A Tale of two Subreddits: Subreddit Post Classification

...

July 2nd, 2021

Joseph Hicks

Table of Contents

Some Background

- Pushshift.io is a reddit content archive
  - Posts, comments, usernames, etc from years of Reddit
- Problem - Pushshift servers have mixed up data from two subreddits!
  - r/neoliberal
  - r/Conservative
- Mixed up data from the 2nd half of 2020
  - Only have the post title and description
- That's about 69,000 posts from both subreddits!
  - Source: subreddit stats

Source: maxpixel.net

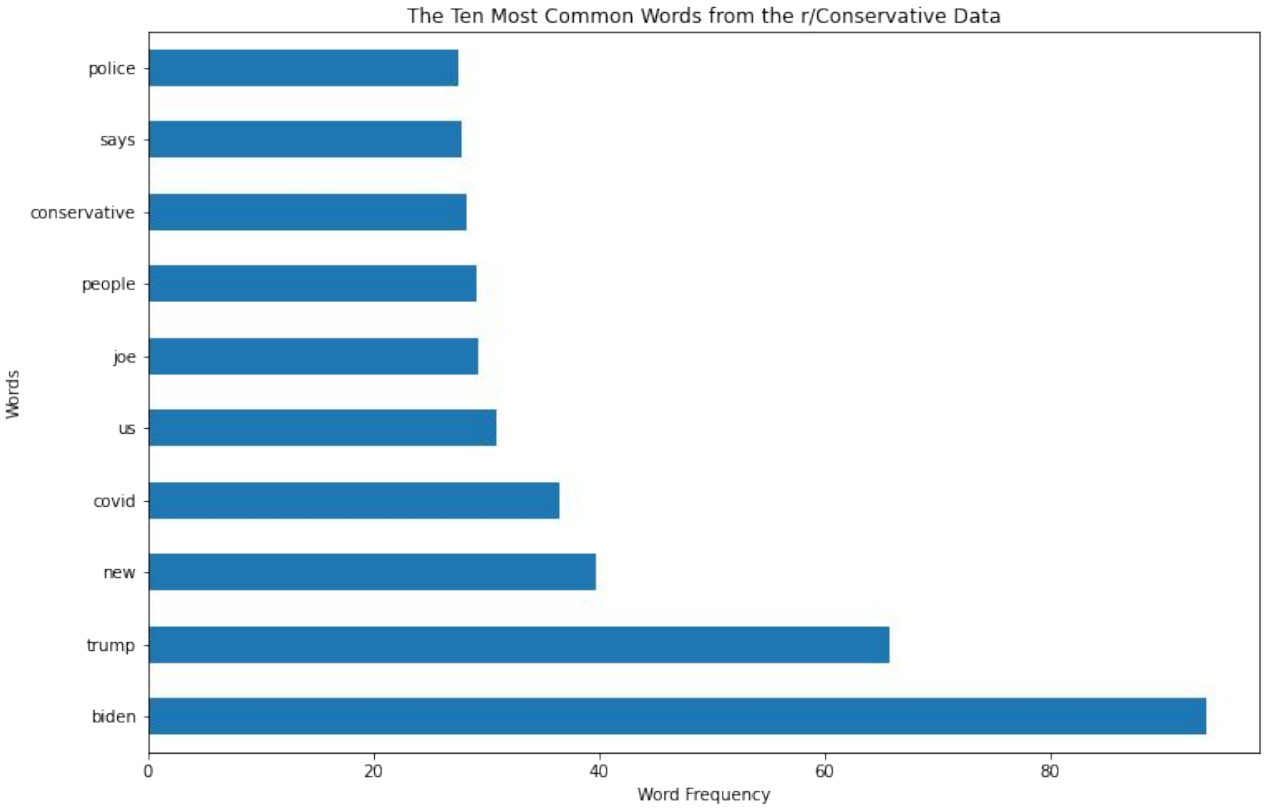
Project objective:
Create a model that can accurately separate posts from r/Conservative and r/neoliberal using their titles and post descriptions

Data Gathering and Cleaning

- Gathered using Pushshift API
  - From subreddit url
- Gathered 100 posts over 3 day intervals from both sources
  - Wanted to balance the number of posts from both subreddits
  - Ended up with a 50:50 split
  - About 9,000 posts in total
- Extracted the title and description
- Removed about 26 nulls
- Combined the description and title features
  - Lots of empty strings and placeholders in description
- Vectorized the text with TF-IDF
  - Allowed for EDA

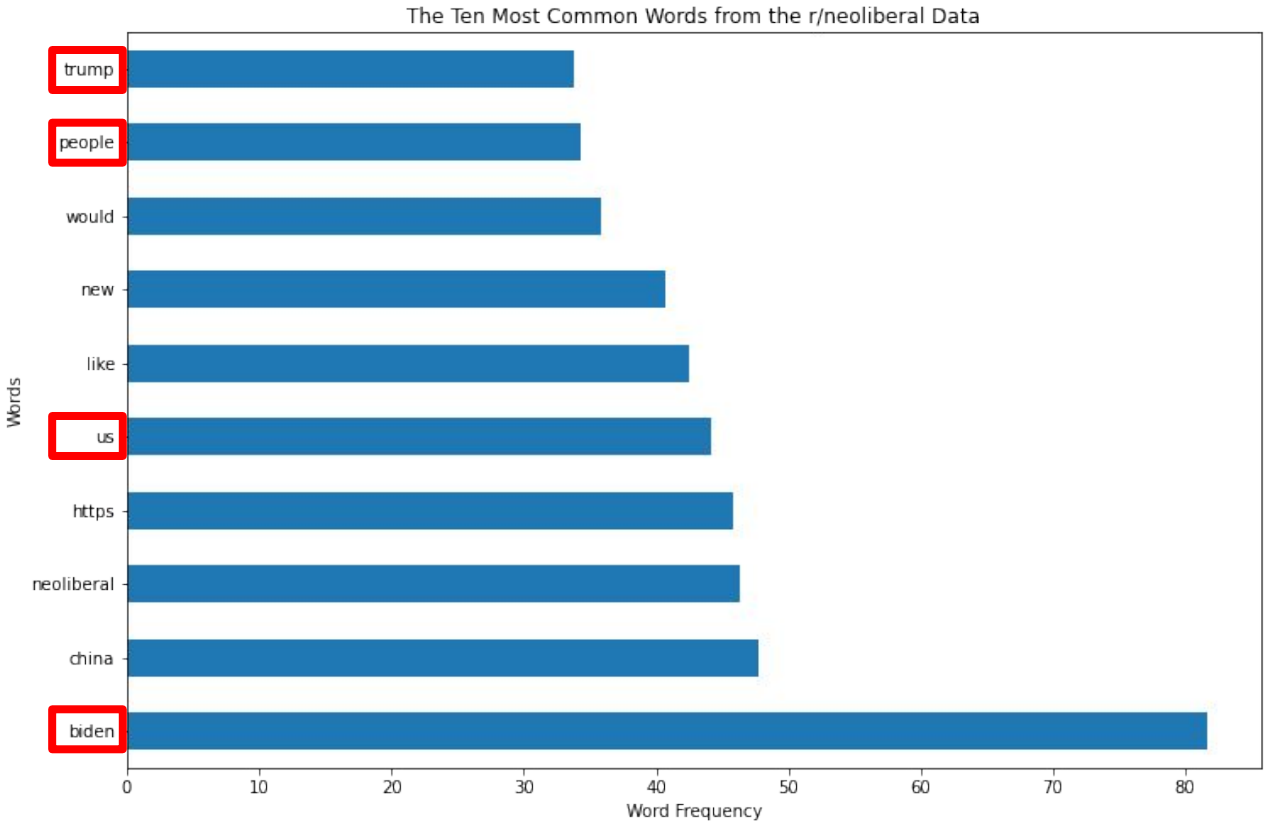
Some r/Conservative Stats

- 263 posts/day on average as of 2021



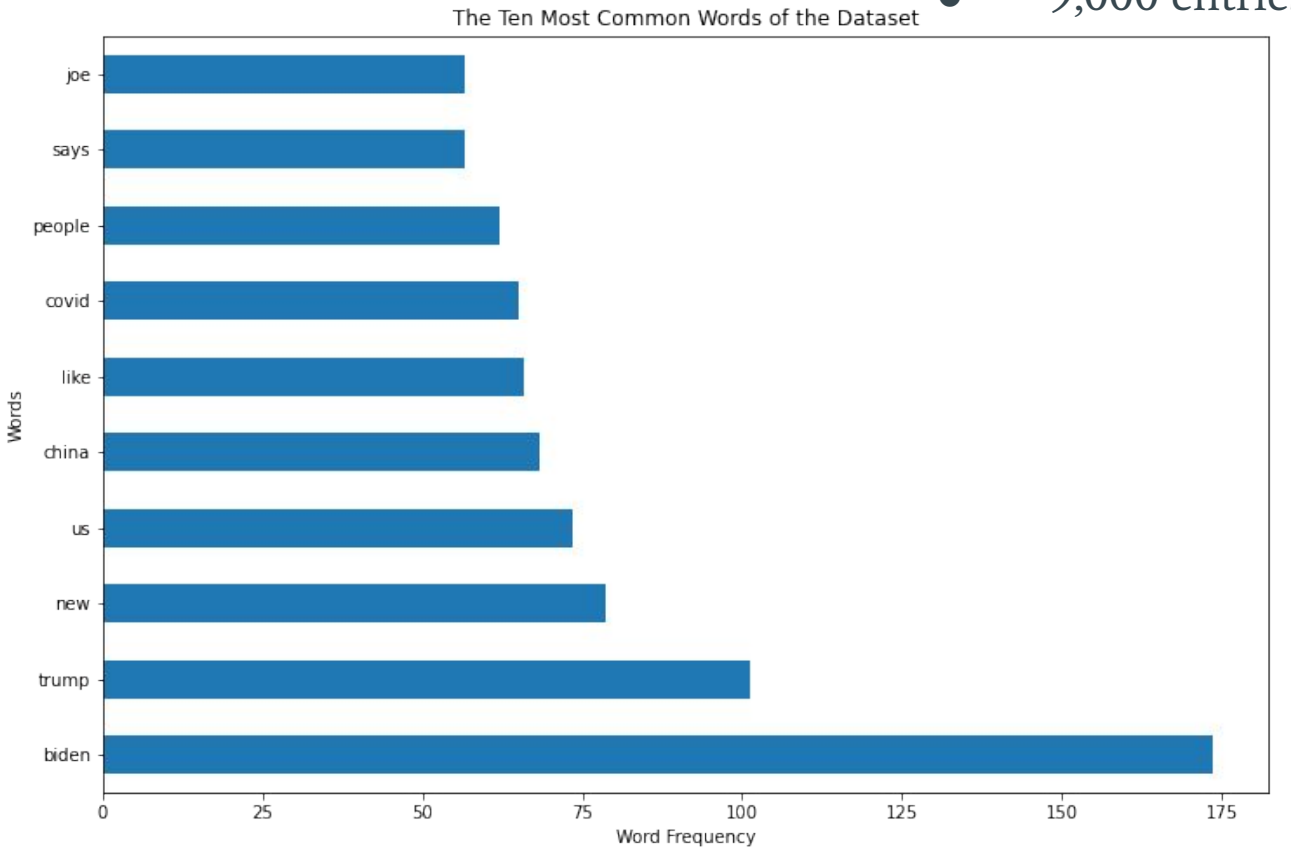
Some r/neoliberal Stats

- 120 posts/day on average as of 2021



Overall Dataset Stats

- Pulled from the first 6 months of 2021
- ~ 9,000 entries



Modeling

50.5%

This is the baseline model - the number to beat

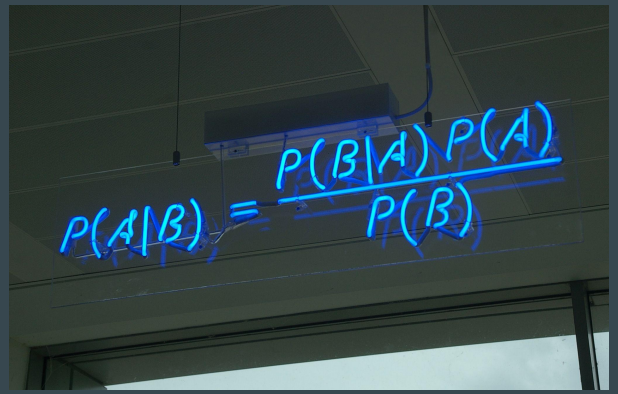
Iteration 1 - The Random Forest Classifier

- An ensemble model that uses many small trees in aggregate to predict
- Pretty slow tuning process
  - Largest gridsearch took about an hour
- Did not perform all that well
- Did several iterations - increased accuracy by 0.05

Training Accuracy	Testing Accuracy	Recall	Precision
0.68	0.67	0.67	0.68

Iteration 2 - The Multinomial Naive Bayes Classifier

- Uses Bayes' Theorem to predict a target class
- Good improvement over the random forest model
- Still plenty of room for improvement



Source: wikimedia

Training Accuracy	Testing Accuracy	Recall	Precision
0.724	0.72	0.72	0.72

The Last Chance - Polynomial Kernel SVM

- Learned of this model in the final hours of the project
- Uses a polynomial kernel to transform data for classifying
- Took *forever* to tune - hours for each gridsearch
- Offered a slight improvement over the other models
- Performed better on unseen data - the test set!

Training Accuracy	Testing Accuracy	Recall	Precision
0.72	0.74	0.67	0.68

Result:

Created a SVM model with 74% accuracy in classifying subreddit posts from similar subreddits

Next Steps

1. Use the model for classifying the mixed up 2020 data in batches
    a. Have a human check these results, though
2. Tune the SVM model more
3. Consider using a more powerful model
    a. Possibly a neural net
4. Increase the complexity of the models
    a. Increase the variance
5. Limitation - not useful outside of the late 2020- early 2021 time frame
    a. Political talking points change quickly
    b. Make sure that data is constantly refreshed if you want to continue to use this model

Thank You!

Any Questions?