Sally Nguyen | Andrewlu Xiao | Brian Kosiadi | Garrett Chaffey | Han Mai

Spring 2022 BANA 275 : NLP Final Project

Classifying Adult vs. Youth Anime Using Synopsis and Genre

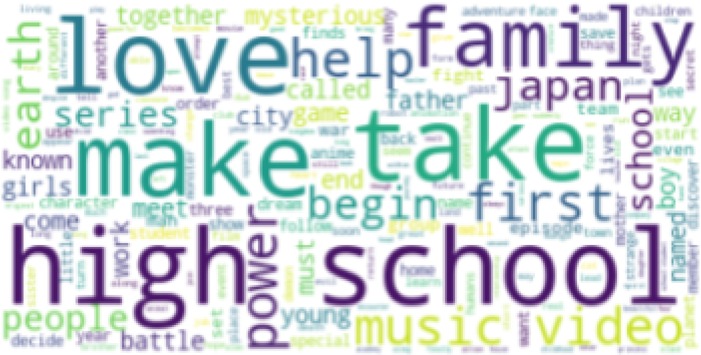
1. Project Description and Objectives

Our project focuses on filtering anime synopsis to predict anime rated for children versus anime rated for adults. Since many people just see animation and cartoons as 'childish', the surveillance on what is actually being watched tends to be more lax than live-action content which could be problematic when many animes have explicit content in terms of gore, violence, and sex. The use-case for this would be to incorporate parental controls for child safety and protection. This could be used to block off access to anime that is deemed inappropriate or traumatizing for children. Taken from the opposite point of view, another use-case of this would be filtering and recommending anime for older viewers who don't want to sift through children's shows to find one they would be interested in.

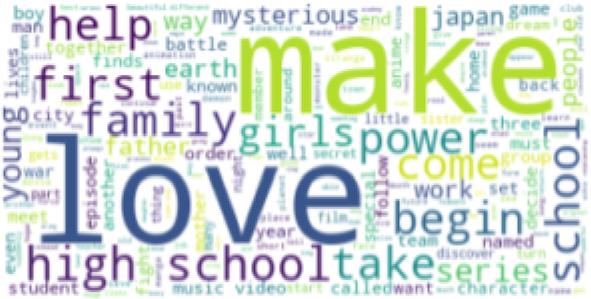
2. Data Descriptions

Our dataset was gathered from Kaggle and titled Anime Data Set with Reviews - MyAnimeList; it contains 16,000 unique observations and 12 columns scraped from two tables. After exploring all the variables, we decided to use the anime title, synopsis, genre, and content rating. In order to classify our anime as either adult or non-adult, we used two criterias. First, we created a list of adult genres that included hentai, yaoi, yuri, shounen ai, shoujo ai, and ecchi. On MyAnimeList each anime has both a list of genres it falls under as well as an age rating, for each show if it belonged to any of the above mentioned "adult" genres (hentai, yaoi, yuri, ecchi etc etc) it was flagged as adult content. Second, any anime that had a content rating of PG-13 and above was also flagged as adult content. The rest would be classified as non-adult or children's anime. Using the above two classification rules, we added an additional column to our dataset to denote whether each anime in question belonged to the adult content category or child content category. Animes which were categorized as “adult” were labeled with a 1, while non-adult/childrens animes received a 0. We then cleaned the text of the anime synopses dataset to remove stop words and proper nouns (like names of people/places/things) to try to minimize noise in the data (ie names/proper nouns are unlikely to provide insight into whether a show is for mature audiences or not) while at the same time reducing the number of features (ie unique words/terms) we would need to examine.

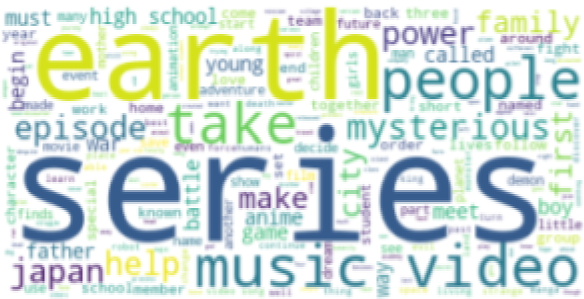
As part of our exploratory data analysis, we created word clouds to try and visualize some of the most frequently occurring terms within the dataset we examined. Below are three word clouds created from our data, both taken as a whole and subsetted by our own label of “adult” or not.



Word Cloud of All Anime



Word Cloud of Adult Anime



Word Cloud of Children's Anime

3. Methodology

Vectorizers Used:

1. CountVectorizer
2. TF-IDF (term frequency inverse document frequency)
3. Embedding

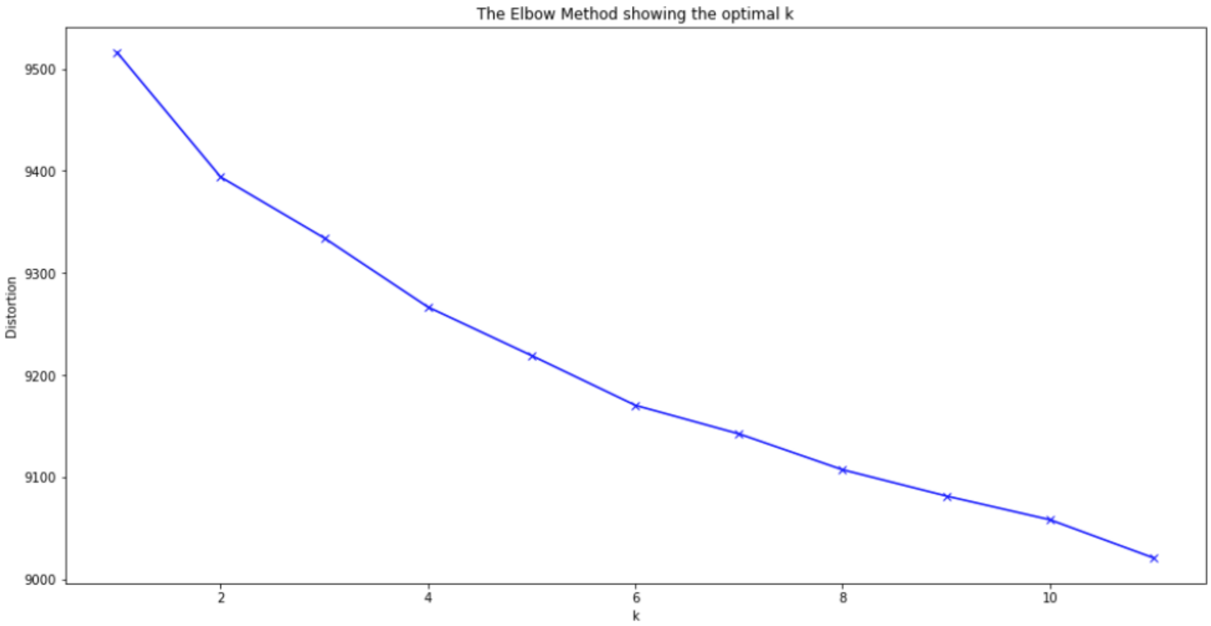
Models Used:

1. KMeans
2. KNN
3. DBScan
4. Agglomerative Clustering
5. Naive Bayes Classification
6. Compound and Polarity Scoring
7. Random Forest Classification
8. Logistic Regression

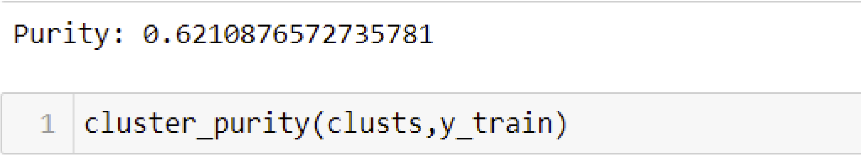
Model Evaluations:

K Means Clustering

After splitting the data into train and test sets, we ran the elbow method and Kneelocator to select the optimal number of clusters to cluster the data - 9.

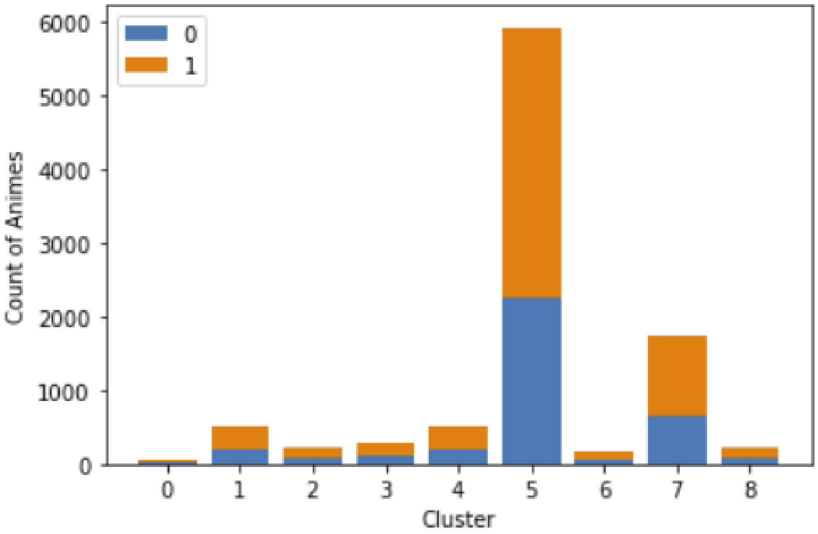


Using sklearn KMeans to fit to the weighted vectorized train dataset, “tf-idf”, we were able to calculate a cluster purity score of **62.1%**. We also got a silhouette score to better understand the clusters. With a silhouette score of .00675, we can assume that there were overlapping clusters.



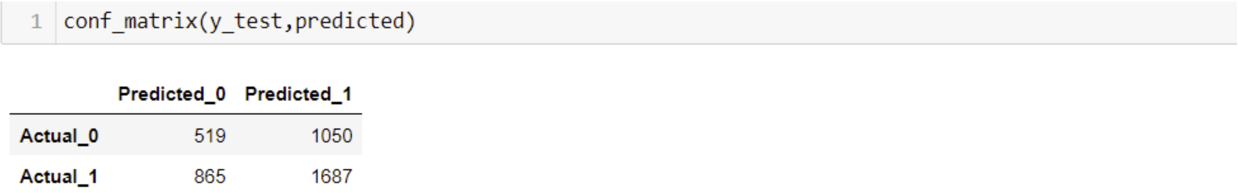
```
Purity: 0.6210876572735781
```

```
cluster_purity(clusts,y_train)
```



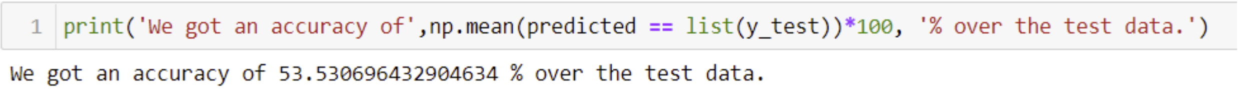
K-Nearest Neighbors

After Kmeans, we also ran tf-idf vectorization on the test set. We then created the KNeighborsClassifier, setting n-neighbors to 10, and fit it to the train set. After which, we used that knn model to predict on the test set and created a confusion matrix out of the results. From that, we got an accuracy score of **53.53%**.



```
conf_matrix(y_test,predicted)
```

	Predicted_0	Predicted_1
Actual_0	519	1050
Actual_1	865	1687



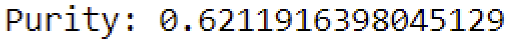
```
print('We got an accuracy of',np.mean(predicted == list(y_test))*100, '% over the test data.')
```

```
We got an accuracy of 53.530696432904634 % over the test data.
```

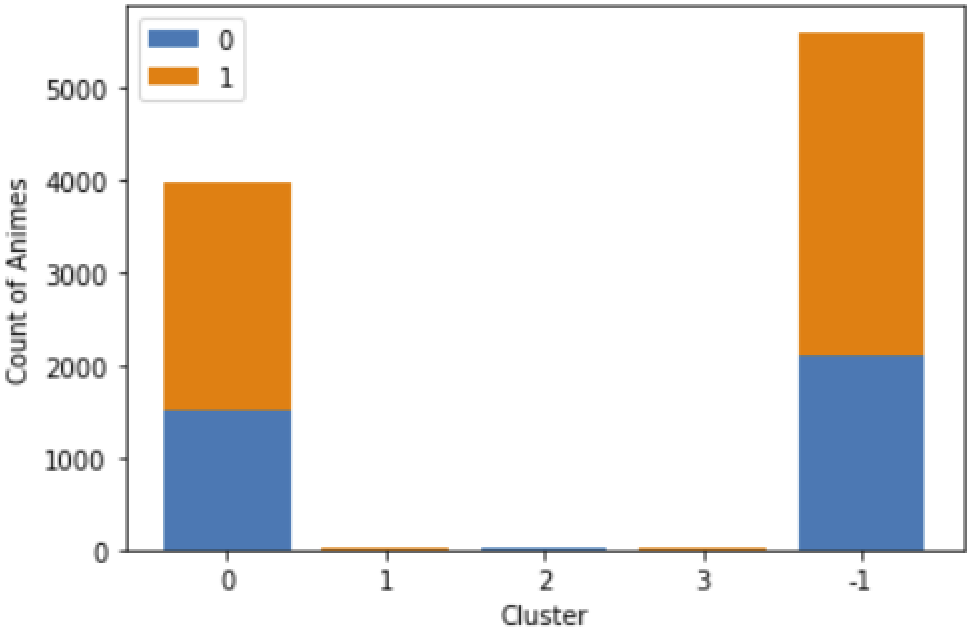
DBScan

We then used DBScan or density-based spatial clustering to try a different method of clustering our data. DBScan groups data points together that are close based on a set minimum number of points per cluster as well as distance between points - using the

Euclidean distance method. It uses 2 parameters: eps and minPoints. Eps are how close points should be to one another to be included in a cluster. MinPoints is the minimum number of points that is required to create our cluster. For our model, we ran a function that would test out different ep-ranges and number of minimum points to find our optimal eps and minpoint. Our optimal eps was 0.86.The optimum minimum point which we called our minsample was 10 data points. We then fit the DBScan model with eps set as 0.86 and minimum sample as 10 to our vectorized train set and calculated our cluster purity score. Here we can see there were two main clusters that held the majority of data points.

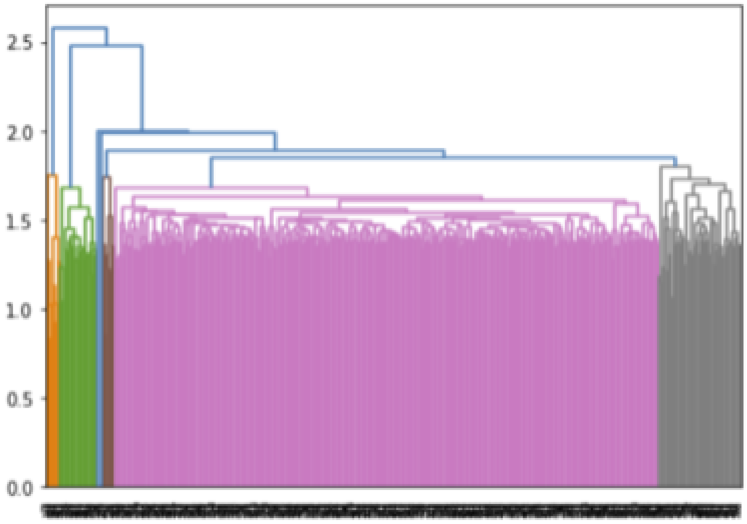


```
Purity: 0.6211916398045129
```

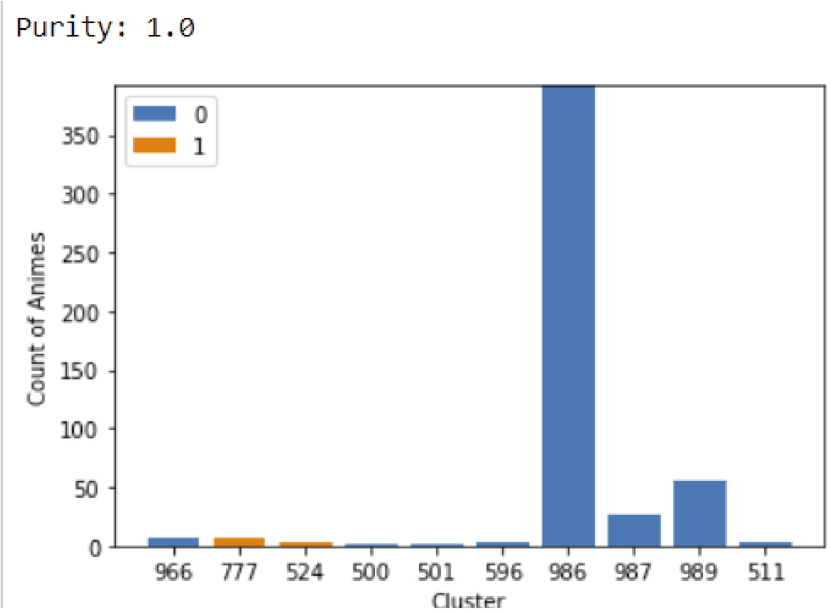


Agglomerative Clustering

In keeping with clustering experimentation, we also tried agglomerative clustering. This started off by having all the data points in its own cluster and then successively merging similar, close points together until it forms into an overarching tree of clusters that contains all the points, also known as a dendogram. Because the agglomerative clustering starts off with each point in its own cluster and then merges similar points together, the purity of the agglomerative clusters are much higher than the other clustering methods.



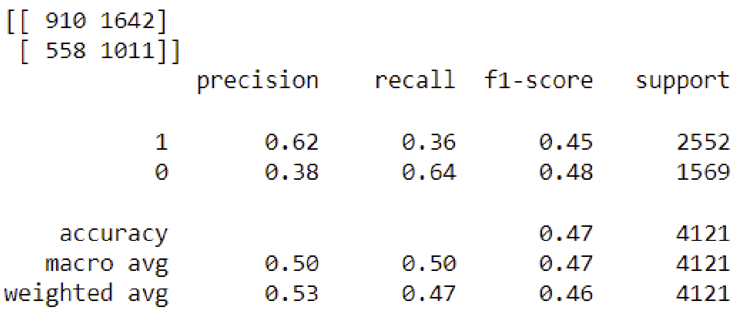
Dendogram of Anime Data Points



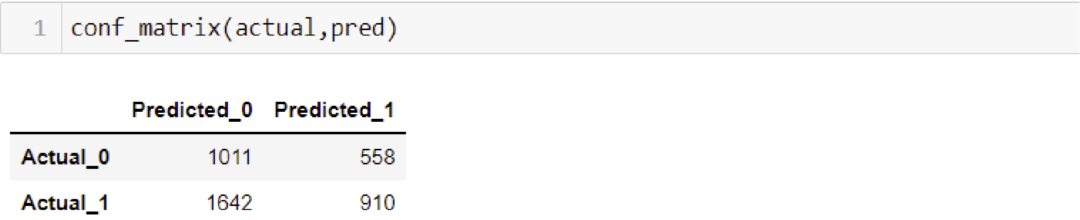
```
Purity: 1.0
```

Naive Bayes

Beyond clustering, we also tried to classify the animes using a Gaussian Naive Bayes model which assumed that our data follows a normal distribution. Similar to KNN, we fitted the model to the train data and then used that to predict our test dataset. We then created a confusion matrix from the results and calculated the model accuracy over the test data which was **46.61%**.

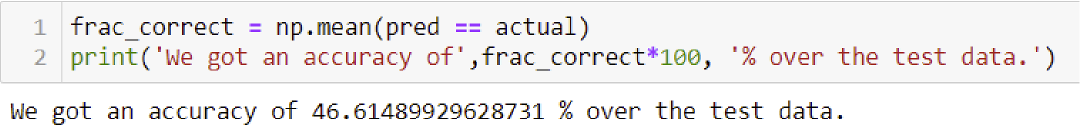


```
[[ 910 1642]
 [ 558 1011]]
              precision    recall  f1-score   support

           1       0.62      0.36      0.45      2552
           0       0.38      0.64      0.48      1569

    accuracy                           0.47      4121
   macro avg       0.50      0.50      0.47      4121
weighted avg       0.53      0.47      0.46      4121
```



```
conf_matrix(actual,pred)
```

	Predicted_0	Predicted_1
Actual_0	1011	558
Actual_1	1642	910

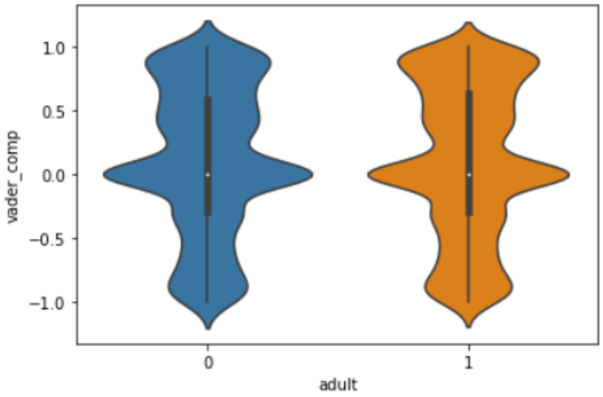


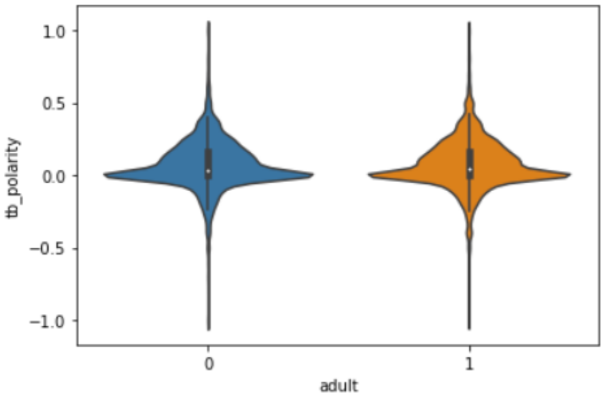
```
frac_correct = np.mean(pred == actual)
print('We got an accuracy of',frac_correct*100, '% over the test data.')
```

```
We got an accuracy of 46.61489929628731 % over the test data.
```

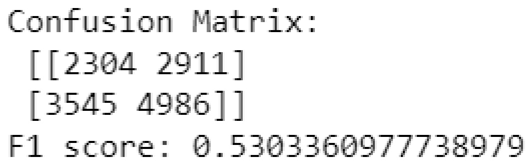
Compound and Polarity

Another method we used was checking the Vader Compound and Polarity scores based on synopsis. However, looking at the Vader Compound score graph, there was not a big difference between the adult (1) and non-adult group (0).





We tried to test our hypothesis that the synopsis for adults will have higher polarity (more extreme) than kids synopsis. Therefore, from these polarity scores, we binned them into 2 groups: 0 (kid) and 1(adults) and then checked with our classifier columns in the dataset for the accuracy score. We were able to get an F1 score of **53.03%**.

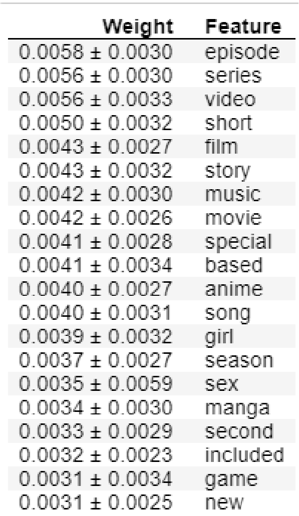


```
Confusion Matrix:
 [[2304 2911]
 [3545 4986]]
F1 score: 0.5303360977738979
```

Random Forest Classifier

Besides all the above models, we also tried a supervised ensemble method - Random Forest Classifier in classifying adults and non-adult anime based on synopsis. We applied lemmatization and tf idf on both train and test set before running the RFC model. By setting the n-estimators to 200, we get the accuracy of predicting the train set (**96.6%)** and test set (**60.5%)**.

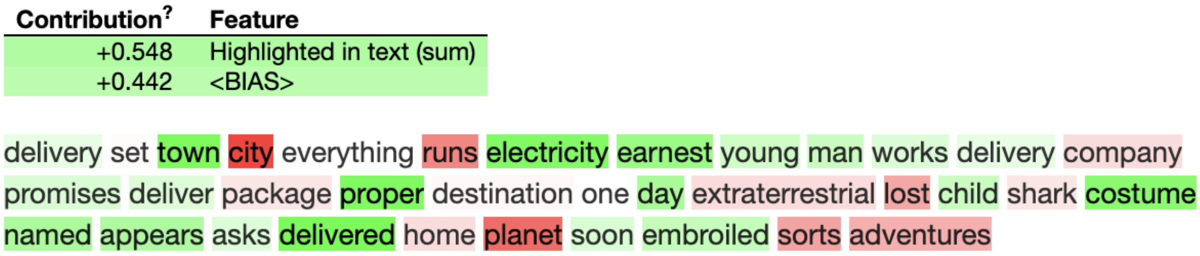
We also showed weights of the top 20 features and got the below results:



Weight	Feature
0.0058 ± 0.0030	episode
0.0056 ± 0.0030	series
0.0056 ± 0.0033	video
0.0050 ± 0.0032	short
0.0043 ± 0.0027	film
0.0043 ± 0.0032	story
0.0042 ± 0.0030	music
0.0042 ± 0.0026	movie
0.0041 ± 0.0028	special
0.0041 ± 0.0034	based
0.0040 ± 0.0027	anime
0.0040 ± 0.0031	song
0.0039 ± 0.0032	girl
0.0037 ± 0.0027	season
0.0035 ± 0.0059	sex
0.0034 ± 0.0030	manga
0.0033 ± 0.0029	second
0.0032 ± 0.0023	included
0.0031 ± 0.0034	game
0.0031 ± 0.0025	new

Logistic Regression

Another method we tried was logistic regression. We performed logistic regression using both features gained from CountVectorizer and TFIDFVectorizer. After testing different parameters for each regression model we found that TFIDF consistently had a slightly higher accuracy when compared to CountVectorizer. Going further with TFIDF as our selected model we supplemented our findings with the eli5 package which showed the top 20 classifier weights for the logistic regression model. The overall accuracy for the training set was **61.97%** and the accuracy for testing was **62.35%**. An interesting graphic of using eli5 was being able to see which words specifically contributed to the text being classified as adult or kid friendly as can be seen in the example below.

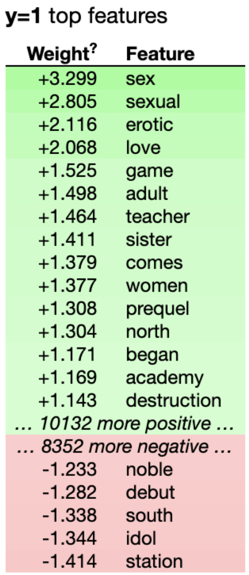


Contribution?	Feature
+0.548	Highlighted in text (sum)
+0.442	<BIAS>

delivery set town city everything runs electricity earnest young man works delivery company promises deliver package proper destination one day extraterrestrial lost child shark costume named appears asks delivered home planet soon embroiled sorts adventures

In this figure you can see that some of the words are highlighted with different shades of green and red. The green words weight the synopsis more towards an “adult” classification while the red words weight the synopsis towards youth.

Another chart garnered from eli5 was the words weight chart wherein each word was given a weight and said weight determined how impactful the word was towards classifying the synopsis as adult or child friendly. As we dummified our adult category to be 1 and our child friendly category to be 0, The negative weights can be interpreted such that those corresponding words would cause a synopsis to more likely be classified as child friendly.



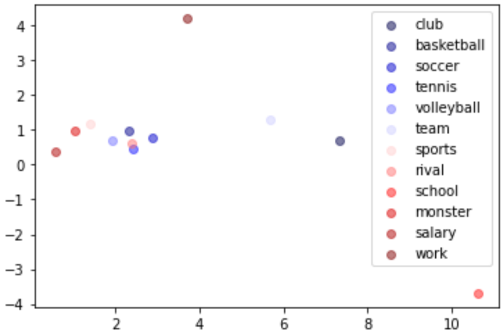
y=1 top features

Weight?	Feature
+3.299	sex
+2.805	sexual
+2.116	erotic
+2.068	love
+1.525	game
+1.498	adult
+1.464	teacher
+1.411	sister
+1.379	comes
+1.377	women
+1.308	prequel
+1.304	north
+1.171	began
+1.169	academy
+1.143	destruction
… 10132 more positive …	
… 8352 more negative …	
-1.233	noble
-1.282	debut
-1.338	south
-1.344	idol
-1.414	station

Overall the key findings is that sex, sexual, and erotic would heavily weigh the anime towards adult and station, idol, and south would heavily weigh the anime toward youth.

5. Embeddings Model

We also explored the usage of embeddings as a way to possibly boost our classification accuracies. We used the Gensim package to create a model which identified relationships of each word and how they were connected to other terms as defined in our corpus of anime synopses.



In the above graphic, we took a subset of some of the words within our corpus and visualized their distance to each other. As can be seen, the terms which were highly related to sports are all close together (basketball, volleyball, soccer etc) while terms like “work”, “salary” and “club” appear farther away. Additionally, as part of our “smell test” to ensure our outputs were not erroneous we examined a few words and their relations as defined within the text. According to our corpus, “class” - “adult” = “student”, and “friend” + “enemy” = “rival” which we found to be appropriate given these terms often appear in similar feature spaces. Finally, we explored the usage of creating embeddings vectors for use in classification and after running multiple KNN models we found that the highest accuracy which we achieved was 57% on the test data set.

6. Challenges (and How We Overcame Them)

The biggest challenges we faced during this project were: 1) a slightly imbalanced dataset, 2) substantial noise within the data itself due to the inclusion of names of people, places, or things alongside japanese words that were written in English within the synopsis and 3) significant overlap between the wording used in both adult and child friendly anime synopses, ie synopses for even pornographic content only rarely used sexual/sexually related wording/terminology.

After we categorized our dataset based on both the ratings derived from MyAnimeList and the genres each anime was classified under, we were left with 8,529 animes labeled as adult content and 5,209 animes which were labeled as child friendly. As the division between counts for each class was not too egregious we ensured that our train and test splits each had an equal ratio of adult vs child friendly content; especially after we ran our first few classification models and noticed a suspiciously high accuracy in prediction which turned out to be due to the models essentially labeling everything as adult content which while technically correct was due to the fact that there was a lack of child friendly data samples and an overpowering amount of adult data points.

Our next challenge was the inclusion of words in the synopses list which would not inform our classifications. Features like the names of people, places, or things for the most part do not provide any indication of whether or not the show in question is adult content or child friendly. In addition many of the synopses included English spellings of Japanese words which again for our purposes were not useful for the classification of shows based on the contents of their synopses. In order to overcome this, we used PyEnchant, a Python library containing English dictionaries, and checked each unique term within the aggregated synopses list. If the term in question was not within PyEnchant’s English dictionary it was removed. Before this cleaning, our feature list contained 42,345 unique words, however after running each word through the PyEnchant dictionary and removing each feature that was not explicitly an English word our final cleaned dataset consisted of 23,345 unique words (a reduction of 303,890).

Our final challenge was the existence of “grey area” features. Our dataset was pulled from a public website lacking age verification or content filtering of any kind. We hypothesize that because of this, there are certain rules which each anime page must adhere to in order to retain their page on the host website. We noticed that even in strictly 18+ content the wording used in the synopses was innocuous and taken as a whole, similar to the wording used in child friendly content. In order to account for this at least partially, we identified “red flag” words that appeared far more often in adult content compared to child friendly content. As was previously touched on in the logistic regression section, the words “sex”, “sexual”, “game”, and “erotic” were all words that appeared more often in explicit (and specifically lewd) content. And these words (and other similarly adult themed words) were used to weight a classification towards being adult.

7. Conclusion

While we found logistic regression to be the most effective classification model by testing accuracy, we still found the other models to have practical usages. We observed clustering algorithms with purity score as well as what major themes or features the clusters comprised of. With classification methods such as random forest and logistic regression, we can discover what specific keywords have greater weights in classifying anime.

While optimizing accuracy would be the typical next step, another area we could experiment with is the precision recall threshold. In this use case, type 1 and type 2 errors should not be treated equally, as the type 2 error of falsely labeling adult content as safe for children would have much greater repercussions than falsely labeling content safe for children as adult content. Thus, we would seek to improve recall more than precision. Ultimately, our project still showed signs of success in identifying and classifying adult and children content, and could be continually built upon and optimized for a practical use-case.

Blog:
https://medium.com/@bkosiadi/classifying-adult-vs-youth-anime-using-synopsis-and-genre-5ae7b27ac829

Github:
https://github.com/AndrewluXiao/Anime_Classifier