Progress Report: 10/14/2020

Approach

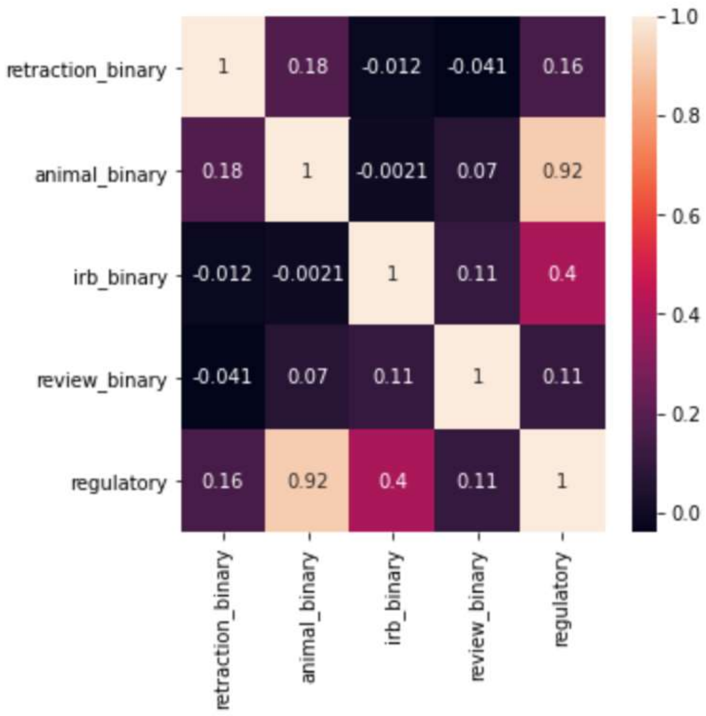
My approach to the project has been to collect the necessary data, clean it appropriately, explore it fully, and then conduct modeling to determine if it is possible to predict if a journal article will be retracted. To collect the necessary data, I used the biopython library to call the appropriate searches in the PubMed database. One search was conducted to collect retracted articles that were completely accessible, while the second search was conducted to collect the most recent articles from the accessible journals in the PMC database (the free full-text archive of the PubMed database). This process took over 16 hours to complete.

I cleaned the data in several ways. I concatenated the data for both retracted and non-retracted articles separately. In both new dataframes, I dropped all of the rows that had null values for the text of the article. I then added a "retraction_binary" column for both dataframes to indicate if the article had been retracted. I unpacked the keywords indicated in each article so that the format was changed from a large string to a list of individual words. I cleaned the text of the article by tokenizing the words with Regex (removing numbers, symbols, and non-English letter characters that were not connected to any letters), removing any words that had more than 45 characters, then lemmatized the remaining words. Lemmatized and non-lemmatized words were added to each dataframe as two separate columns for later exploration. The two dataframes were concatenated to create the total dataframe. Several unnecessary columns were dropped and duplicates were removed. CSV files were created throughout this process.

To explore the data, feature engineering was used in conjunction with data visualization. Several new columns of data were created and several barplots were created to compare the metadata of retracted articles to non-retracted articles. Naïve Bayes models were created to determine if it was possible to predict if a journal article will be retracted. The models were manipulated by changing the number of features in the vectorizer, the vectorizer type, the stop words list, and the test size.

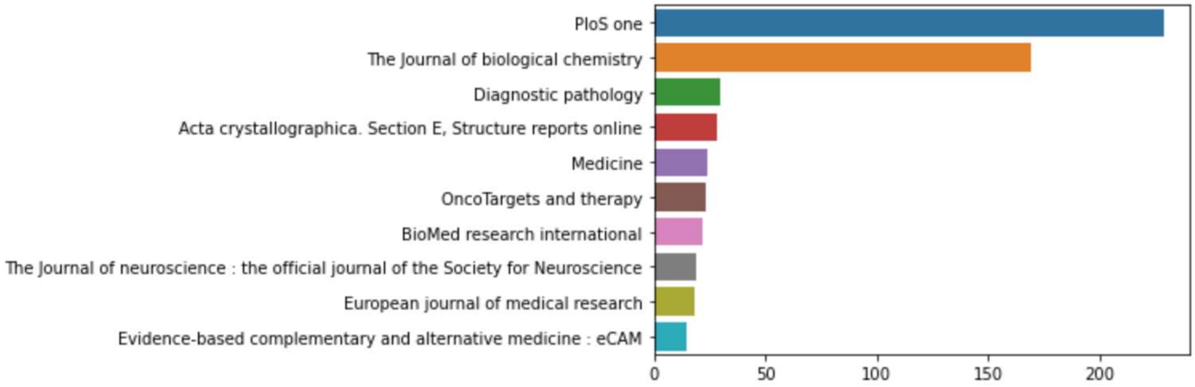
Initial Results

The data is sufficiently cleaned, although there are still symbol/letter combinations in the corpi. The EDA showed that there may be a very small correlation between an article containing animal products/studies and that article not being retracted. However, there is no correlation between an article being retracted and the article focusing on human-based studies or the article being a review article. The heatmap below shows the correlation coefficients for these topics:



There is also no correlation between an article being retracted and the character length or the word count of the article. The top 10 keywords for both retracted and non-retracted articles are very similar. Because the date was manipulated when pulling non-retracted articles, there are significant correlations between the retraction status of an article and year or month. Several of these barplots need to be remade using normalized data since the data is so imbalanced.

Most of the information I could obtain during EDA was that a significant portion of the retractions came from two different journals: PLOS ONE and the journal of biological chemistry. This was somewhat to be expected, as the probability of an article being retracted in a journal is typically 1 in 10,000. I wanted to see if the majority of the articles were from open source/free journals or certain publishers, but the results being so skewed to these two journals significantly influenced the data. Below is a barplot of the top 10 journals with retracted articles:



Setbacks

The biggest setback thus far is that the models created have not been sufficient to predict if an article will be retracted or not. The models that have been created so far are Naïve Bayes models. The **baseline** model accuracy is **69.1%** based on the randomly assigning values based on the probability of the article being retracted in this dataset. The below tables summarize the results of the models that have been created this far:

Only Changing Number of Features

Number of Features	Other Model Changes	Train Accuracy	Test Accuracy	True Positive	False Negative
5000	N/A	71.9%	69.5%	202	182
10000		75.0%	69.9%	198	186
20000		80.0%	70.6%	200	184
50000		85.7%	72.4%	195	189
100000		87.5%	72.6%	197	187

Adding More Stop Words

Number of Features	Other Model Changes	Train Accuracy	Test Accuracy	True Positive	False Negative
5000	Adding Stop Words	71.7%	69.3%	199	185
20000		80.1%	70.8%	200	184
100000		87.6%	72.6%	196	188

Changing to a TF-IDF Vectorizer

Number of Features	Other Model Changes	Train Accuracy	Test Accuracy	True Positive	False Negative
5000	Single words and Bi-grams	75.2%	73.2%	183	201
20000		78.5%	74.5%	135	249
100000		75.7%	73.2%	57	327
5000	Single words, Bi-grams, and Tri-grams	75.1%	73.2%	182	202
20000		78.2%	74.2%	135	249
100000		75.8%	73.8%	67	317

Changing Test Size

Number of Features	Other Model Changes	Train Accuracy	Test Accuracy	True Positive	False Negative
20000	Single words, test size = 0.2	79.7%	70.7%	144	163
20000	Single words, test size = 0.25	80.1%	70.8%	200	184
20000	Single words, test size = 0.3	81.1%	71.6%	237	224
5000	Single words and Bi-grams, test size = 0.2	74.9%	73.3%	144	163
5000	Single words and Bi-grams, test size = 0.25	75.2%	73.2%	183	201
5000	Single words and Bi-grams, test size = 0.3	75.1%	73.8%	219	242

The problem with all of these models is that they are overfit, ranging from slightly overfit to significantly overfit. Additionally, the models that are less overfit have a lower sensitivity score. This sensitivity score is as important as accuracy because the data is so imbalanced.

Future Work

Cleaning up the EDA visualizations will be necessary. More feature engineering may also prove necessary if I think of other ideas as I'm working. Markdown needs to be added to all notebooks that have been created. However, most of my future work will be in creating new models (such as random forests, SVMs, possibly classification neural nets, and models that combine various techniques). Unfortunately, the more complicated models that will be tried will likely increase the overfitting issue that is already apparent in the Naïve Bayes models. Because of this, it may be necessary to try to use the abstract as the corpi, followed with the same methods that have been used to clean/explore/model of the full text. I think that because the text corpi are so large, it is causing the model to be less accurate. Hopefully the abstract will have better results, as I currently have not thought of any other options to deal with the accuracy/overfitting issue.