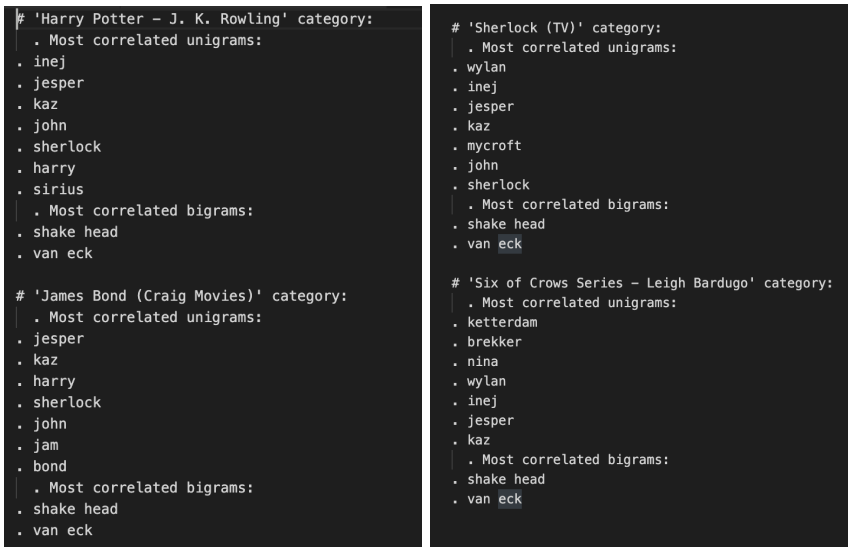
Waverly Wang
CS181Y
Dodds
21 April 2022

CS181Y Final Project Reflection: Fanfiction Classifier

My initial vision was to explore the topic of scraping fanfiction off of Archive of Our Own and analyzing the features of the fanfictions. At first, I was thinking of generating fanfiction with BERT and see if the classifier could detect which were machine made. As I started to work on importing libraries I thought it was easier to figure out how to create classifiers. After I found a notebook by Miguel Zafra who created a multi-class newspaper classifier for different genres, I used that to classify fandoms in fanfiction and ratings in fanfiction. I also found a guide by Amal Nair. I wanted to compare how those two classified and also compare different text-to-featurization representations. I decided to compare TF-IDF Word Vectorization using Random Forest and KNN models with BERT featurization and classification. The main goal was basically just experiment and see what was the best at classifying for both fandom and rating by using the features I got from scraping. I wanted to classify based on the fanfiction bodies but I did try tags later on.

I scraped 386 fanfictions that contained the 4 fandoms Sherlock (TV), Six of Crows (Leigh Bardugo), Harry Potter (J.K. Rowling), and James Bond (Daniel Craig Era). I wanted to begin by classifying fandom. I used Numpy and pandas for storing the data, Pickle for storying classifier, nltk for cleaning, sklearn for TF-IDF vectorization of the features. I used matplotlib and seaborn to visualize the information. I cleaned the data so it only included those 4 fandoms. I also cleaned the fanfictions removing stop words, lemmatizing and removing capitalization etc. With Miguel Zafra's notebook, I made progress on classifying both the fandoms and rating with TF-IDF vectorization. There were 300 features generated using TF-IDF vectorization of the fanfiction bodies I created. I then was able to see what unigrams and bigrams were most correlated with each category. Here were the results.



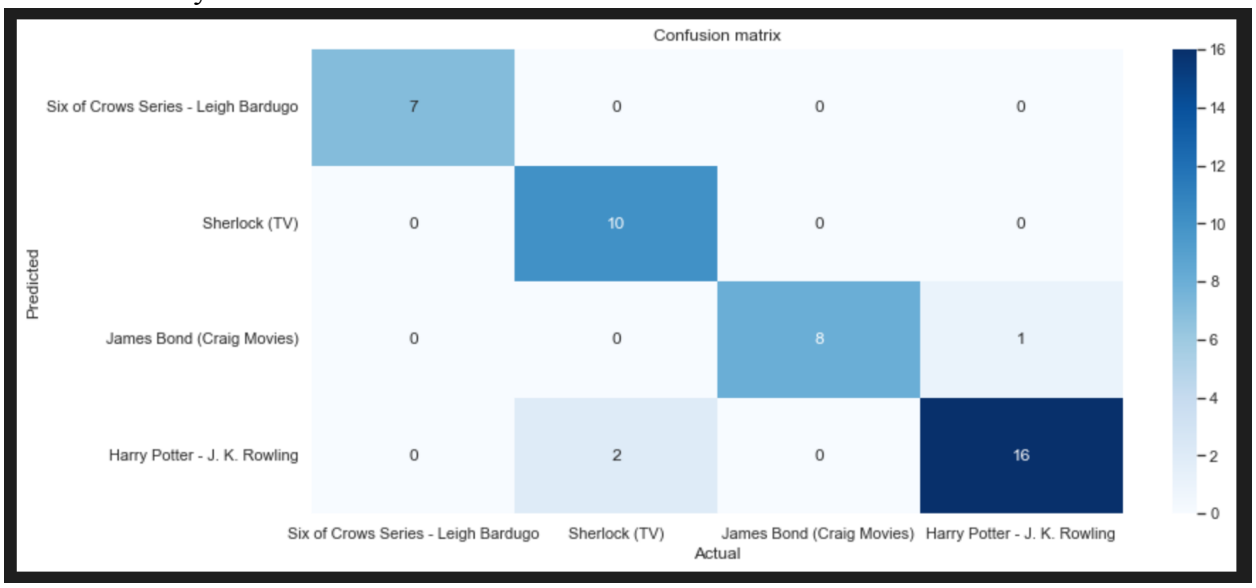
```
# 'Harry Potter - J. K. Rowling' category:
  . Most correlated unigrams:
. inej
. jesper
. kaz
. john
. sherlock
. harry
. sirius
  . Most correlated bigrams:
. shake head
. van eck

# 'James Bond (Craig Movies)' category:
  . Most correlated unigrams:
. jesper
. kaz
. harry
. sherlock
. john
. jam
. bond
  . Most correlated bigrams:
. shake head
. van eck
```

```
# 'Sherlock (TV)' category:
  . Most correlated unigrams:
. wylan
. inej
. jesper
. kaz
. mycroft
. john
. sherlock
  . Most correlated bigrams:
. shake head
. van eck

# 'Six of Crows Series - Leigh Bardugo' category:
  . Most correlated unigrams:
. ketterdam
. brekker
. nina
. wylan
. inej
. jesper
. kaz
  . Most correlated bigrams:
. shake head
. van eck
```

I noticed there were some characters categorized as the correct characters, but for some reason Six of Crows characters would show up in the wrong fandom (Kaz, Jesper, Inej) and I think this

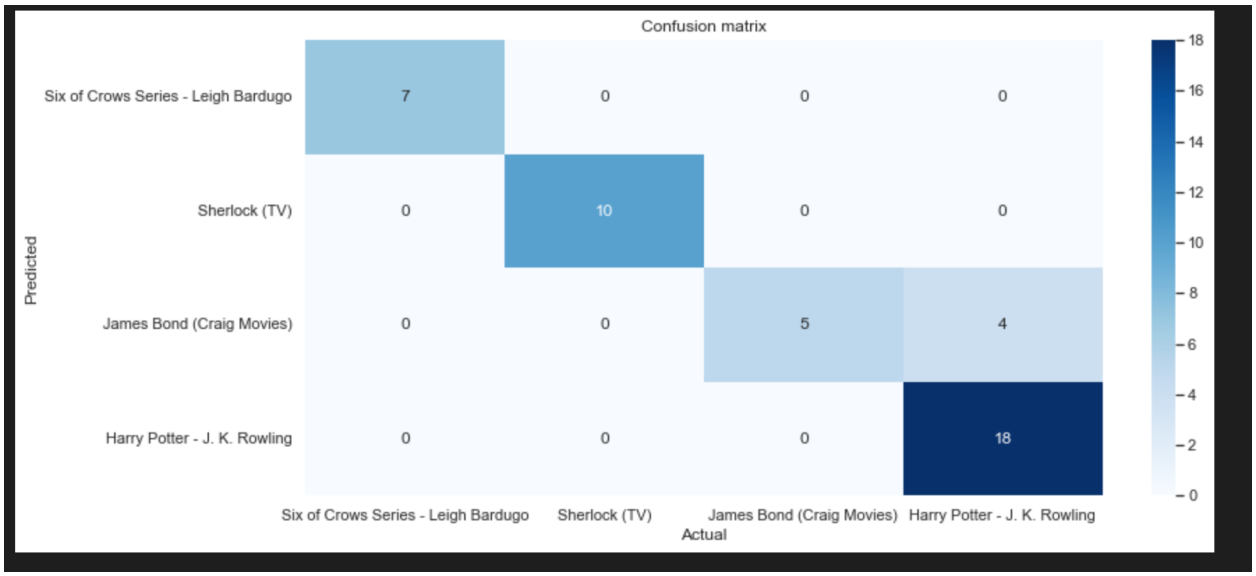
was due to the data having more Six of Crows text then the other fandoms. The Six of Crows fandom has all the right characters under it.

I made a KNN classifier. The results were that the training set accuracy was 100% and 93.18% test set accuracy. Here was the confusion matrix:



The true positive which are the diagonals are all darkened and show there was little to no confusion with the wrong fandom.

The random forest classifier model got 99.4% training accuracy and a 90.09 test set accuracy. Here was the confusion matrix for the RF classification.

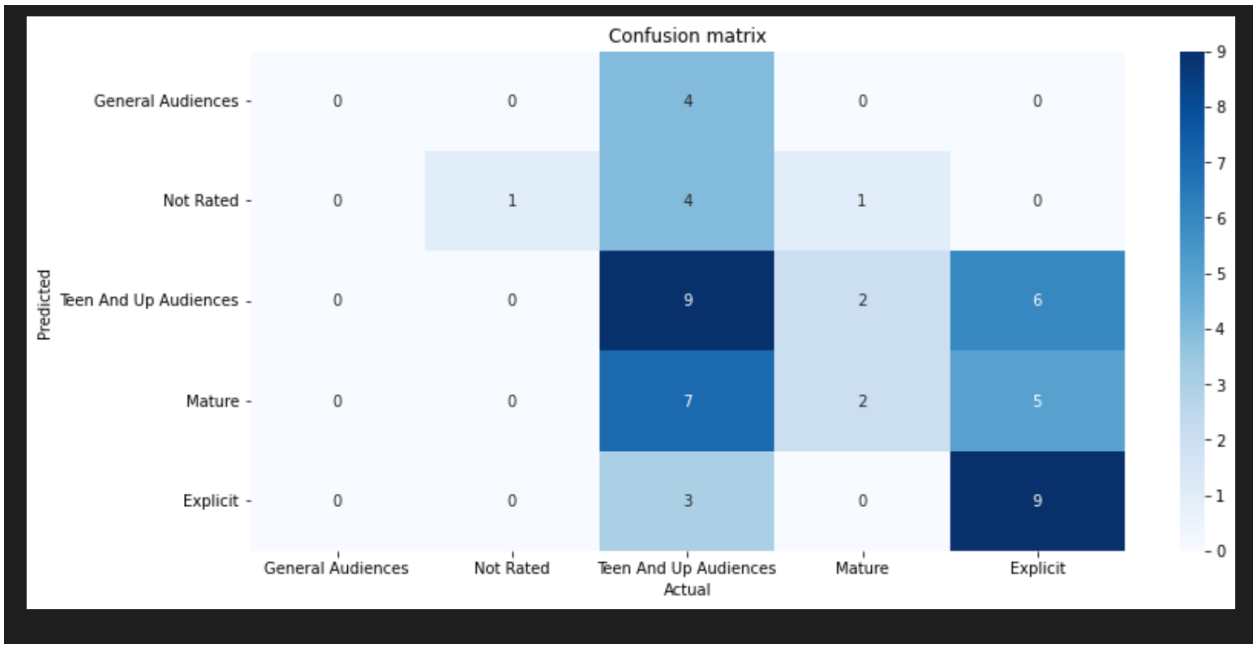


I then used 348 fanfictions to classify ratings. I used TF-IDF Vectorization and here are the most correlated unigrams and bigrams. It appeared that they didn't really correlate with the category and simply seemed to reflect what characters showed up the most for each category.

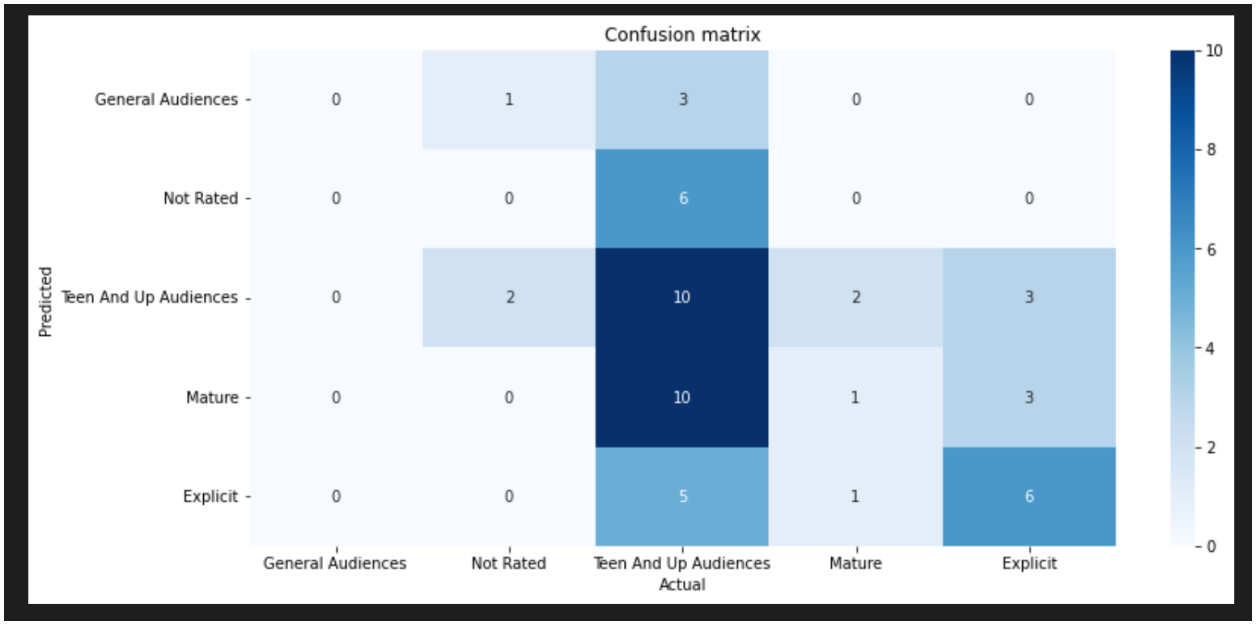
```
# 'Explicit' category:
  . Most correlated unigrams:
. nina
. lips
. sherlock
. press
. john
  . Most correlated bigrams:
. inej ghafa
. kaz brekker

# 'General Audiences' category:
  . Most correlated unigrams:
. soon
. fuck
. sleep
. gaze
. de
  . Most correlated bigrams:
. van eck
. jesper fahey

# 'Mature' category:
  . Most correlated unigrams:
. jesper
. mycroft
. de
. wylan
. jam
  . Most correlated bigrams:
. jesper fahey
. nina zenik
```

```
# 'Not Rated' category:
  . Most correlated unigrams:
. nina
. zenik
. sherlock
. jordie
. john
  . Most correlated bigrams:
. jesper fahey
. nina zenik

# 'Teen And Up Audiences' category:
  . Most correlated unigrams:
. nina
. sherlock
. fahey
. van
. wylan
  . Most correlated bigrams:
. kaz brekker
. van eck
```

It looks like the dataset is so small that it just associates the characters that show up the most in each rating category with each rating. It ended up doing really badly and part of it might be because I forgot to take out the non-rated category. Here are the results of the KNN classifier for the ratings. It got 43% for the training set accuracy and 39.6% for the test set accuracy. Here was the confusion matrix for the ratings with the KNN classifier:



Then I tried the Random Forest classifier. I got 100% for the training accuracy and 32% for the test set accuracy. There was obviously some overfitting happening. Also it seemed everything was being mistaken as Teen Audiences. Here was the RF confusion matrix for ratings:



I think it did badly because I didn't have enough fanfictions. On reflection, I realized that rating is very hard to predict because everyone has different opinions on what counts as General Audiences, Teen, Mature etc.

I needed a variety of fandoms and a large dataset, so that it could understand what in general is associated with each rating rather than just characters.

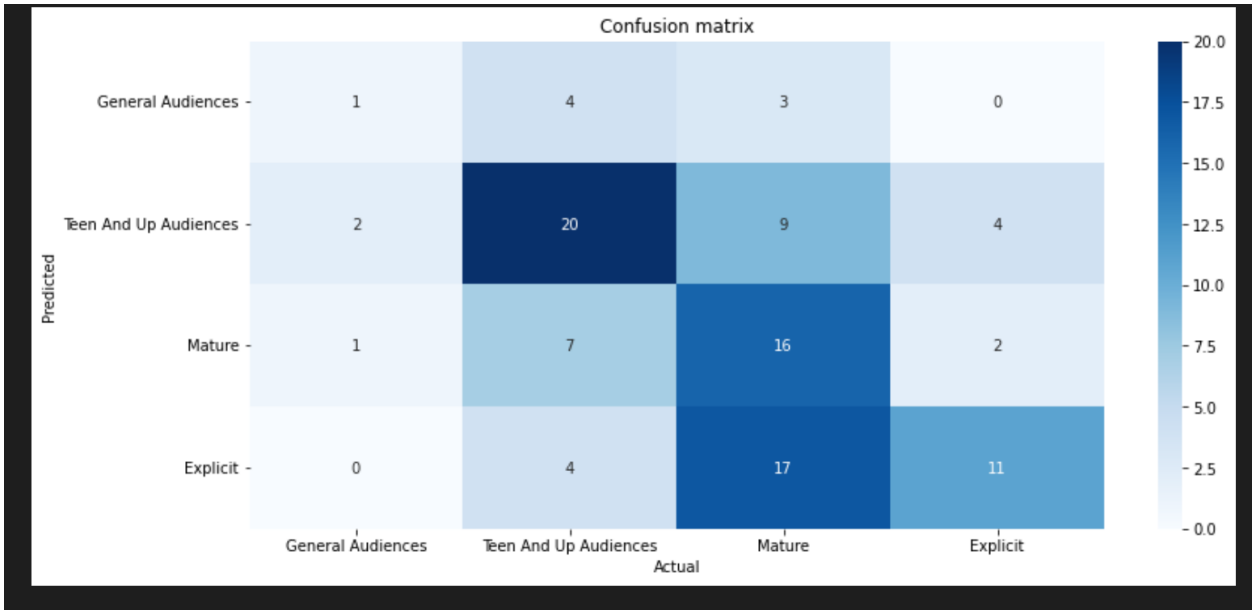
I tried to improve the rating classifier by having the classifier train on 500 fanfictions with a variety of fanfictions beyond 4 fandoms so ratings wouldn't be based on characters. It ended up doing even worse with about 20% test set accuracy.

I then thought maybe I should improve the rating classifier by having it be trained on the tags rather than the body of the fanfiction. Tags are ways writers can categorize their fanfiction by tropes or themes (fluff, hurt/comfort, romance, …etc.). This improved the rating classification to 47% which was better than guessing. I removed the Not-Rated category and made a KNN classifier. Here are the most correlated unigrams. They look more correlated!

```
# 'Explicit' category:
  . Most correlated unigrams:
. vaginal
. masturbation
. jobs
. anal
. oral
. porn
. sex
  . Most correlated bigrams:
. anal sex
. oral sex

# 'General Audiences' category:
  . Most correlated unigrams:
. rotting
. family
. soulmates
. sex
. romance
. fluff
. childhood
  . Most correlated bigrams:
. romance fluff
. fluff humor
```

```
# 'Mature' category:
  . Most correlated unigrams:
. ptsd
. childhood
. tension
. masturbation
. burn
. story
. porn
  . Most correlated bigrams:
. post traumatic
. sexual tension

# 'Teen And Up Audiences' category:
  . Most correlated unigrams:
. go
. sexual
. sad
. teen
. relationships
. smut
. sex
  . Most correlated bigrams:
. eventual smut
. teen romance
```

Here is the confusion matrix for the rating classifier on tags for 500 fics:



As you can see, there is less confusion about what's Teen Audiences are! Teen audiences and mature audiences are more distinct. I think Mature is being confused with Explicit. I decided not to do a RF classifier for tags since it usually did worse.

I then repeated this process but with BERT. With BERT, I didn't clean as much because I researched that can remove the context between words and BERT relies on that. The fandom classifier got 68% evaluation accuracy using 512 tokens and 3 epochs. It got 34% with an outside test set it had never encountered before. I also made a predictor that can predict fandom based on sentences. The rating did even worse with 34% evaluation accuracy with 512 tokens, 3 epochs, 1.2 loss. It got 18% on the outside test set. It can also predict based on sentences. I think the reason it's doing pretty badly is due to the fact BERT can only take so many tokens. 512 tokens is not enough to predict fandom and rating that well.

For the ways I'd like to extend the project if I had more time, I think I'd like to try more classifiers. In the newspaper classifier guides I used, there were lots of models trained like SVM, MultinomialNM, Multinomial LogReg, GBM etc. I didn't try those and I'm curious how those would do. I also would like to make an app that could scrape a fanfic and then tell it's rating and fandom! Find a way to put in text and then have it predict the fandom or rating based off of that! I think I could get there because the newspaper classifier guide also made an app to scrape news and classify it.

If I could tackle the project differently I would want to tell myself that once one model does well, you don't have to keep trying to improve it with other models. I think once I had realized fandom was classified with 93% I shouldn't have tried to spend time scraping an additional dataset to train more to improve it. I realize now 93% is very good! I would spend more time trying to create a dataset that had enough fanfictions so the models could have enough data to go off of because I think there still wasn't enough. I also feel like it would have been good to have a separate notebook for cleaned data and a separate model for building the model because I put them both in the same file and it became really long and annoying to keep loading

the dataset to clean it. Also I realized I forgot to take out the non-rated category for the dataset and that would have helped the classification of rating.