Using NLP to classify Reddit posts: math vs physics

Mike Bell

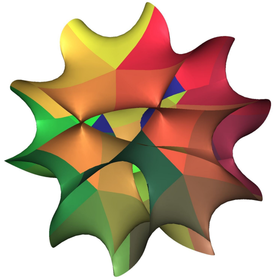
GA DSI EC 13 - October 23, 2020

Problem Statement

Mathematics and physics are interconnected disciplines, with a long, rich history of codevelopment and evolution, and it is often stated that math is the language of physics. However, in this project we want to investigate the similarities and differences between the everyday language used around each subject.

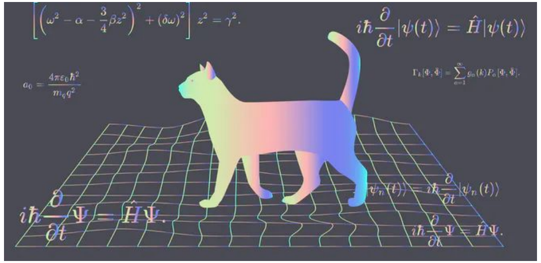
More specifically: Based on its text alone, how accurately can we predict if a given post came from a mathematics or physics subreddit?

We use natural language processing (NLP) and machine learning techniques to build classification models to try to answer this question.



r/math

- 1.4 Million subscribers
- “As per the sidebar, the subreddit is intended for mathematical topics. Posts requesting help with basic math and homework should go to more specialized subreddits (respectively, /r/learnmath and /r/cheatatmathhomework). The posts in /r/math tend to be mostly about topics at an undergraduate level (this is descriptive, not prescriptive: you may post about topics more or less advanced)....”


$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle$
$i\hbar\frac{\partial}{\partial t}\Psi = \hat{H}\Psi.$

r/physics

- 1.5 Million subscribers
- “The aim of /r/Physics is to build a subreddit frequented by physicists, scientists, and those with a passion for physics. Papers from physics journals (free or otherwise) are encouraged. Posts should be pertinent, meme-free, and generate a discussion about physics. Please report trolls and intentionally misleading comments.”

Data Collection

- We use the Pushshift API (https://github.com/pushshift/api) to scrape posts from r/math and r/physics.
- Limited to 100 posts per request.
- Each post has a feature created_utc which gives a timestamp it was created, we save the oldest timestamp in the current request and start our next request looking only for posts created before this time.
- Each post consists of features including: title, selftext (post body text), id, author, time of creation, flair, url, etc. - We only use title and selftext for training our models.
- After cleaning: 4423 total posts : 2217 from r/math (50.1%) and 2206 from r/physics (49.9%)

- (Actually able to get over 150,000 posts, but did not have the computational power to incorporate more data into modeling process)

Data Cleaning and Preprocessing

- Most posts don’t have body text, replace resulting NaNs with empty strings.
- Remove empty posts and posts that have been flagged as being removed (such posts are still retrieved by Pushshift).
- r/math is highly moderated (posts asking for math homework help are removed, for example)
  - Of 3900 most recent posts, at least 1600 had been removed.
  - I chose to not include such posts in my analysis.
- Basic text cleaning: Convert to lowercase, remove common stop-words (‘a’, ‘the’, ‘is’, etc.), remove URLs, html codes such as &amp;, numbers and symbols.
- Combined all text (title and body) into a single feature.
- Encode subreddit as 0 (r/math) and 1 (r/physics)
- Lemmatize all text.

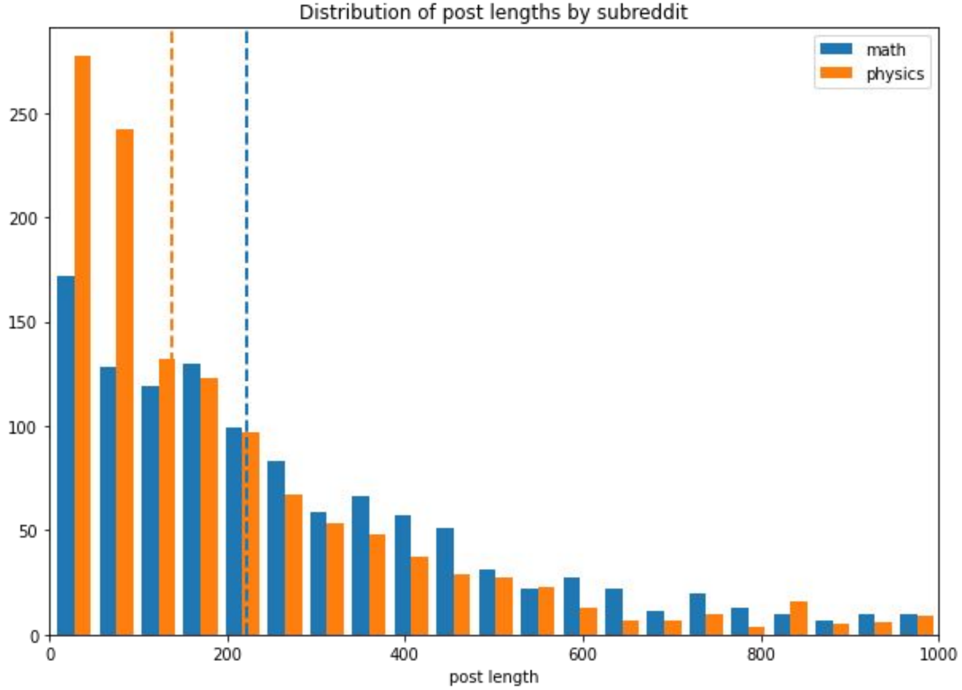
Vectorization

- Goal: Convert unstructured data (text) into structured data (numerical features), we use two main methods:
- CountVectorizer: Feature for each unique word in training set, value is the number of times that word appears in the post.
- TfidfVectorizer (Term Frequency - Inverse Document Frequency): Scales the number of times a term appears in a posts, by an inverse measure of how frequently that term appears across all posts. Gives low weights to common words.
- Both methods can also create featuress for *n-grams* or strings of n contiguous words.

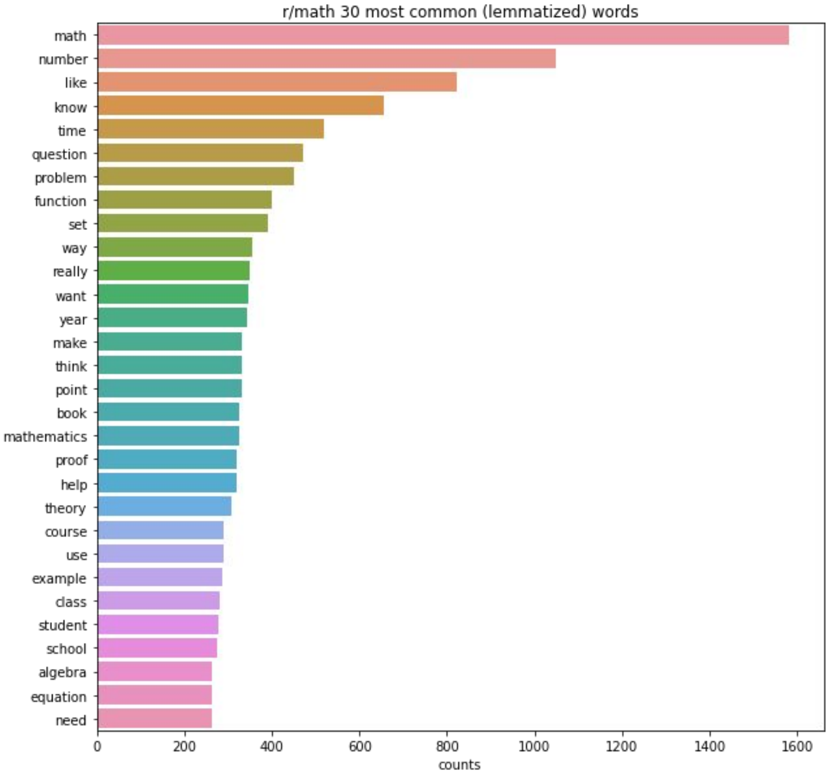
Exploratory Data Analysis

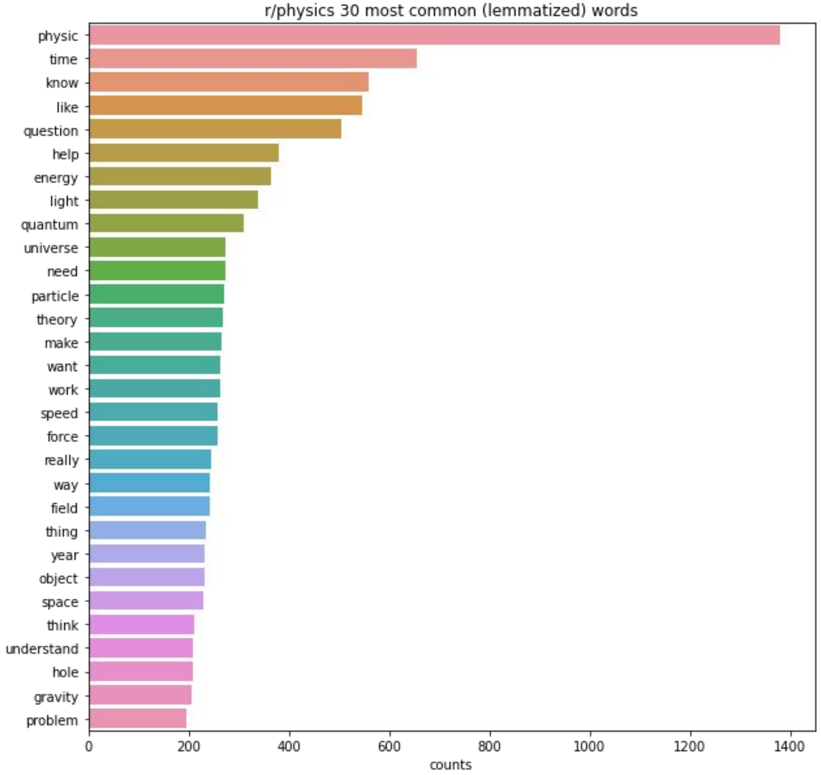
Post Length by Subreddit

subreddit	Median Length	Mean Length	Max Length
r/math	222	330	4992
r/physics	141	257	14226



Most Common Words





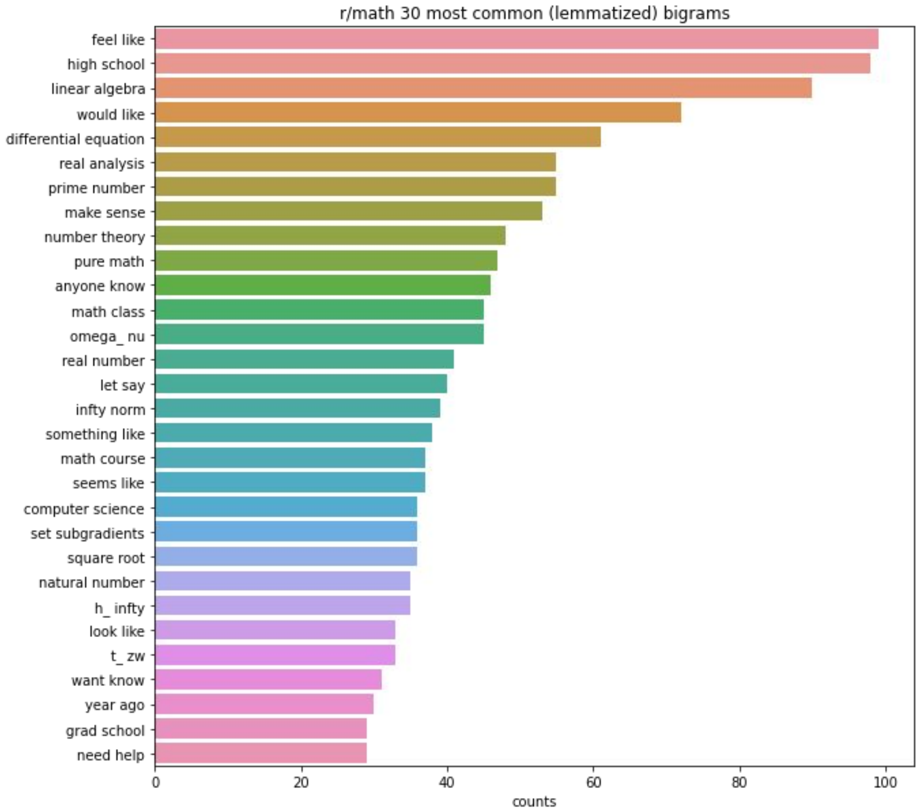
- **Top 30 words r/math only :** math, number, function, point, set, class, proof, book, student, mathematics, course, equation, use, example

- **Top 30 words r/physics only:** physic, energy, light, quantum, need, particle, field, universe, force, space, object, understand, speed, velocity

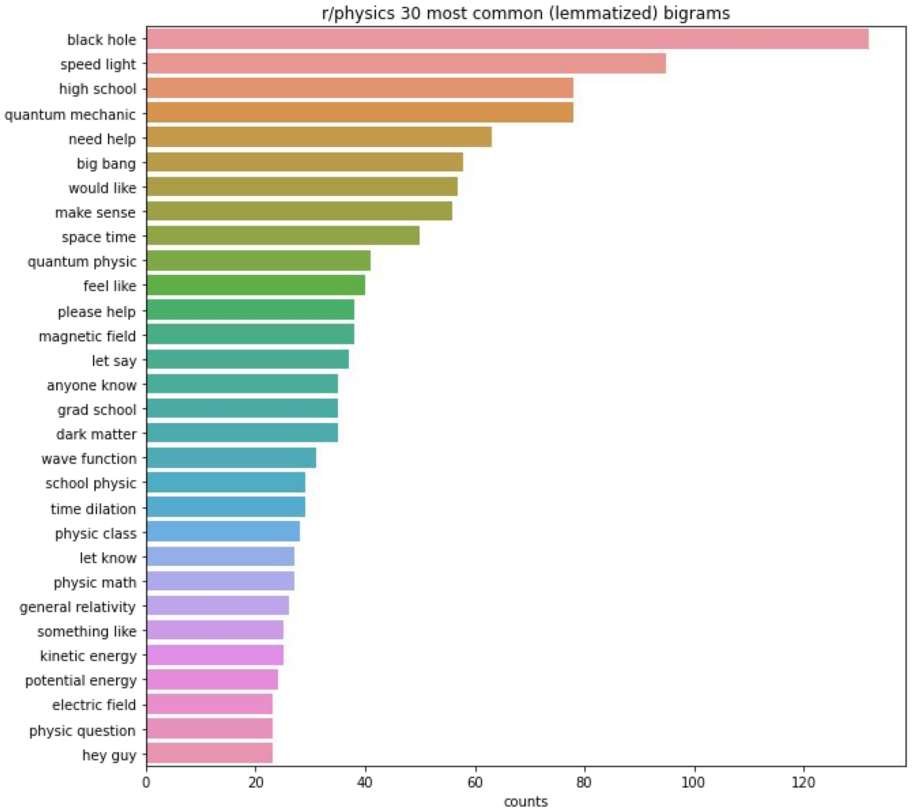
- **Top words in common:** theory, make, thing, want, think, really, question, year, problem, know, time, work, school, way, like, help

Bigrams

Big rams?



r/math 30 most common (lemmatized) bigrams



r/physics 30 most common (lemmatized) bigrams

Modeling

Classification Models

Initially consider 10 classification models: k Nearest Neighbors, Naive Bayes, Logistic Regression, SVM, Decision Trees, Bagged Trees, Random Forests, Adaboost, Gradient Boosted Trees, and Extremely Randomized Trees.

Train-Test Split: Our dataset of 4423 posts is split into 70/30 train/test sets.

The train set is used in 5-fold cross-validation for tuning hyperparameters and model selection, and the test for final model evaluation.

Baselines

- Balanced training data: Nearly exactly 50/50 split between class 0 (math) and 1 (physics).
- To start, we train 40 basic models: The 10 previously mentioned models using default scikit-learn parameters using four different vectorized feature sets (CountVectorizer and TfidfVectorizer with bigrams on and off).
- Our main evaluation metric is accuracy (proportion of posts correctly classified), though we also compute sensitivity, specificity, and precision.
- Mean 5-fold cross validation score on the training set, and score on the test set are used to select best models.

Best basic models

- The best default models were Naive Bayes, SVM, and Logistic Regression using both CountVectorizer and TF-IDF and with and without bigrams.
  - All around 90% 5-fold CV accuracy and 90% test accuracy.

- Tree-based models were next best, but were computationally expensive to train.
  - ExtraTree and Random Forest outperformed Adaboost and Gradient Boosted trees for this task.

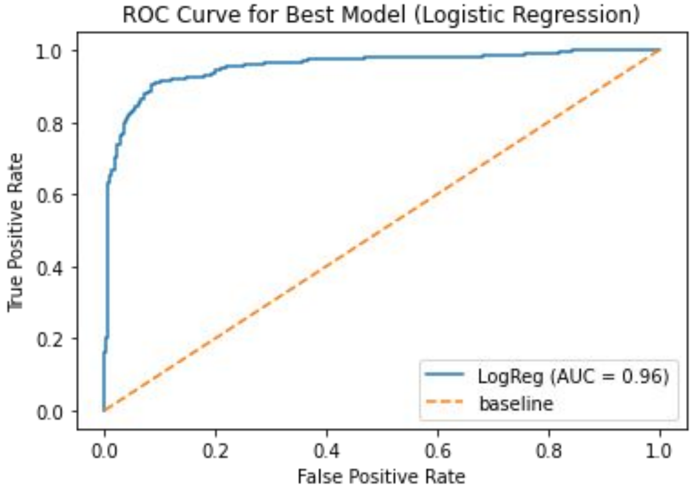
- Bagging, simple decision trees and knn did the worst, but all had accuracy over 71%, a good bit over baseline of 50%.

Hyperparameter Tuning

- Final Best Candidates: Naive Bayes, SVM, and Logistic Regression (all with CountVectorizer and TF-IDF)
- We also include Random Forest Classifier, for a total of 8 model pipelines.
- Performed 5-fold Cross-Validated GridSearch to find optimal parameters for each vectorizer, model combination.

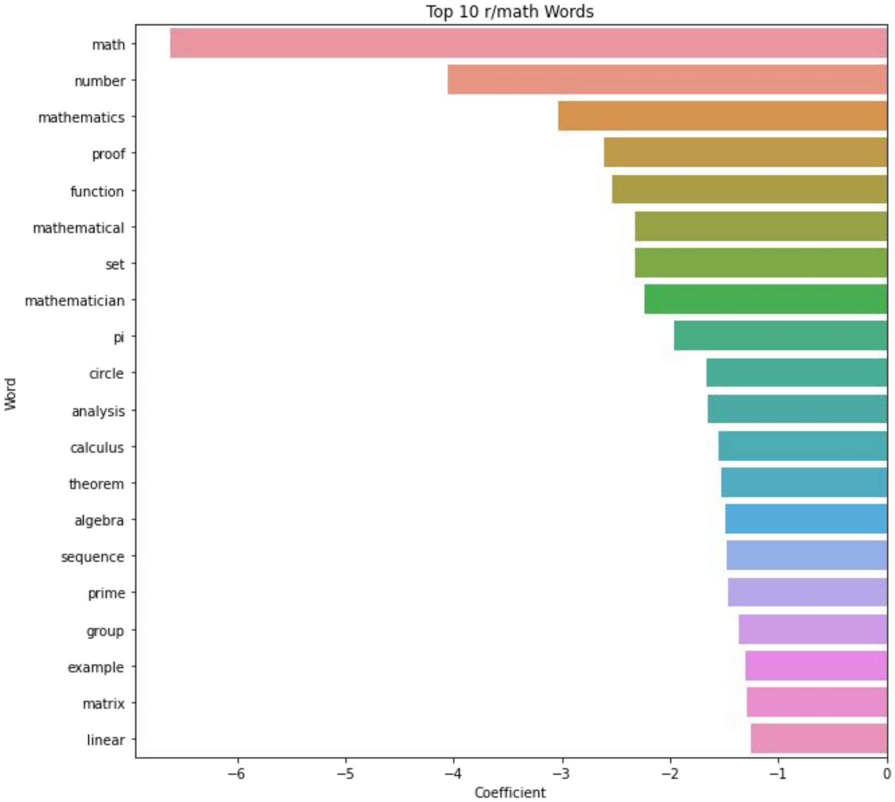
Best Models

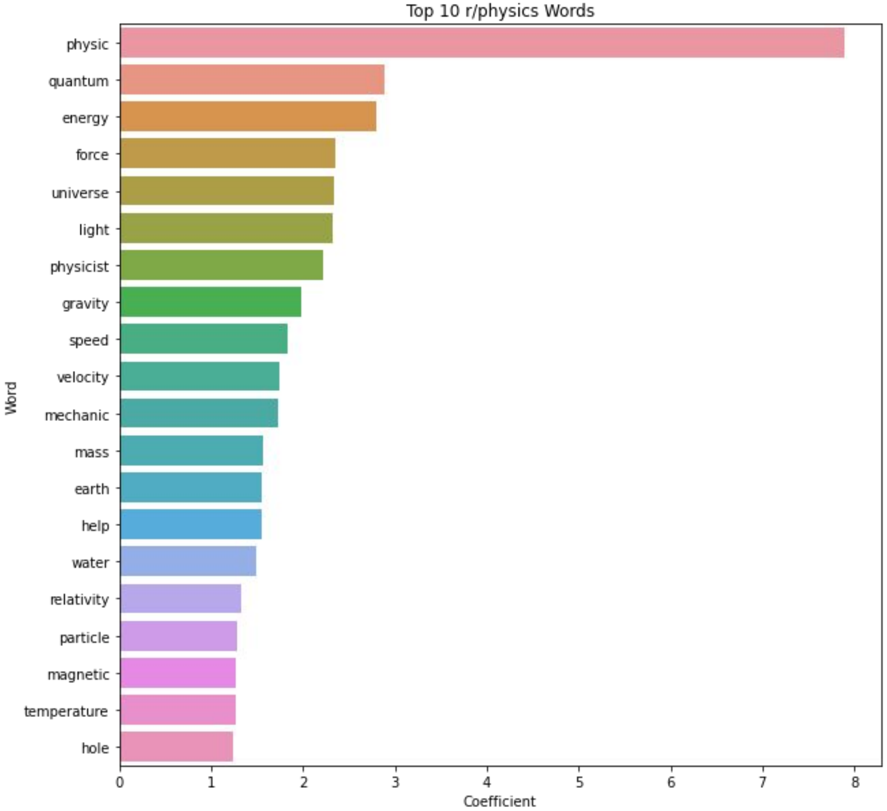
Model	Train Score	CV Score	Test Score	Sensitivity (TPR)	Specificity (TNR)
LogReg (TF-IDF)	0.97323	0.89596	0.90508	0.90271	0.90745
SVM (TF-IDF)	0.99962	0.89973	0.90282	0.88688	0.91874
NaiveBayes (CountVect)	0.95666	0.89898	0.89944	0.873303	0.92551
RandomForest (CountVect)	0.99962	0.85111	0.86214	0.80995	0.91422



- All models seem to be overfitting, but seem to generalize fairly well to new data (much above baseline of ~50% accuracy)
- All models have higher specificity (True Negative Rate) than sensitivity (True Positive Rate) - all predict the negative (r/math) class with over 90% accuracy.
- RandomForest had a much lower sensitivity than the others.
- Optimal vectorizer parameters for logreg, svm, nb all had max_df = 0.5 (ignore words/terms appearing in over 50% of posts).

Word Importance





More Data...

We were actually able to scrape and clean/process 142410 posts from r/math (51.6%) and r/physics (48.4%).

The Logistic Regression model produced above, trained on just ~3100 examples had an accuracy score of 0.86812, while the SVM model did even better with an accuracy of 0.87243!

Conclusion

Seems to be fairly easy to distinguish physics and math posts, our models achieved around 90% accuracy on unseen data.

Each subject has highly specific technical words which may make classification easy.

Future work could include trying to find subreddits with much closer linguistic/jargon overlap, performing multiclass classification on posts from 3 or more subreddits.

Thank you!