IMPROVING THE AUTO-MODERATOR FOR R/BOOKS & R/WRITING

Project 3: Web APIs & NLP

Rachel Z. Insler

DSIR - 22221E

1. BACKGROUND

What is reddit.com?

“

Each day, millions of people around the world post, vote, and comment in communities organized around their interests.

Source: ***redditinc.com***

BACKGROUND INFORMATION

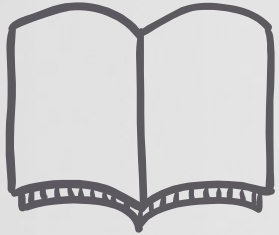
- community = ‘subreddit’
- largely user-edited via up- & down-votes on posts
- also enforcement by human & auto-moderators

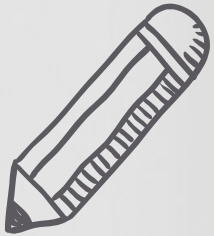
Source: Ionos.com

430 million reddit users

What could possibly go wrong?

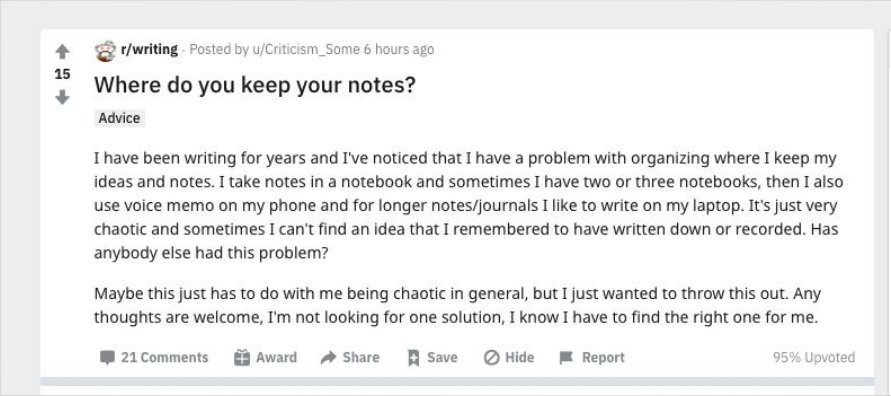
A TALE OF TWO SUBREDDITS...





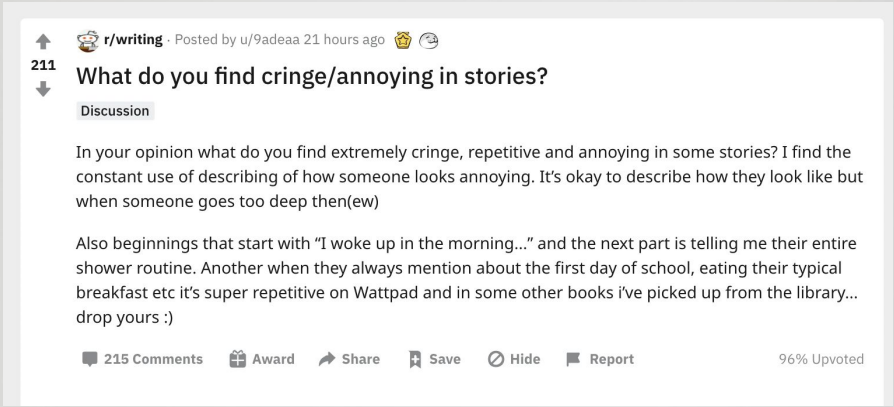
r/writing

- 1.7m members
- “Discussions about the writing craft”



r/writing

- 1.7m members
- “Discussions about the writing craft”



r/books

- 19.2m members
- “It is our intent and purpose to foster and encourage in-depth discussion about all things related to books, authors, genres, or publishing in a safe, supportive environment.”

r/books

- 19.2m members
- “It is our intent and purpose to foster and encourage in-depth discussion about all things related to books, authors, genres, or publishing in a safe, supportive environment.”

PROBLEM STATEMENT

This project aims to develop a model that accurately determines whether attempted posts to reddit.com/r/writing belong on that site or are better-suited to reddit.com/r/books.

Ultimately, this model could be incorporated into the auto-moderator algorithm on both subreddits, and used to suggest that the user post on the alternative subreddit.

2.
DATA COLLECTION

Using pushshift.io API

120,000

Observations from pushshift api: timestamp, text, subreddit

60,000

per subreddit (books & writing)

30,000

each submissions and comments

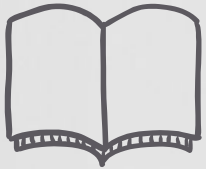
3.
CLEANING & PRE-PROCESSING

CLEANING

- removed [removed] and [deleted] observations
- removed duplicate posts
- created ‘text’ column for both submissions and comments
- removed rows with fewer than 10 chars of text

PRE-PROCESSING

- binarize target variable: {'books' : 0, 'writing': 1})
- removed special characters
- to lemmatize or not to lemmatize?
- tokenized
- removed English stopwords
  - re-checked for empty rows



4.
MODELING

And modeling and modeling and modeling

MODELING

- created a smaller, balanced sample with 2,500 observations per class
  - null model = 0.5 accuracy

- used train-test split

- Model A: Count Vectorization + Logistic Regression
  - **Transformation Hyperparameters:**
    - max_features = 5_000, max_df = .95, min_df = .05, ngram_range=(1,2)
  - **Estimator Hyperparameters:** Default
  - **Results:**
    - Training Score: 0.757
    - Testing Score: 0.762
    - Cross-val score is: 0.746

And when I woke up...

Model Name	Transformer	Estimator	Transformer Hyperparameters	Estimator Hyperparameters	GridSearch	Train	Test	CV Score
A	CountVectorizer	LogisticRegression	max_features = 5_000, max_df = .95, min_df = .05, ngram_range= (1,2)	Default	No	0.757	0.762	0.746
B	CountVectorizer	LogisticRegression	max_features = 10_000, max_df = 0.9, min_df = 2, ngram_range= (1,2)	C = 0.1	Yes	0.895	0.811	0.784
C	CountVectorizer	LogisticRegression	max_features = 15_000, max_df = 0.8, min_df = 2, ngram_range= (1,2)	C = 0.1	Yes	0.818	0.786	0.770
D	CountVectorizer	LogisticRegression	max_features = 10_000, max_df = 0.7, min_df = 2, ngram_range= (1,2)	C = 0.1	Yes	0.816	0.785	0.770
E	CountVectorizer (added 12,500 rows to each class)	LogisticRegression	max_features = 10_000, max_df = 0.7, min_df = 2, ngram_range= (1,2)	C = 0.1	Yes	0.831	0.811	0.806
F	TFIDF	LogisticRegression	max_features = 12_500, ngram_range = (1,2)	C = 1	Yes	0.889	0.832	0.829
G	TFIDF	LogisticRegression	max_features = 12_500, max_df = 0.7, min_df = 3, ngram_range = (1,3)	C = 1	Yes	0.889	0.831	0.829
H	TFIDF	LogisticRegression	max_features = 10_000, max_df = 0.65, min_df = 5, ngram_range = (1,4)	C = 1	Yes	0.885	0.830	0.829
I	TFIDF	LogisticRegression	max_features = 10_000, max_df = 0.65, min_df = 5, ngram_range = (1,4)	C = 0.1	No	0.836	0.814	0.786
J	CountVectorizer	BernoulliNB	Default	Default	No	0.828	0.717	0.724
K	CountVectorizer	BernoulliNB	max_features = 10_000, max_df = 0.7, min_df = 2, ngram_range = (1,2)	Default	No	0.799	0.719	0.721
L	TFIDF	BernoulliNB	max_features = 10_000, max_df = 0.7, min_df = 2, ngram_range = (1,4)	Default	No	0.792	0.714	0.718
M	CountVectorizer	AdaBoostClassifier with LogReg	max_features = 10_000, max_df = 0.7, min_df = 2, ngram_range= (1,2)	n_estimators=150	No	0.813	0.787	0.780
N	CountVectorizer	RandomForestClassifier with Decision Tree	max_features = 10_000, max_df = 0.7, min_df = 2, ngram_range= (1,2)	n_estimators=150, max_depth = None	Yes	0.995	0.797	0.795
O	CountVectorizer	ExtraTreesClassifier with DecisionTree	max_df = 0.7, min_df = 2, n_gram_range = (1,2)	n_estimators=300, max_features=auto	No	0.997	0.784	0.754
P	CountVectorizer	Support Vector Machine	max_df = .7, min_df = 2, ngram_range=(1,2))	Default	No	0.876	0.814	0.807

MODELING: THE CANDIDATES

Transformer	Estimator	Transformer Hyperparams	Estimator Hyperparams	Train	Test	CV Score	Grid Search
Count Vectorizer	**Logistic Regression**	**max_features = 10_000 max_df = 0.7 min_df = 2 ngram_range = (1,2)**	**C = .01**	**0.831**	**0.811**	**0.806**	**Yes**
TF-IDF	**Logistic Regression**	**max_features = 12_500 ngram_range = (1,2)**	**C = 1**	**0.889**	**0.832**	**0.829**	**Yes**

5. SELECTION & INTERPRETATION

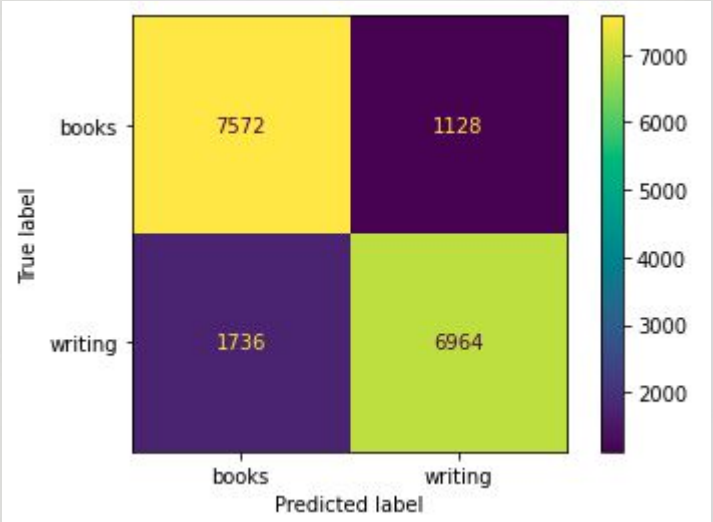
Picking a winner based on a larger dataset (87,000 posts)

UPDATED MODELING RESULTS

Transformer	Estimator	Transformer Hyperparams	Estimator Hyperparams	Train	Test	CV Score	Grid Search
Count Vectorizer	**Logistic Regression**	**max_features = 10_000 max_df = 0.7 min_df = 2 ngram_range = (1,2)**	**C = .01**	**0.841**	**0.829**	**0.826**	**Yes**
TF-IDF	**Logistic Regression**	**max_features = 12_500 ngram_range = (1,2)**	**C = 1**	**0.845**	**0.836**	**0.831**	**Yes**

PREDICTIONS & RESULTS FOR TEST DATA

- **Model more likely to say ‘books’ than writing**
- **Overall accuracy - 84%**
  - **14,536 correctly classified**
  - **2,864 misclassified**
    - **2,189 of those comments!**
- **How does the model do on a submission-only dataset?**



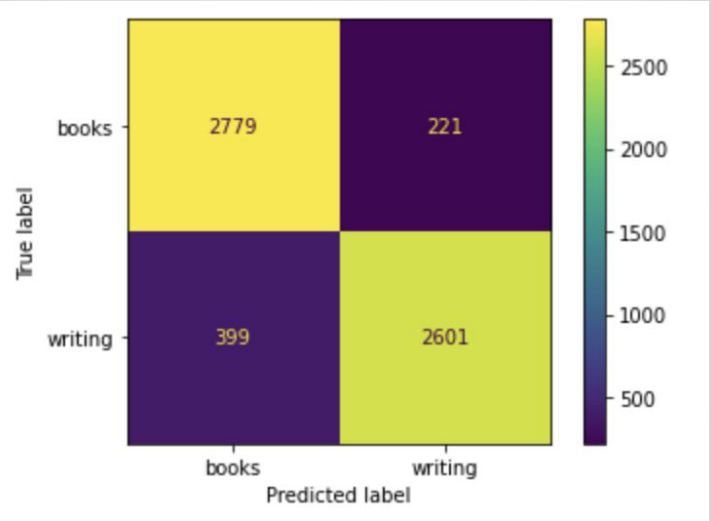
MODELING RESULTS: SUBMISSIONS ONLY

Transformer	Estimator	Transformer Hyperparams	Estimator Hyperparams	Train	Test	CV Score
TF-IDF	**Logistic Regression**	**max_features = 12_500 ngram_range = (1,2)**	**C = 1**	**0.906**	**0.897**	**0.895**

PREDICTIONS & RESULTS FOR SUBMISSIONS

- **Model still more likely to say ‘books’ than writing**

- **Overall accuracy - 90%**
  - **5,380 correctly classified**
  - **620 misclassified**



6. CONCLUSIONS & FURTHER EXPLORATION

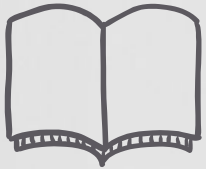
CONCLUSION

Our model determines with 89.7% accuracy whether attempted submissions to the ‘writing’ and ‘books’ subreddits belong on those sites or are better suited for the other site.

Ultimately, this model could be incorporated into the auto-moderator algorithm on both subreddits, and used to suggest the alternative subreddit to users.

NEXT STEPS

- Run all candidate models on submission-only dataset
- Explore where and when classification errors happen
  - There were posts that the model incorrectly classified with near-certainty
- Explore specific words with strong positive and negative coefficients

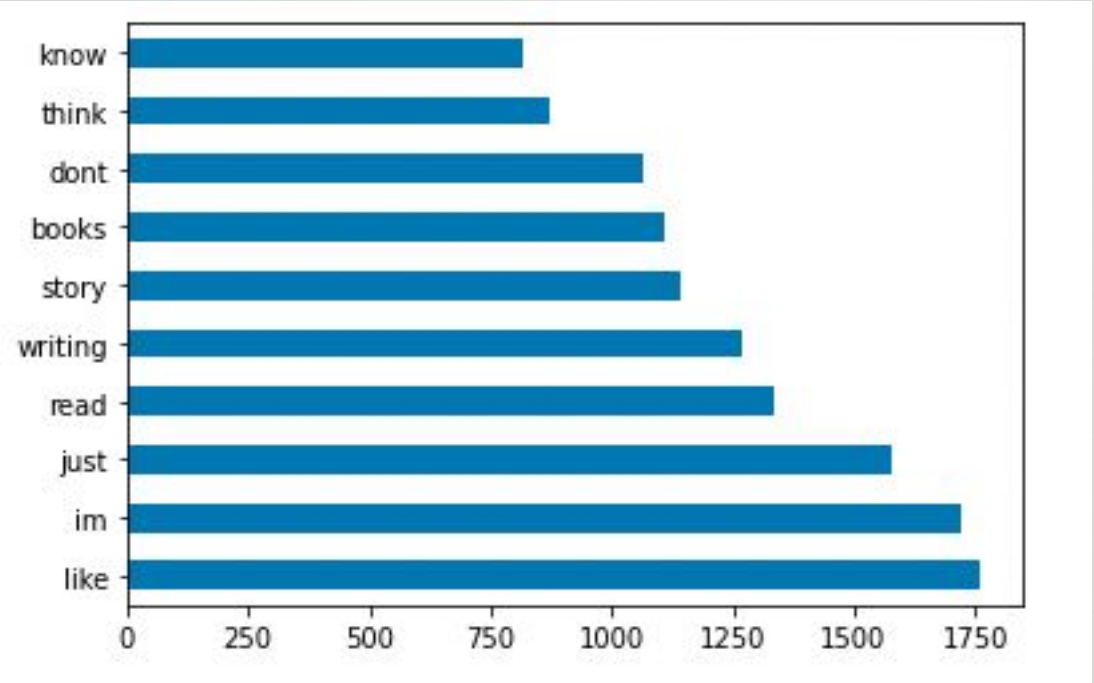


THANKS!

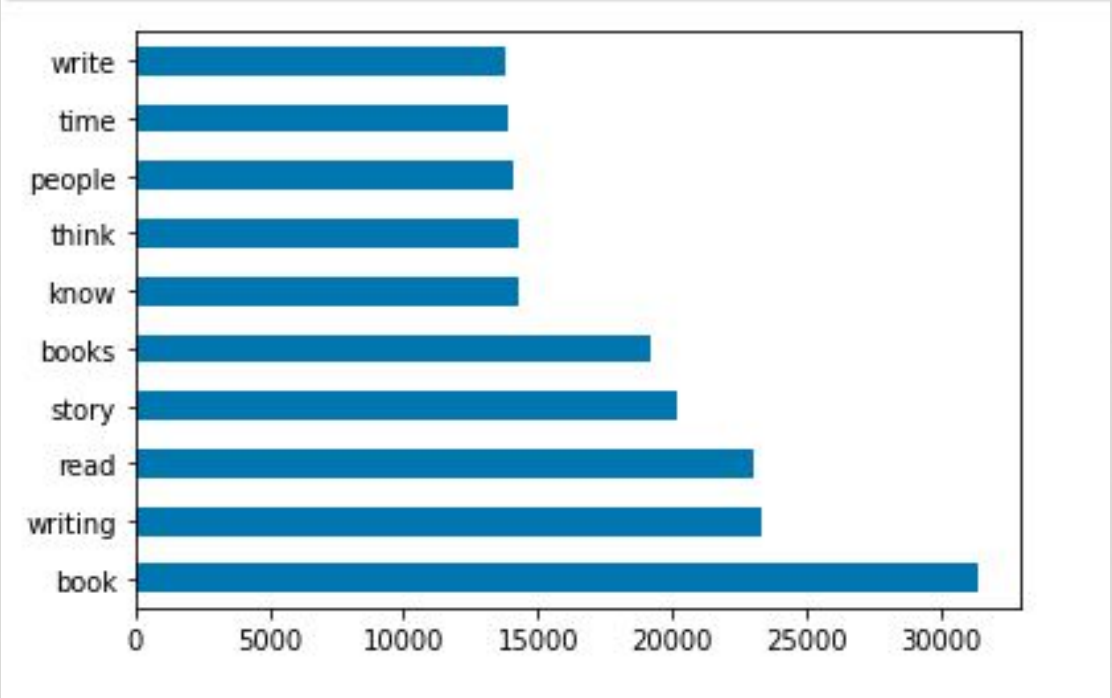
Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash

TOP OCCURRING WORDS IN SAMPLE



NEW TOP OCCURRING WORDS



COEFFICIENTS

	coefficient	words
12237	7.256213	writing
12163	5.013875	write
10174	3.135529	story
1858	2.628173	character
378	2.187365	advice
11688	2.101862	want
12228	2.038970	writers
12221	1.957939	writer
5281	1.948334	idea
5303	1.765130	ideas

	coefficient	words
6246	-0.994568	literature
9529	-1.007205	series
4120	-1.107383	finished
6153	-1.207970	library
8919	-1.373419	remember
9053	-1.505933	review
8595	-3.967145	reading
8466	-4.314328	read
1411	-5.228115	books
1210	-5.609298	book