Project 7

NLP and Classification Models

Andrei Muravev
8/23/2021

Introduction:

In our days popularity of remote learning, jobs and shopping increasing every day. With this, more and more people start to use different online platforms for communications. One of the most popular communication platforms is **reddit**. For my project, I'm going to take two the most popular subreddits: **'books**' and '**parenting**' and try to identify if words in posts from books and parenting different enough that analytical models can predict which subreddit a post came from? To be more specific, I'm going to scrape approximately **10_000 posts** from this two subreddits, mix and identify them. Several classification models - **KNN, Logistic Regression, Decision Tree and Random Forest** will try to predict where posts came from. I'm choosing only classification models for this project because they suppose to group posts together on the basis of common features.

Goals: Collect 11_250 posts from books and 5250 posts from parenting subreddit, analyze, clean, combine and classify if they belong to a certain subclass.

Workflow:

- Use **Reddit API Pushshift** for scraping posts.

- Clean posts using 3 different approaches: **Porter Stemmer**, **RogexpTokenizer** and **WordNetLemmatizer**.

- Count words in subreddits using the **Count Vectorizer**, **Tf-Idf Vectorizer** and **Word Cloud**.

- Experiment with 4 classification models: **Logistic Regression**, **KNN**, **Decision Tree** and **Random Forest** to predict what subreddit a certain post came from.

Cleaning Text Winners:

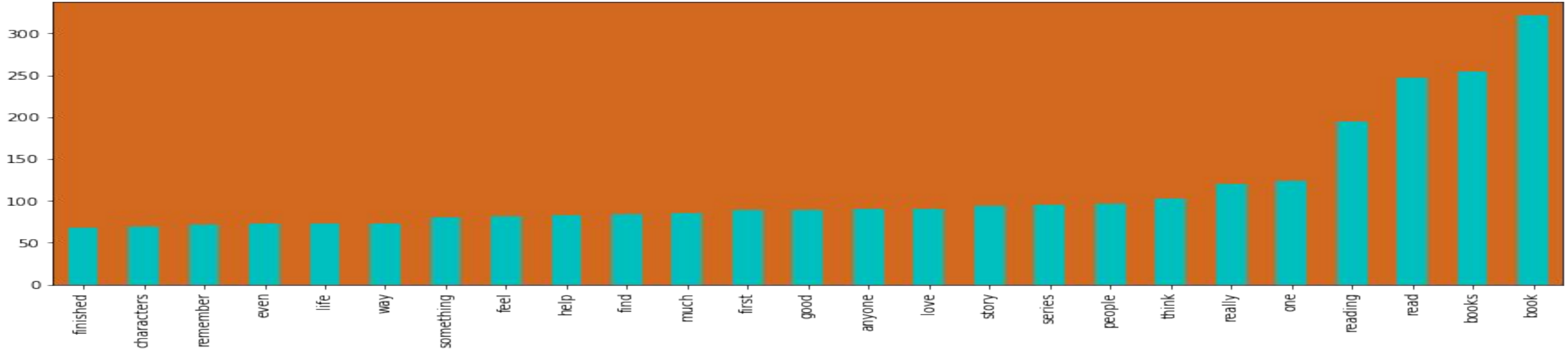
RogexpTokenizer

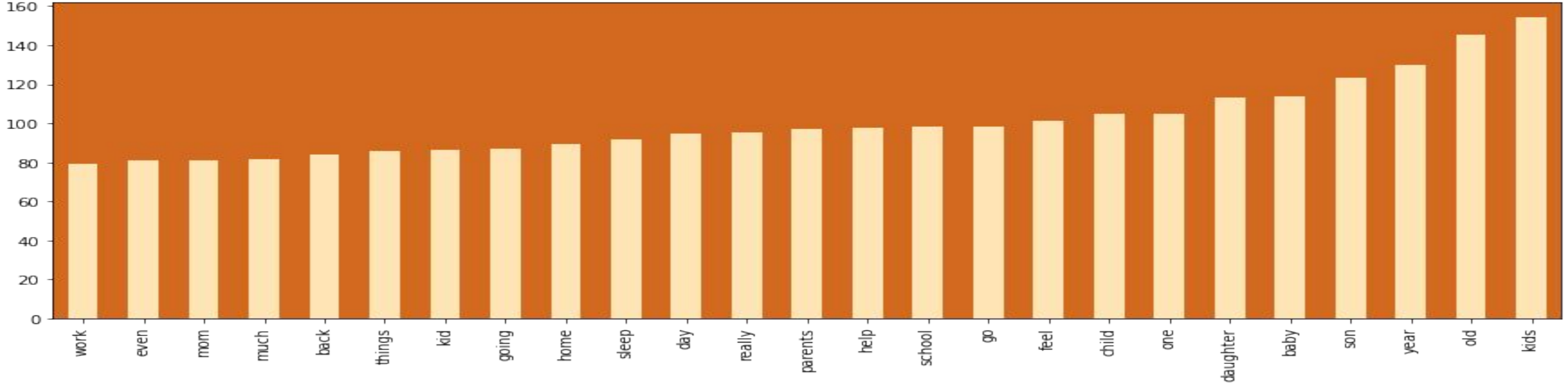
WordNetLemmatizer

Porter Stemmer

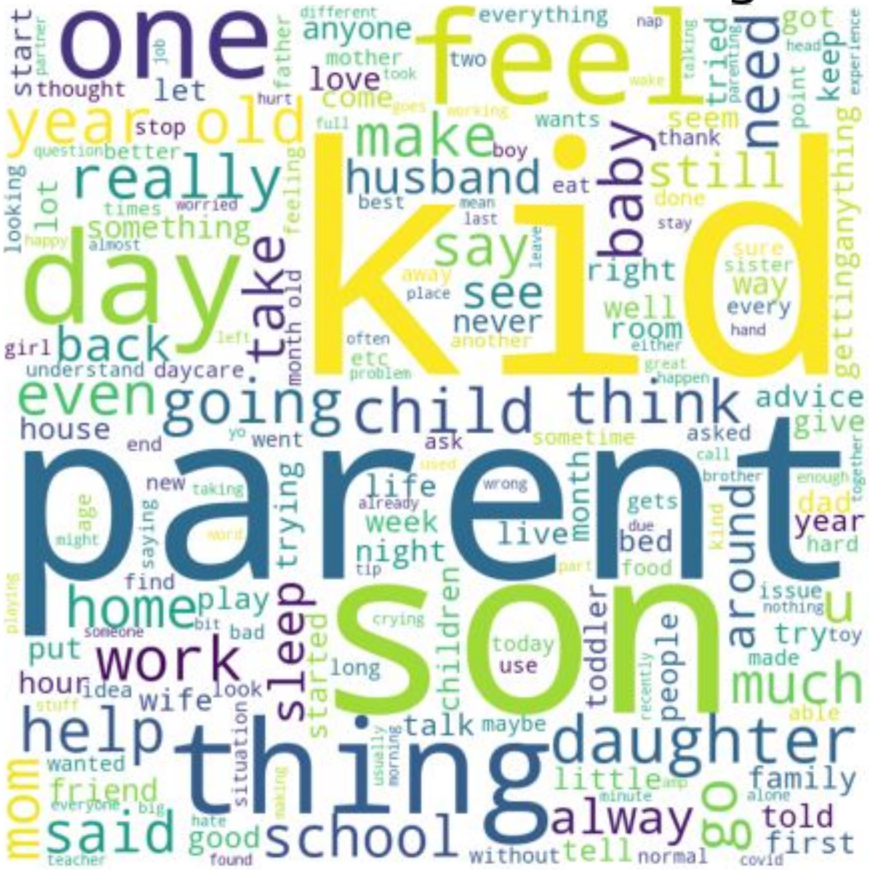
After my exploration, all methods did a great job. But, I like the **Rogexptokenizer** more because it didn't change a root of my words and would delete '/' ';' ':' and other signs attached to words. **Stemmer** operates on a single word without knowledge of the context. **Lemmatizer** refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma

Most common words in both subreddits:

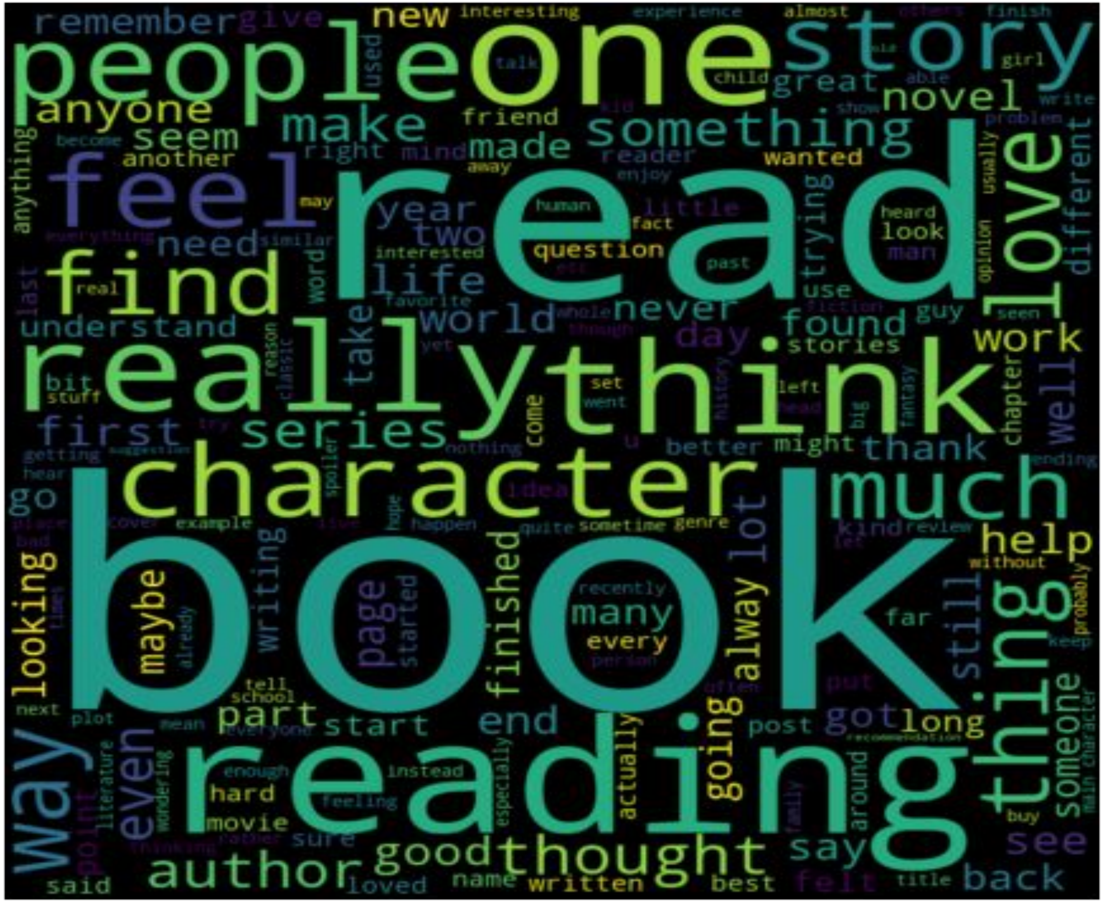




Frequent Words in the Parenting Subreddit

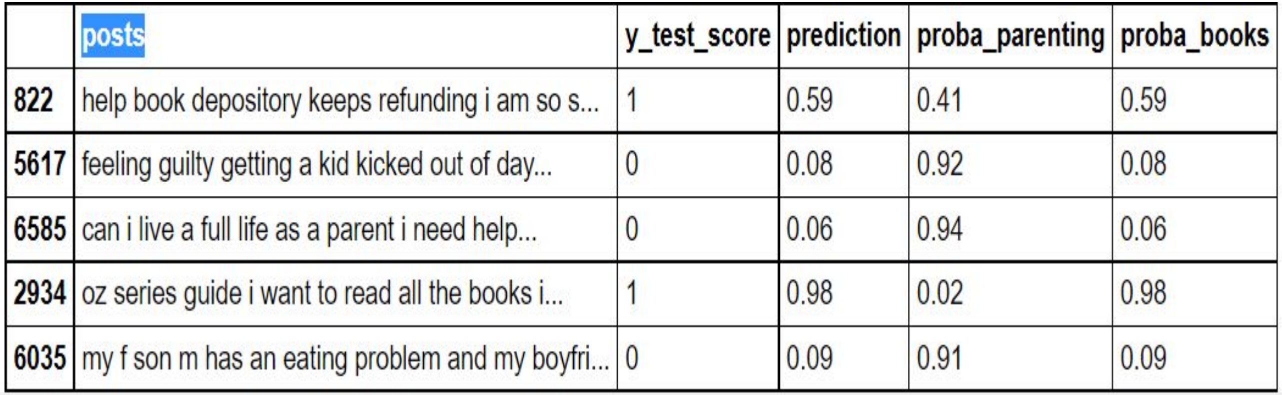


Frequent Words in the Books Subreddit

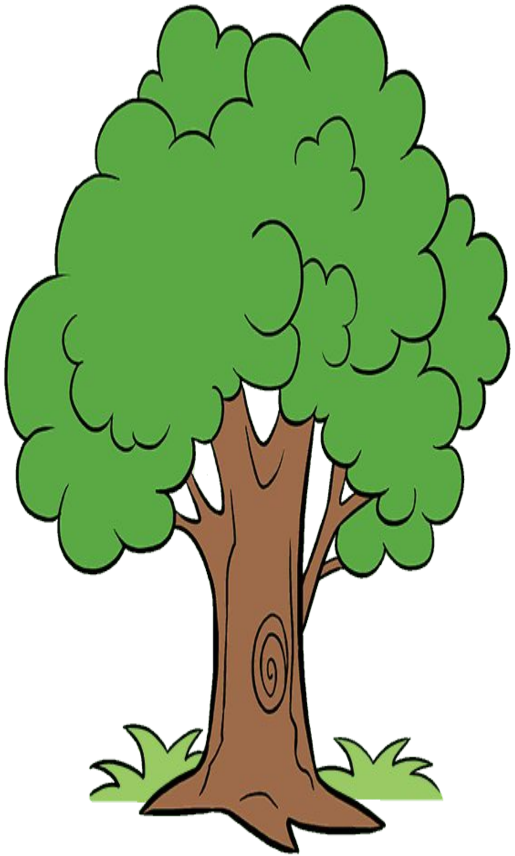


Best models and prediction scores:

- Logistic Regression
- Random Forest
- Decision Tree
- KNN



	posts	y_test_score	prediction	proba_parenting	proba_books
822	help book depository keeps refunding i am so s...	1	0.59	0.41	0.59
5617	feeling guilty getting a kid kicked out of day...	0	0.08	0.92	0.08
6585	can i live a full life as a parent i need help...	0	0.06	0.94	0.06
2934	oz series guide i want to read all the books i...	1	0.98	0.02	0.98
6035	my f son m has an eating problem and my boyfri...	0	0.09	0.91	0.09



Conclusion:

After collecting all results, it is possible to state:

- all models successfully predicted posts that belong to subreddit 'books'. Logistic Regression and Random Forest received 0.98, Decision Tree 0.95 and KNN - 0.53 Moreover if posts belong to 'parenting' subreddit , the models indicated it as well. It means that the best models outperformed the baseline score by more than %50 and this models could be useful in marketing and search bestsellers purposes if further exploration and analysis would be conducted. I've attached predicted results for each individual post to my data folder as a separate 'predictions.csv' file.
- During my project, I've spent a lot of time cleaning my data. Of course, most posts came deleted, removed, included special signs such as (/!,'"&5%$#), emojies or links. All of this could not be helpful for analyzing the texts. Here, I've experimented with three different techniques how to clean my data using Porter Stemmer, RogexpTokenizer and WordNet Lemmatizer. Lemmatizer would cut the endings of words and leave me the root. It is very convenient to use. But, I thought that for this project it didn't work very well because it made some words very short and it was very hard to understand the initial meanings of these words, for example, 'wa' instead of 'want'. Porter Stemmer was good too, but it also changed some words as 'why' to 'whi' and didn't remove all special signs and dotes. RogexpTokenizer was the best. It has a lot of setting that could be changed. I've spent a lot of time experimenting and found one that cleaned my posts as I wanted.
- During Exploratory Data Analysis, additional cleanings of texts were made using two types of stop words 'NLTK english stopwords' plus 'WORDCLOUD stop word's which were helpful in removing unnecessary words such as 'www', 'http', 'is', 'a' and other. After plotting my graphs, it was possible to see which words were most common in subreddit 'books' and subreddit 'parenting' and how many times people used it. It was very significant steps in preparation posts for modeling and creating slides for future stakeholder presentations.