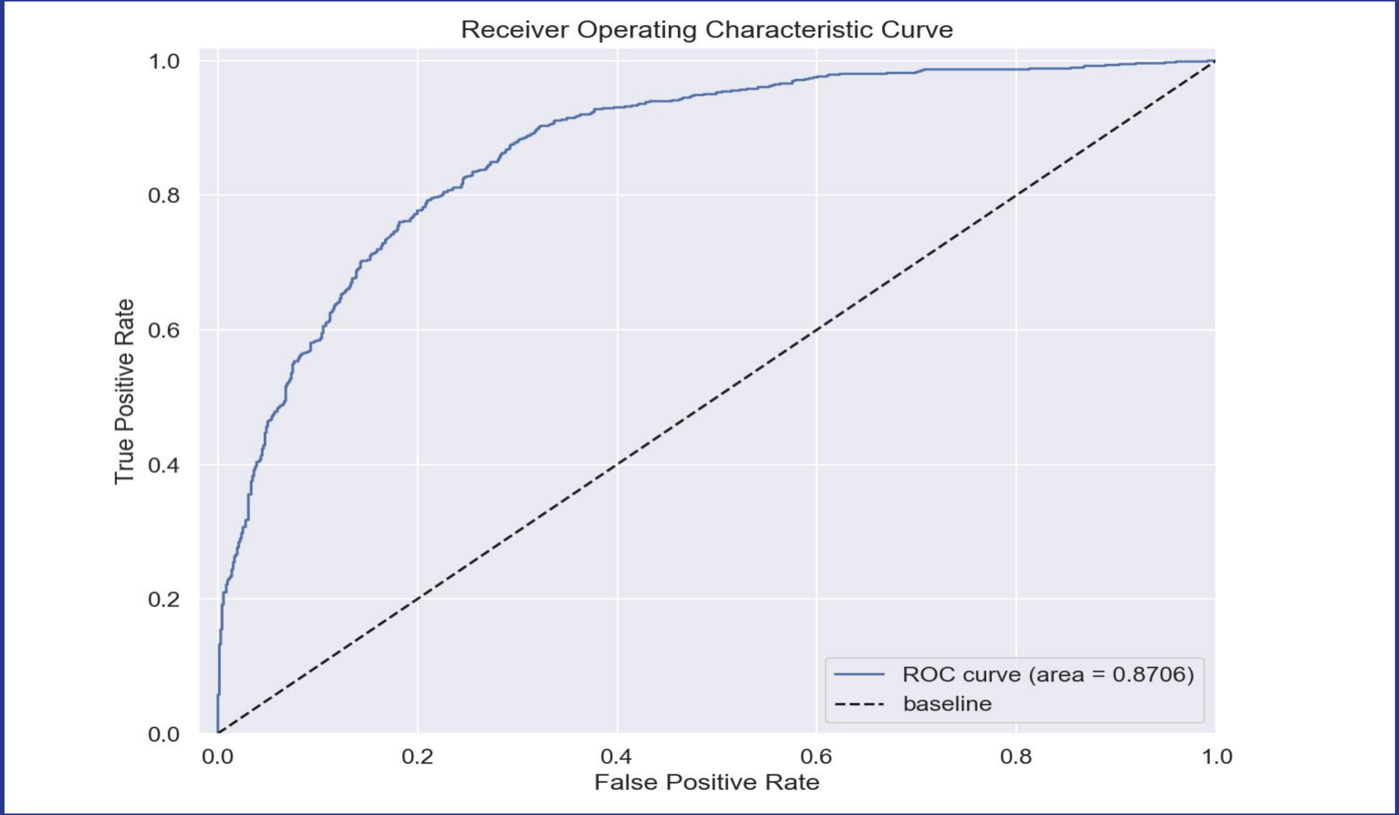
Model Evaluation:

- Accuracy is defined as the percentage of correctly classified instances.

- For this run it calculated 544 True Negatives (r/micro) and 628 True Positives (r/bio) correctly out of 1,484 total predictions.

- Which gives it an accuracy rate of .80

Model Evaluation:



Model Evaluation:

- My AUROC is better than the baseline and the ROC curve is closer to one than the baseline, so it safe to say this is a decent model.

Conclusion:

- My TfidfVectorizer & Logistic Regression model has an accuracy rate of .80 and it passed evaluation.
- This is good enough to help out with my subreddit differentiation.
- I do not need a super accurate model and since I am in a time crunch this should suffice.

Recommendations:

Now that I have a model ready to be put into action, the next step would be to build a function that would do the following and then implement it:

- Scrape the posts from the combined subreddit.
- Pass them into my predictive model then separate them into two dataframes based on their respective classes.
- I would then give the data to reddit so that they could put the posts back where they belong!