Subreddit Classification Using Natural Language Processing

Caroline Clark

Which Subreddit?

Build a model to classify a subreddit post as belonging to either r/artificial or r/datascience

Project Pipeline

Data Acquisition

Data Pre-Processing

Data Visualization

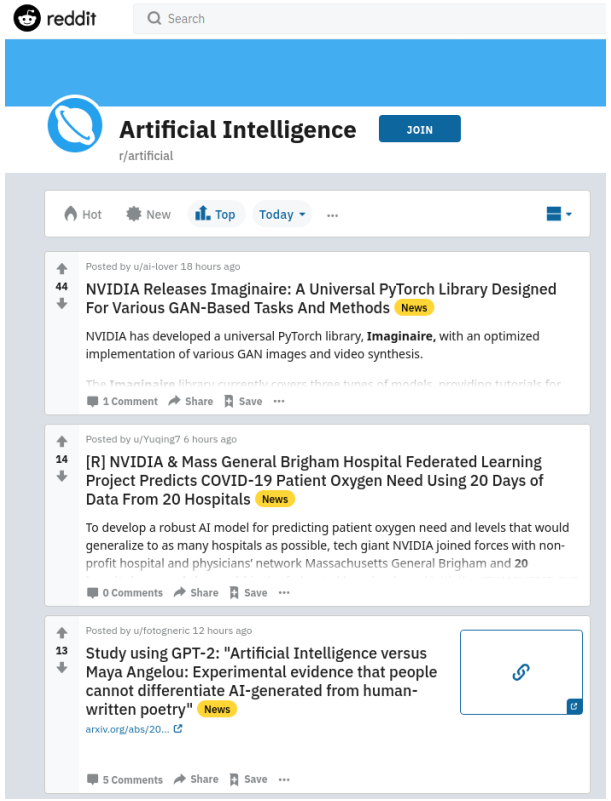
Baseline Naïve Bayes

Optimal Model

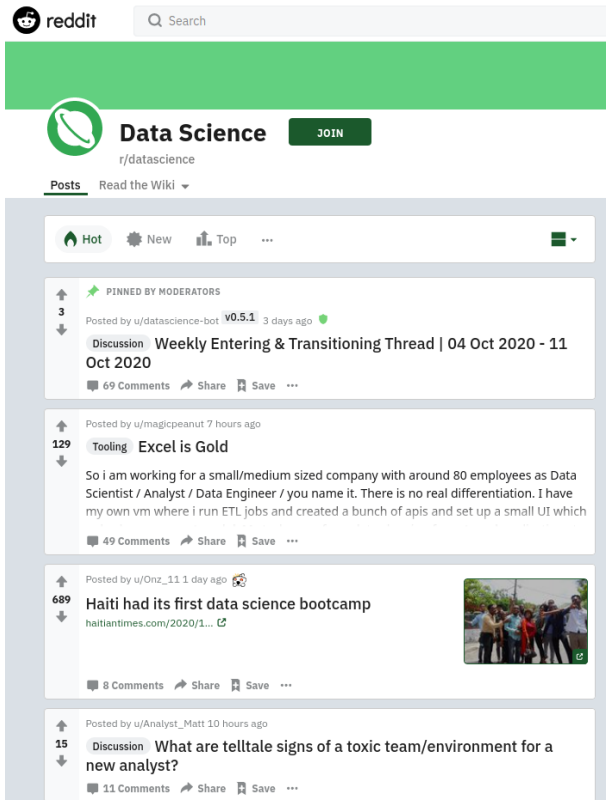
Conclusions

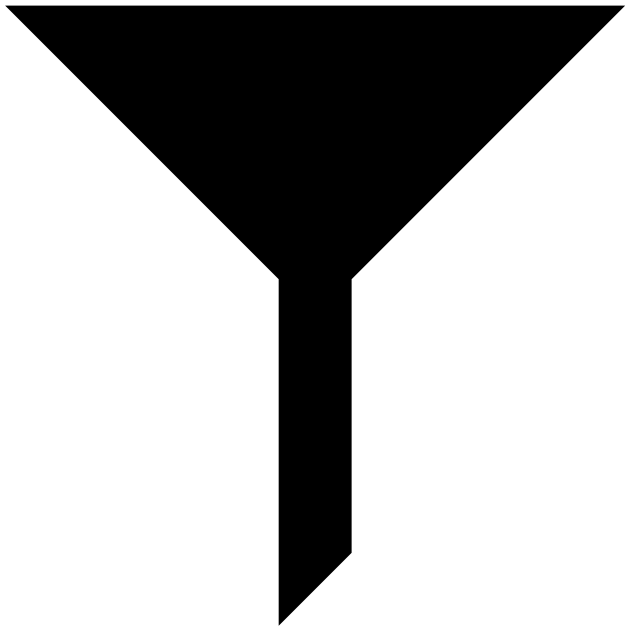
Data Collected with Pushshift API

r/artificial



r/datascience





Data Cleaning Reduced Noise

- Dropped [removed] and [deleted] posts
- Dropped entries with missing post text
- Created 'all_text' feature
- Regex

Regex Used to Clean Text

"That's my goal for the next few years!" says Yann LeCun, the charismatic leader of Facebook AI. Hot off the Press! Here are the links to the November 2018 issue of **Computer Vision News**, the magazine of the algorithm community published by **RSIP Vision**: exclusive interview with **Yann LeCun**, many more articles about computer vision and **free subscription at page 32**.

[HTML5 version (recommended)](https://www.rsipvision.com/ComputerVisionNews-2018November/)

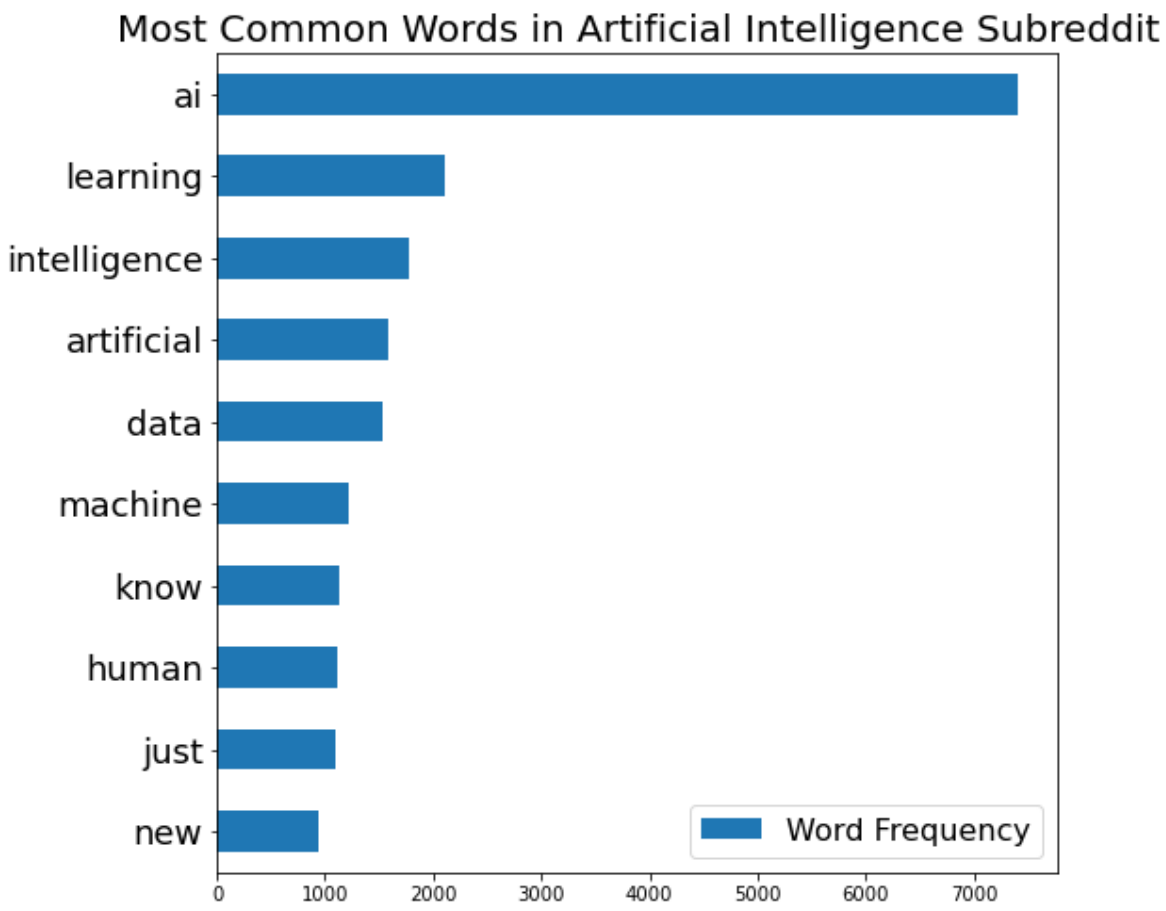
[PDF version](https://www.rsipvision.com/computer-vision-news-2018-november-pdf/)

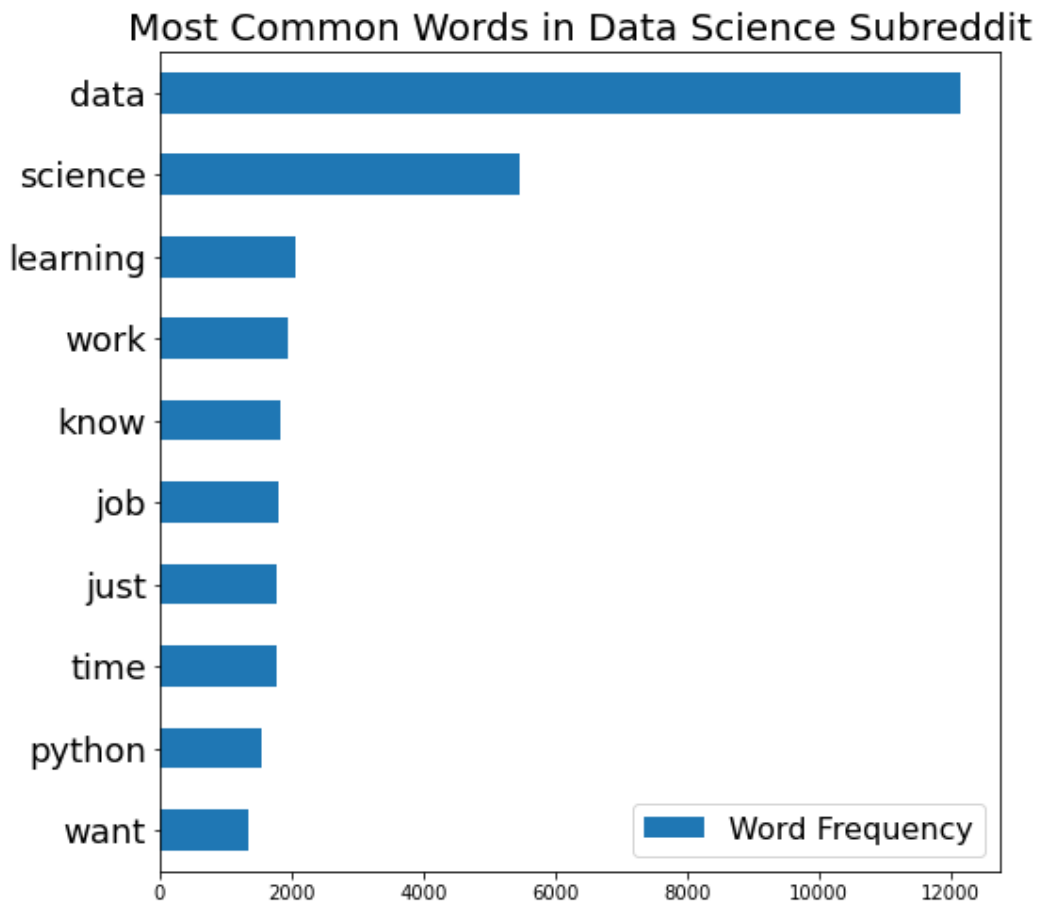
Enjoy!

https://i.redd.it/79khdghqwpw11.jpg

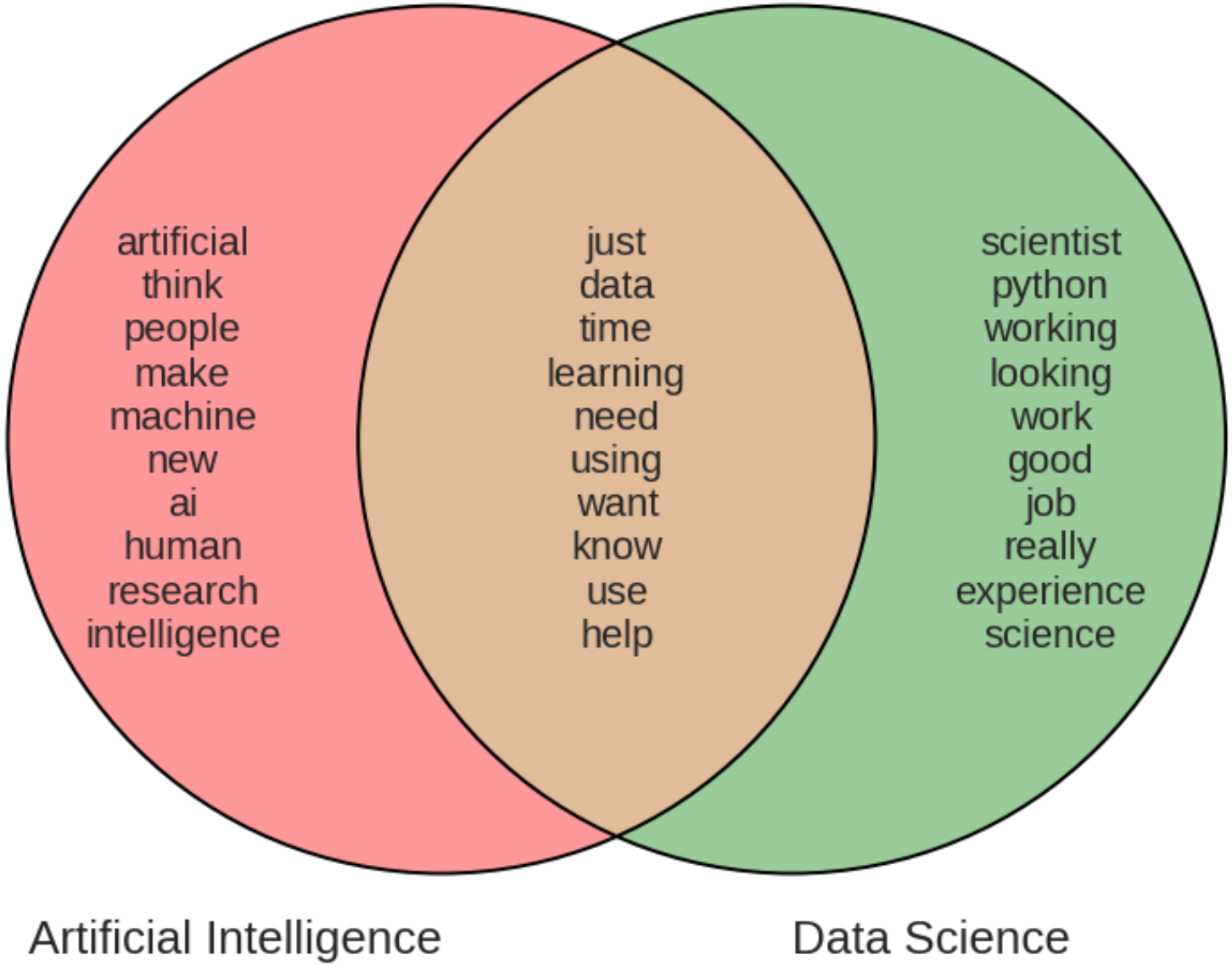
That s my goal for the next few years! says Yann LeCun the charismatic leader of Facebook AI Hot off the Press! Here are the links to the November issue of Computer Vision News the magazine of the algorithm community published by RSIP Vision exclusive interview with Yann LeCun many more articles about computer vision and free subscription at page HTML version recommended PDF version Enjoy! &amp

How Much Overlap in Top Words?





Top Words Overlap ~70%



Data Pre-Processing and Vectorization

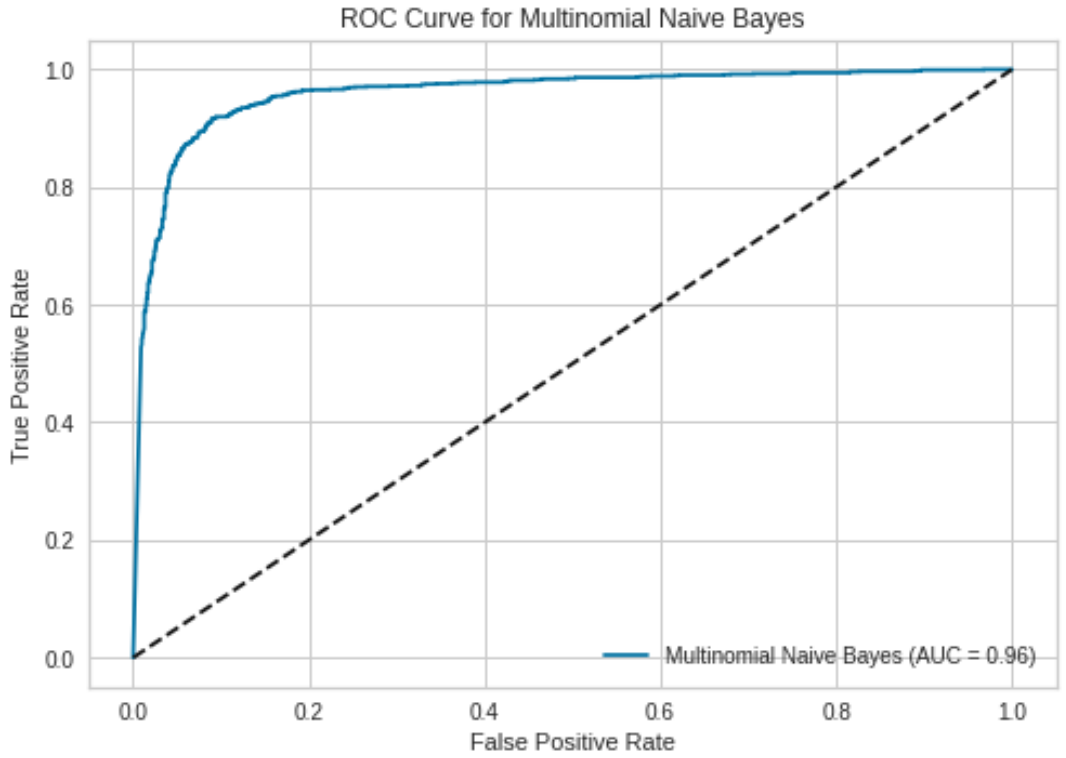
Pre-Processing

- Tokenizing
- Stemming
- Lemmatizing

Vectorization

- CountVectorizer
- TfidfVectorizer

Multinomial Naïve Bayes with CountVectorizer Achieved 90.67% Accuracy



Testing Accuracy	Training Accuracy
90.67%	95.53%

Multinomial Naïve Bayes

`alpha` 2

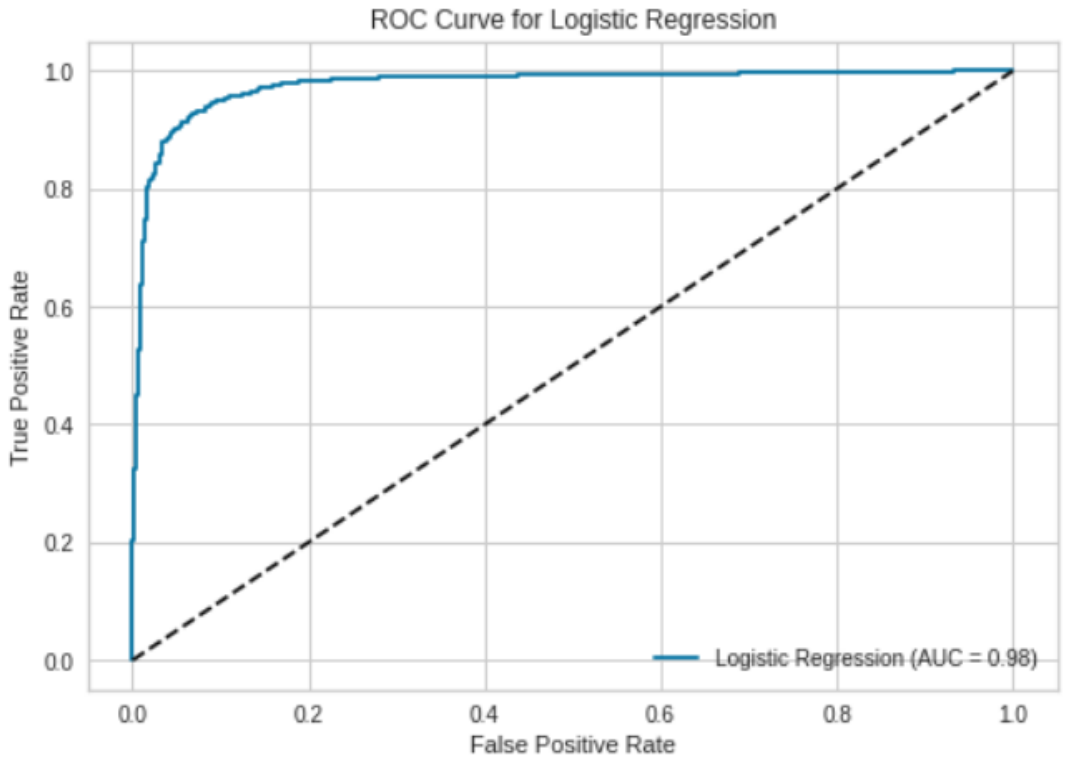
CountVectorizer

`ngram_range` (1, 3)

`max_df` 0.5

`min_df` 2

Logistic Regression with Stemming and TfidfVectorizer Achieved 92.66% Accuracy



Testing Accuracy	Training Accuracy
92.66%	97.85%

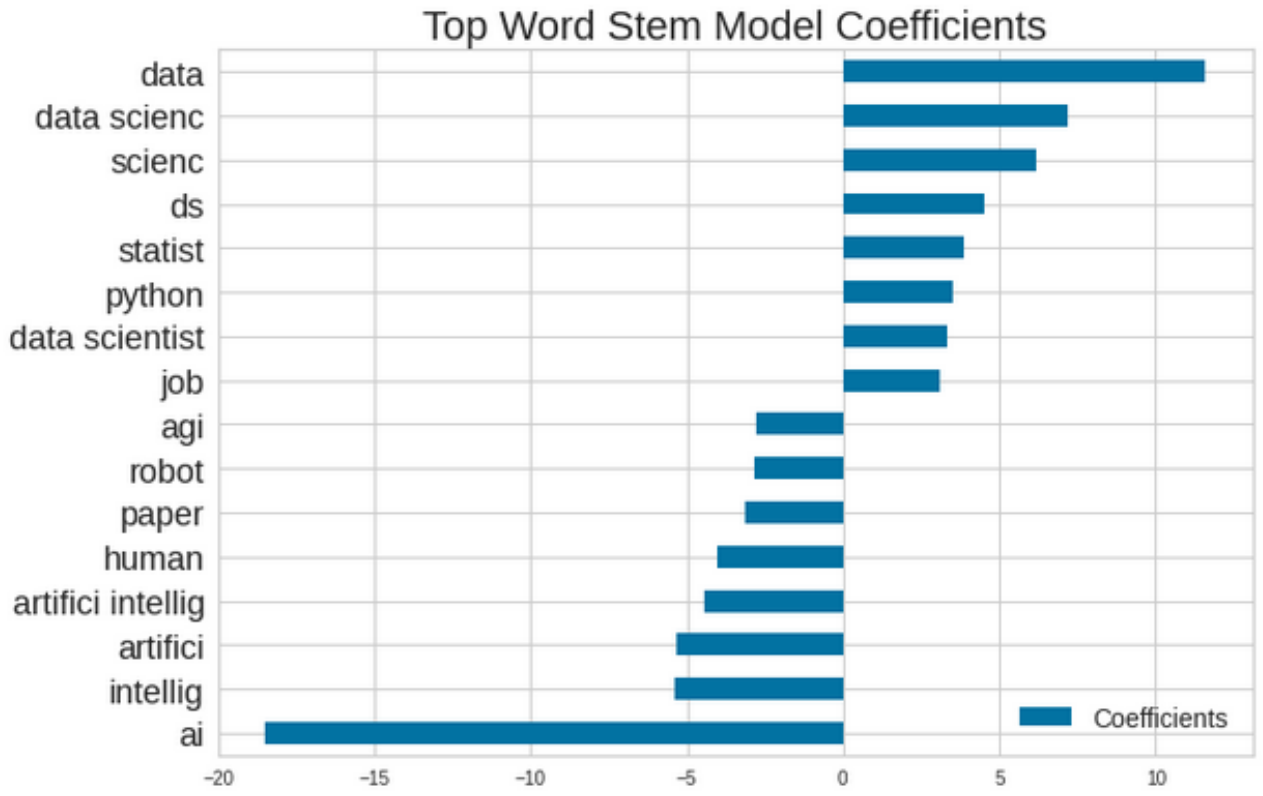
Logistic Regression

`C`	2

TfidfVectorizer

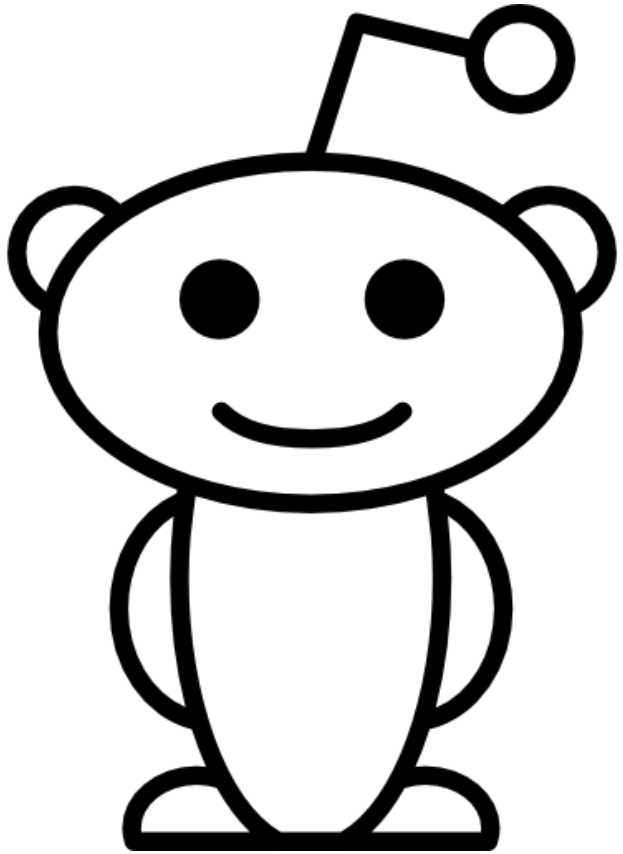
`preprocessor`	stemming
`ngram_range`	(1, 3)
`max_df`	0.5
`min_df`	2

Word Stems that Most Influenced the Classifier



Target Labels

r/datascience	*1*
r/artificial	*0*



Conclusions

- Logistic regression with tuned hyperparameters
  - 92.66% accuracy
  - Coefficients
- More text = better model
  - More relevant?
- Next Steps
  - Bias/variance tradeoff
  - New posts in r/datascience, r/artificial
  - r/deeplearning, r/learnmachinelearning

Questions?

Thank you!