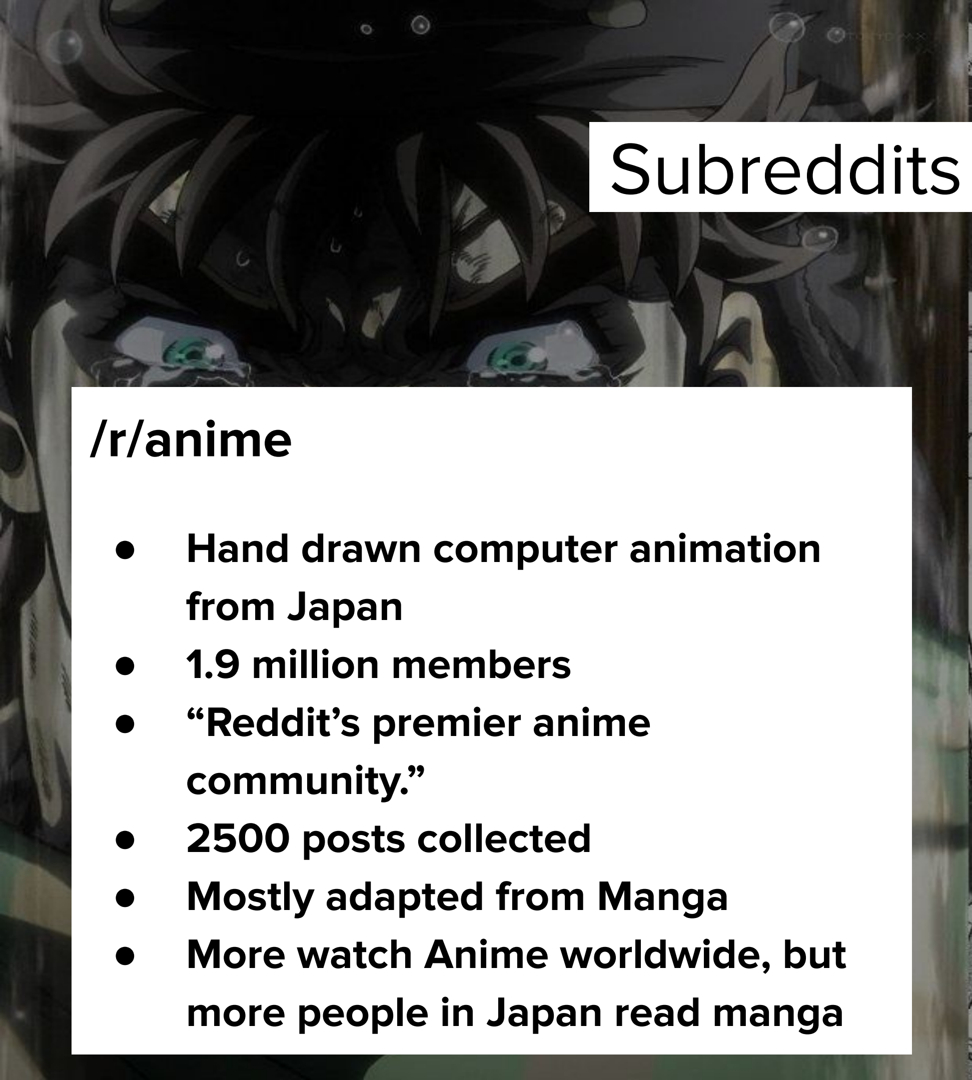
Predicting Subreddit of Origin by Title Using Pushshift API

Clement Gendler

Is it feasible to use modeling to predict the original subreddit of a post? If so, which model performs best?

Subreddits of Interest



/r/anime

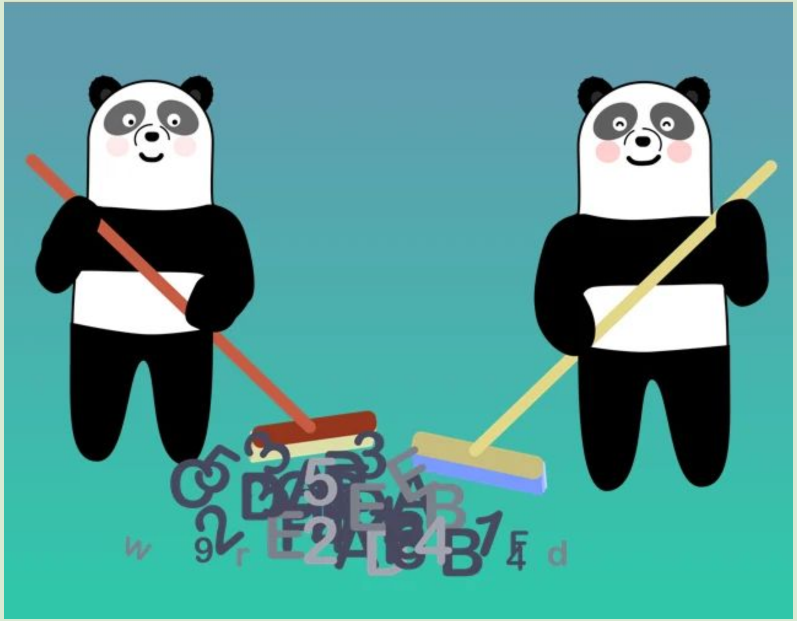
- Hand drawn computer animation from Japan
- 1.9 million members
- “Reddit’s premier anime community.”
- 2500 posts collected
- Mostly adapted from Manga
- More watch Anime worldwide, but more people in Japan read manga

/r/manga

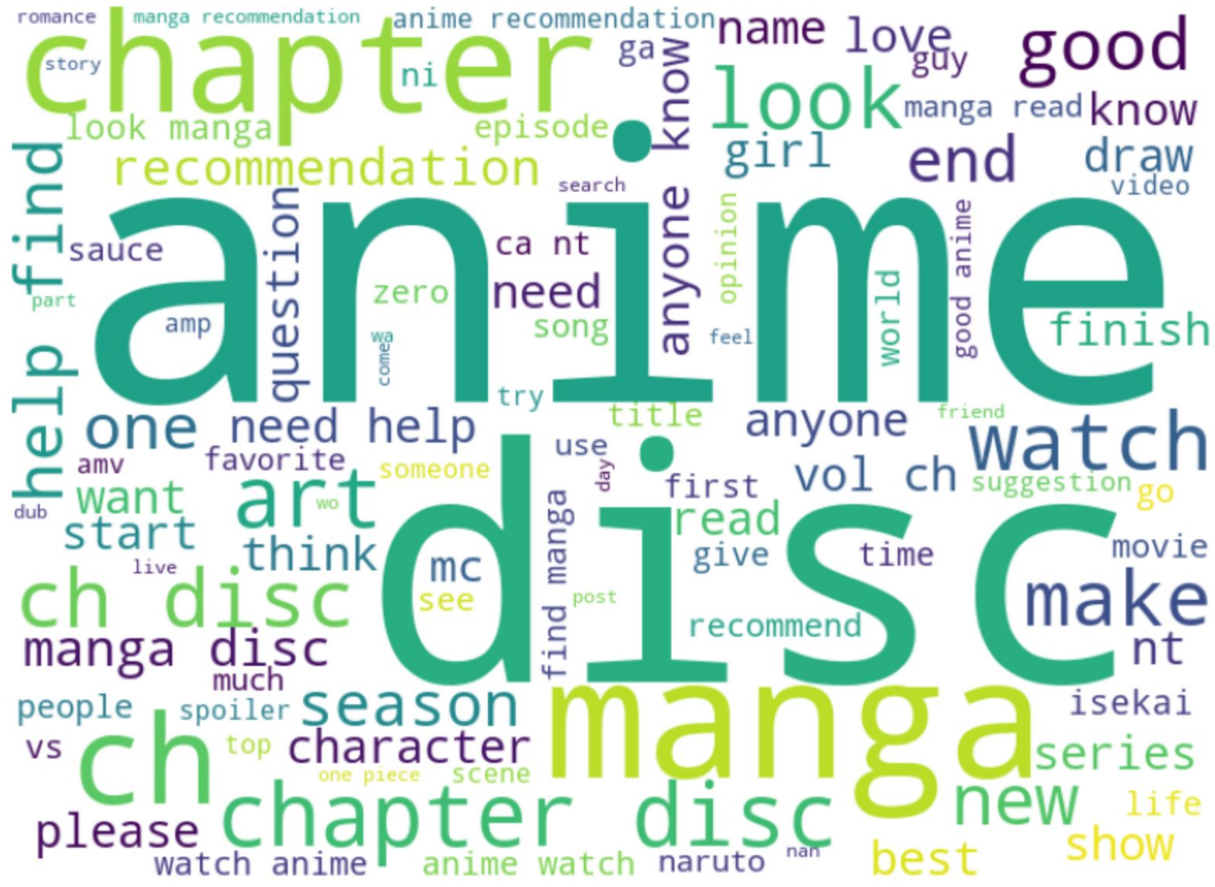
- Style of Japanese comic books and graphic novels
- 1.2 million members
- “Everything and anything manga! Discuss weekly chapters, find/recommend a new series to read, post a picture of your collection, lurk, etc!”
- 2500 posts collected

Data Cleaning & EDA

- Combined Anime and Manga reddit posts into one data set
- Removed nulls
- Replaced [removed] and [deleted] tags with empty string
- Created ‘label’ column to function as target for modeling
- Created Word Cloud to visualize key terms for both subreddits



Key Terms for Both Subreddits

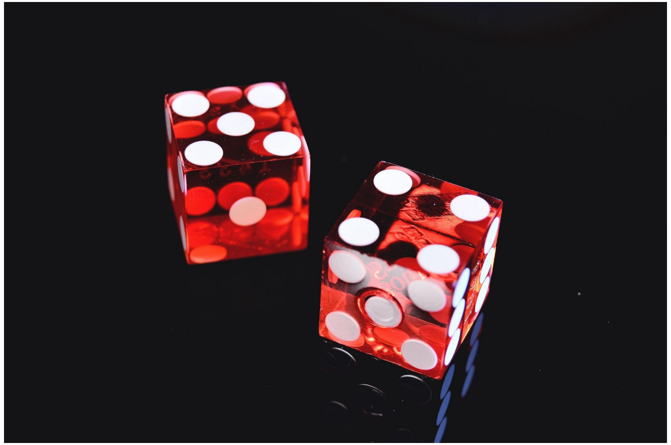


Preprocessing

- Used RegExpTokenizer to manually tokenize title and post columns
- Lemmatized tokens by reducing verbs
- Mapped preprocessing and lemmatized verbs functions to title and post columns
- Looped through title and post columns, using .join to put tokenized words back together

velvet cu
verb /v3:b/ noun
a word that tells you
something is or does.
and 'be' are all verbs
erdict /'v3:dɪkt/
jury in a

Modeling (I)



- Used train_test_split with a test size of 0.33 and a random state of 42
- Vectorized with **CountVectorizer**, using stop_words and strip_accents to further clean data
- Tested with Logistic Regression and Naive Bayes Classification Model

Modeling (II)

Logistic Regression:

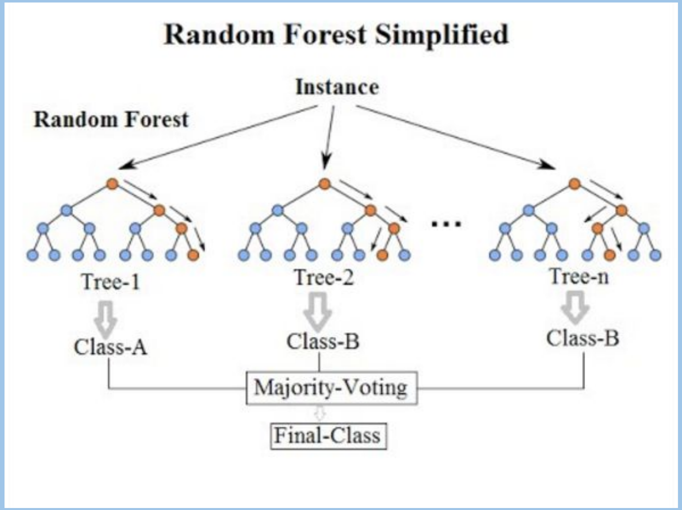
- Predicts probability based on label (classifying subreddit: (Anime 0, Manga 1)
- Score: 0.8866 or roughly **89%**

Naive Bayes:

- Multinomial model that predicts probability based on discrete features and probability of training data
- Score: 0.8587 or roughly **86%**

Moving Forward

- Test with both title and post
- Automate to further increase accuracy with pipeline
- Experiment further with stemming
- Use more models, like Random Forest and Decision Trees
- Implement GridSearchCV in more models
- Display results with confusion matrix
- Explore other columns like score



Conclusions & Recommendations

- Logistic Regression is the most accurate model (**89%** vs 86%)
- Using multiple models, even of the same type, can be beneficial to maximize accuracy
- CountVectorizer can automate a large portion of manual cleaning without reducing accuracy