What can data science do?

Using NLP and modeling techniques to classify posts from different subreddit’s

Magnus Bigelow

What Problem are we Looking to Solve?

Problem 1	Problem 2
Given the text contained within the title and original post from r/woodworking and r/mtb can we predict which subreddit the post came from with >85% accuracy?	Further, using the same model and hyperparameters can we achieve >80% accuracy using the two similar subreddits r/mtb and r/bicycling?

The Data

We collected the 10,000 most recent posts from 3 subreddit's using the pushift API and concatenated the text contained in the title and original post to create a 'text' variable

r/woodworking: all things made from trees

Mean text length: 193

Median text length: 88

r/mtb: reddit for mtn bikers

Mean text length: 278

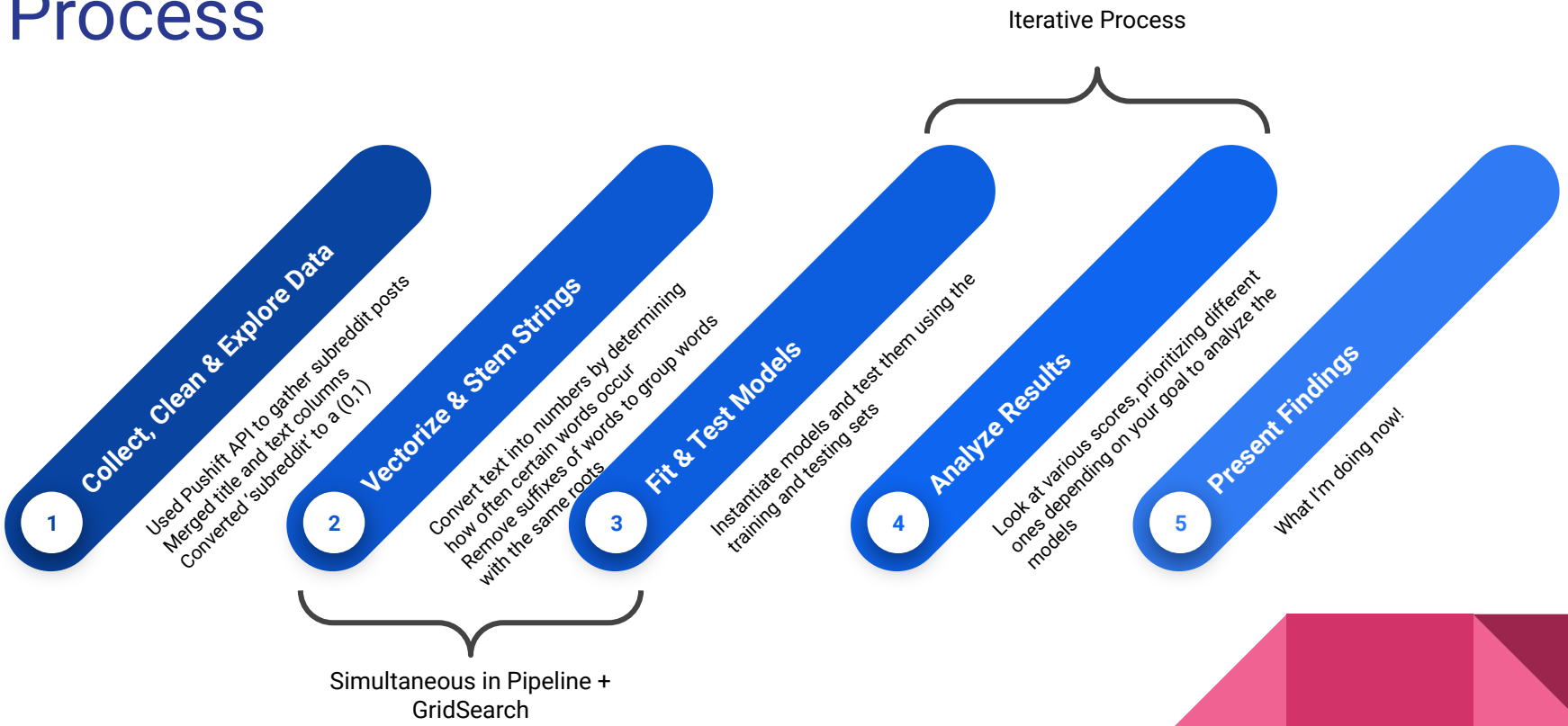
Median text length: 113

r/bicycling: two wheels, powered by a person

Mean text length: 250

Median text length: 84

Process


Iterative Process
1
Collect, Clean & Explore Data
Used Pushift API to gather subreddit posts
Merged title and text columns
Converted 'subreddit' to a (0,1)
2
Vectorize & Stem Strings
Convert text into numbers by determining how often certain words occur
Remove suffixes of words to group words with the same roots
3
Fit & Test Models
Instantiate models and test them using the training and testing sets
4
Analyze Results
Look at various scores, prioritizing different ones depending on your goal to analyze the models
5
Present Findings
What I'm doing now!
Simultaneous in Pipeline + GridSearch

Problem 1:

Given the text contained within the title and original post from r/woodworking and r/mtb can we predict which subreddit the post came from with >85% accuracy?

Stemming & Vectorization Approaches

No Stemming

Look at the the tokens (i.e. words) as they originally appear in the documents

Porter Stemmer

Remove short and long suffixes from numerous tokens to get their roots

Word Net Lemmatizer

Remove short suffixes from numerous tokens to get their dictionary form

Count Vectorizer

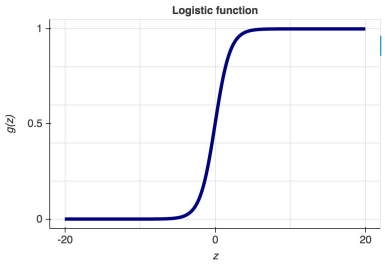
Simple count of the number of times a token appears in each document

Tfidf Vectorizer

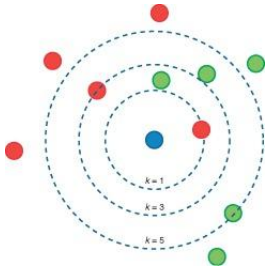
Uses equation that compares the frequency a token appears in a document to the frequency it appears in the corpus

Modeling Approaches

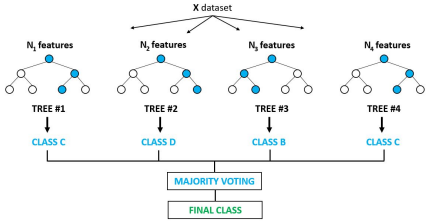
Logistic Regression



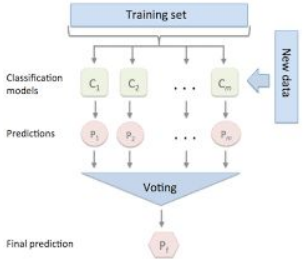
K-Nearest Neighbors



Random Forest



Voting

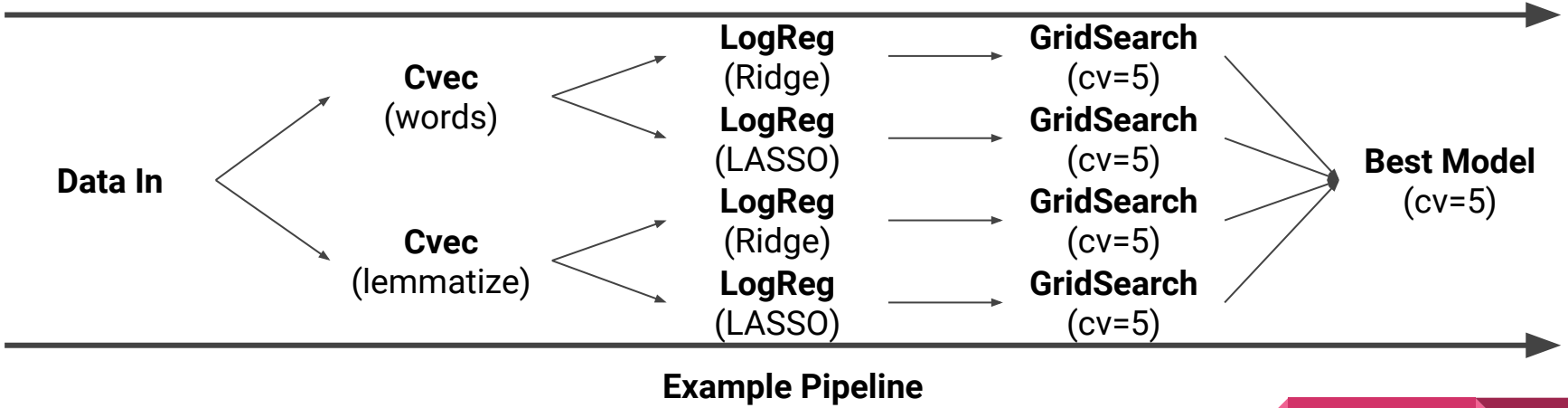


Naive Bayes

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

How do we know we have the best model?

We will likely never have the best model but we can test hundreds of models fairly simply, if not quickly using a pipeline and grid search.



Example Pipeline

Results

Vectorizer	Model	Accuracy	Precision
CountVectorizer	Logistic Regression	0.921	0.922
CountVectorizer	K-Nearest Neighbors	0.820	0.804
CountVectorizer	Naive Bayes	0.914	0.905
CountVectorizer	Random Forest	0.917	0.906
TfidfVectorizer	Logistic Regression	0.919	0.920
TfidfVectorizer	K-Nearest Neighbors	0.737	0.715
TfidfVectorizer	Naive Bayes	0.737	0.656
TfidfVectorizer	Random Forest	0.918	0.909
CountVectorizer	**Voting**	**0.923**	**0.923**

Accuracy:
correct / total

Precision:
correct 1's / predicted 1's

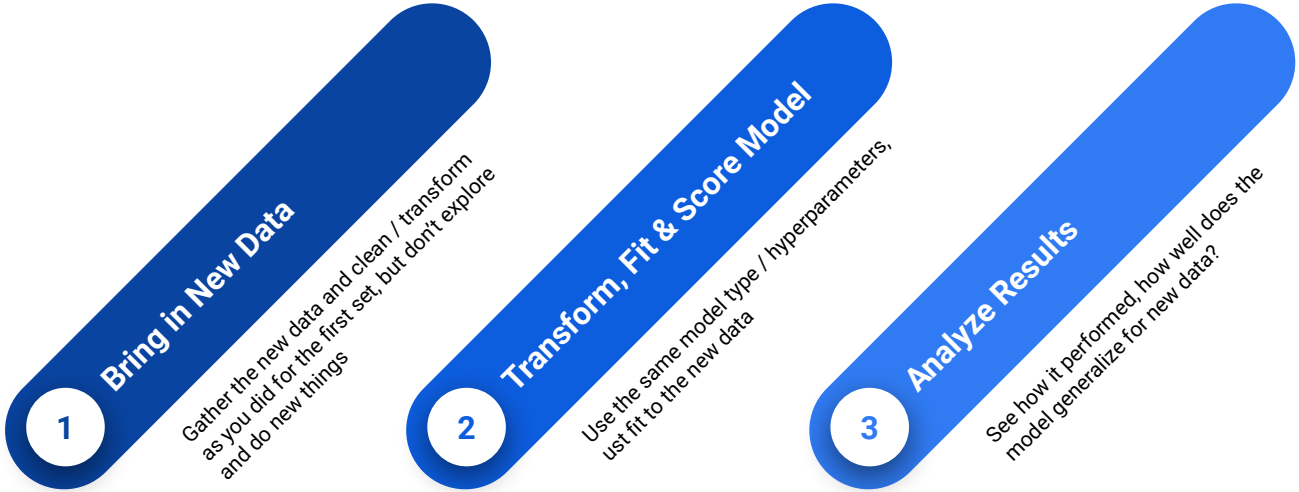
If Accuracy >> Precision: model is predicting too many 1's

If Accuracy << Precision: model is predicting too many 0's

Problem 2:

Further, using the same model and hyperparameters can we achieve >80% accuracy using the two similar subreddits r/mtb and r/bicycling?

Modeling Process & Results



Subreddit 0	Subreddit 1	Accuracy	Precision
r/mtb	r/woodworking	0.923	0.923
r/mtb	r/bicycling	0.763	0.746

How did we do?

Problem 1	Problem 2
Given the text contained within the title and original post from r/woodworking and r/mtb can we predict which subreddit the post came from with >85% accuracy?	*Further, using the same model and hyperparameters can we achieve >80% accuracy using the two similar subreddits r/mtb and r/bicycling?*
Success: *VotingClassifier achieved 0.92 test accuracy*	**Failure: *VotingClassifier achieved 0.76 test accuracy***

Any questions?