Reddit as a Window into Ancient Worlds

BY HARRY DZEBA

BACKGROUND

Researchers at a University have to stay current by keeping up with the latest texts published in their field

PROBLEM

With the staggering increase in the amount of info available online, they are finding it increasingly difficult and time-consuming to find all relevant materials that are being published

TASK

Leverage the power of machine learning and natural text processing to recognize topics relevant to researchers’ work

OUR APPROACH

Start with the history department, as the language used does not change as much as in other disciplines

- Train the model to recognize the difference between Ancient Rome and Greece subreddits
- Train the model to recognize multiple historic topics
- In the final stage, train the model in other academic disciplines

DATA CLEANING

Find relevant words to input into our model

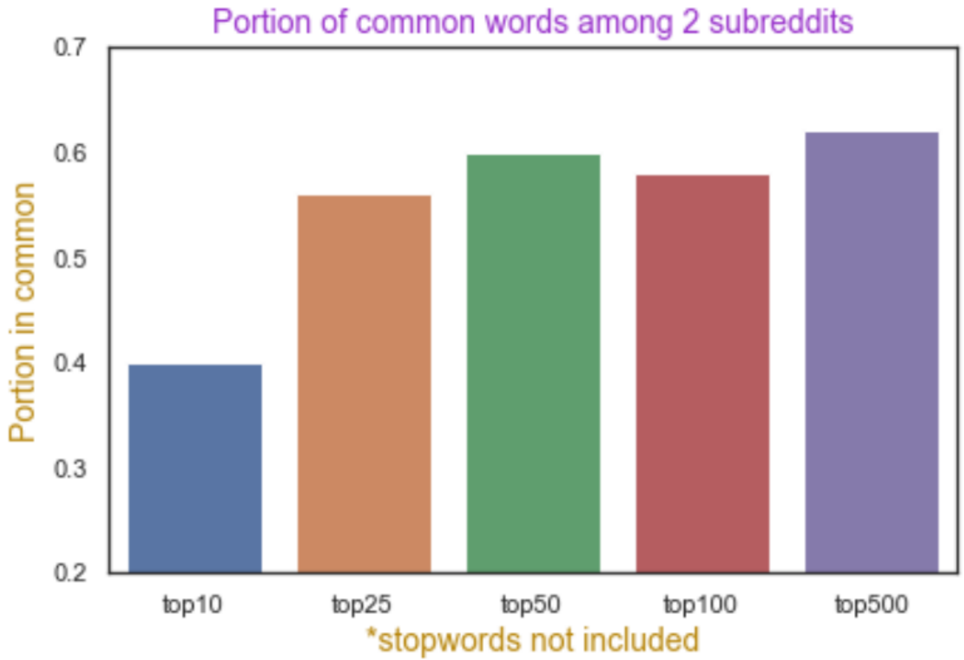
- Scrape Reddit posts
- Remove duplicates
- Merge, clean, lemmatize
- Stop words

What we uncovered during processing

3 key areas of interest:

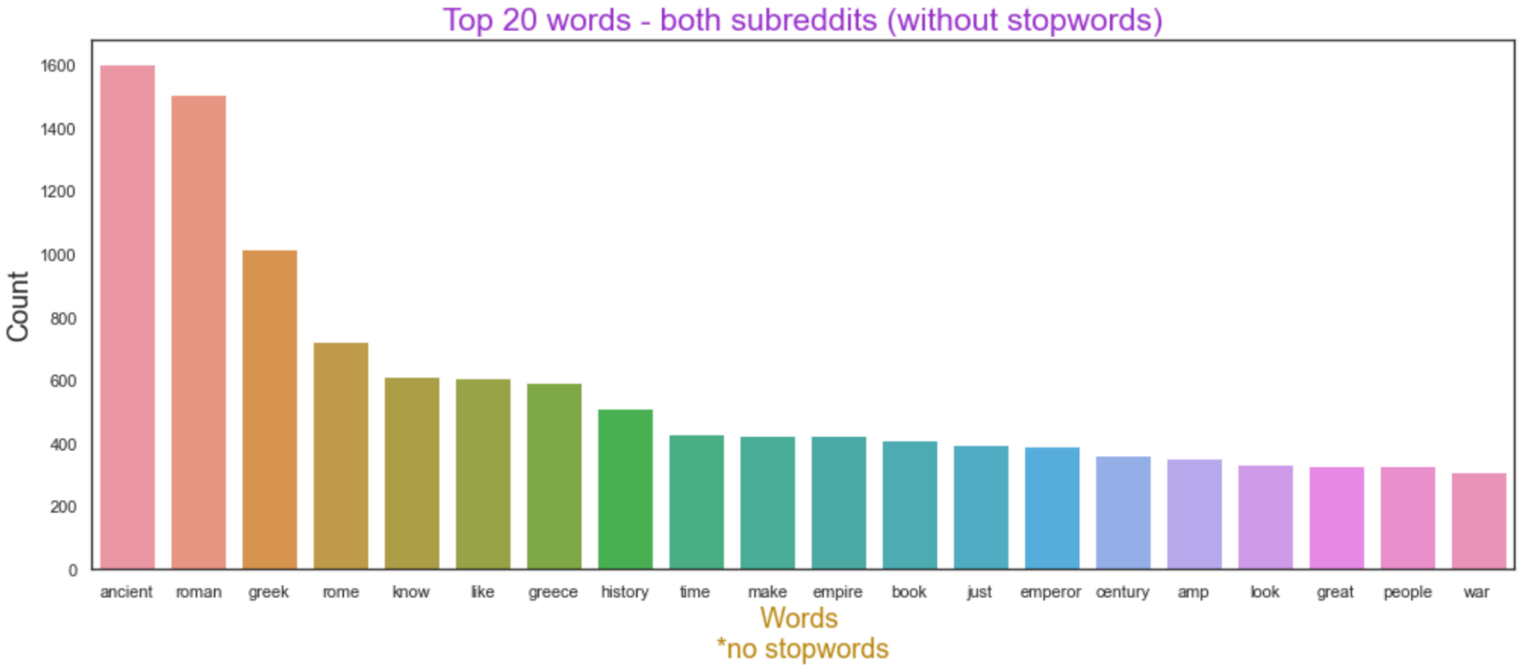
- Rome/Greece common words
  - **60%**
- Most common words on each subreddit
  - **Is it just a word-search?**
- Adding stop words
  - **Turn it into a true machine learning**

KEY OBSERVATION 1: words in common



- Percentage of common frequent words is between 40-60%, depending on how many top words are counted
- These lists do not contain common stopwords, so the true percentage is even higher

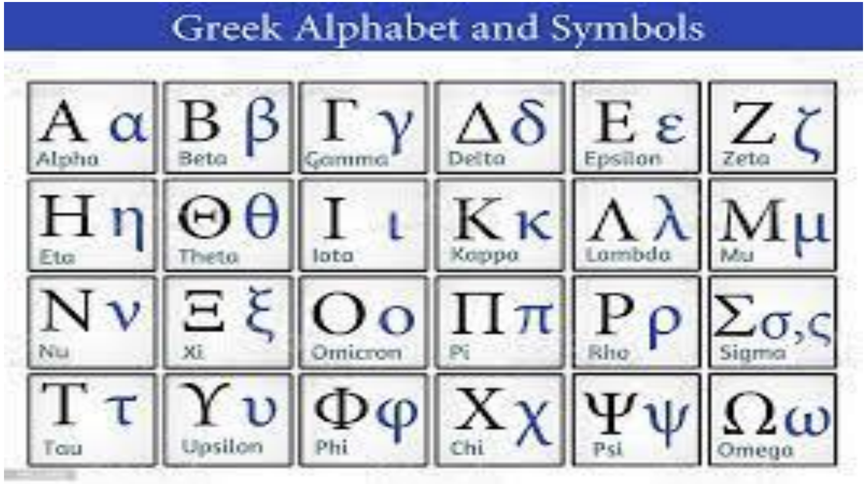
KEY OBSERVATION 2: frequent words



- Many of these words like greek, roman, Greece, Rome make it too easy for our model to distinguish the two forums.
- We will need to add the to stop words

KEY OBSERVATION 3: stop words

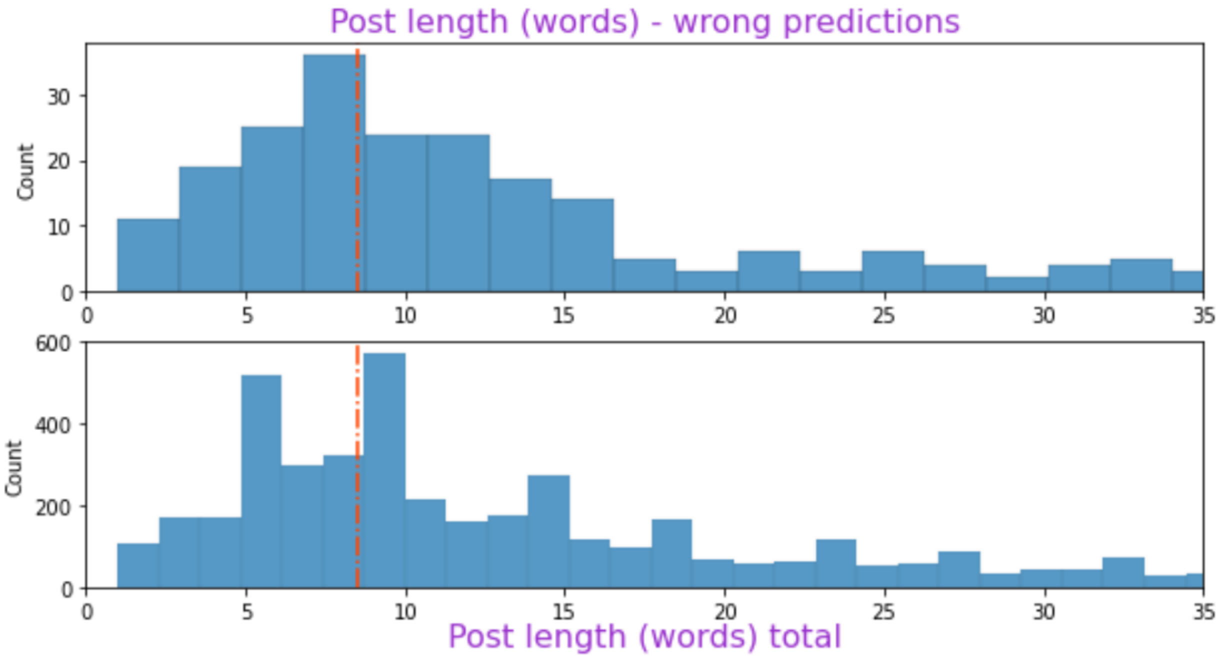
['ancient', 'roman', 'rome', 'romans', 'greek', 'greece', 'greeks', 'amp', 'know', 'like', 'make', 'look', 'just', 'use', 'rea
lly', 'ádám', 'ένα', 'αν', 'από', 'αρχαία', 'αἰδοῖον', 'γε', 'για', 'δεν', 'είναι', 'ζωή', 'θα', 'και', 'καὶ', 'κύβος', 'λοιμο
ῦ', 'με', 'μου', 'νέο', 'να', 'νεφέλην', 'οι', 'που', 'σε', 'στο', 'στον', 'τα', 'τη', 'την', 'της', 'τι', 'το', 'τον', 'του',
'τους', 'των', 'τῇ', 'τῶν', 'φωτογραφίες', 'χαίρετε', 'χωρίς', 'and', 'of', 'passed', 'עברתי', 'ἀθήναζε', 'ἀνερρίφθω', **'Award',
'Belfast', 'Doc', 'Dublin', 'Festival', 'Film', 'Greek', 'IndieCork', 'International', 'OFFICIAL', 'Rotterdam', 'SELECTION', 'Spirit',
'Thessaloniki', 'WINNER', 'of', 'the',** *'Director', 'Doyle', 'Dublin', 'Gráinne', 'Humphreys', 'IFF', 'Ronan', 'Scannain',* ***'BURNING',
'STREAMS', 'and', 'are', 'describe', 'difficult', 'drawn', 'embrace', 'hard', 'harder', 'if', 'knew', 'obscure', 'ones', 'onl
y', 'quantify', 'saw', 'subjects', 'that', 'the', 'they', 'to', 'world', 'would'***]



- On the top of list are some of the ‘easy’ words we excluded
- We have another list that doesn’t include Greek-alphabet words

MODELS

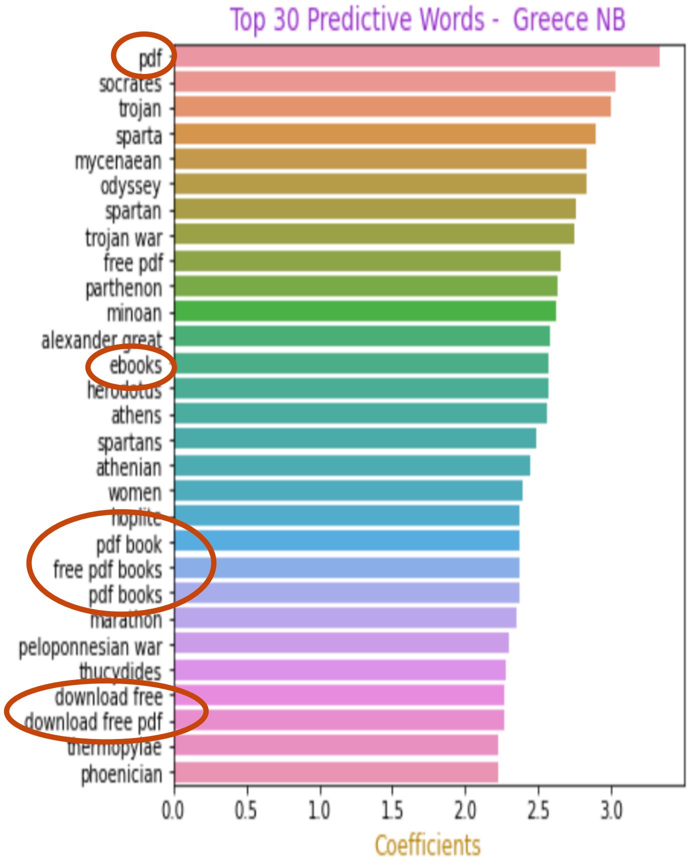
No	MODEL (vector)	train accuracy	test accuracy	parameters	other features	comments
1	Naive Bayes (countvec)	0.88	0.81	max_feat:4000 min_df:3 alpha:0.5	nothing extra - simple model	first model after baseline, already shown good improvement
2	Naive Bayes (countvec)	0.90	0.84	max_feat:2500 min_df:4 alpha:0.1	only posts with 8+ words	after testing the wrong predictions from model 1, only longer posts were taken. immediate improvement.
3	Naive Bayes (tfidf)	0.91	0.81	max_feat:4000 min_df:3 alpha:0.5	none	tfidf vectorization didn't bring about much improvement
4	Naive Bayes (tfidf)	0.93	0.85	max_feat:4000 min_df:3 alpha:0.5	only posts with 8+ words	- best score so far - minimal tuning required, so very easy and quick to run
5	Naive Bayes (countvec)	0.87	0.80	max_feat:4000 min_df:3 alpha:0.5	extra stopwords	- similar results to model 1 but with more stopwords, therefore considered a succes
6	Naive Bayes (tfidf)	0.93	0.84	max_feat:4000 min_df:3 alpha:0.5	only post with 8+ words, and extra stopwords	- simple model with best scores - production model - simple to tune
7	Random Forest (countvec)	0.81	0.78	max_feat:4000 criterion:entropy max_depth:5 min_leaf:2 estimators:300	extra stopwords	-random forest did not show as high accuracy as NB - predicted Greece 60% of the time
8	Random Forest (tfidf)	0.81	0.76	max_feat:4000 criterion:entropy max_depth:12 min_leaf:1 estimators:300	only post with 8+ words, and extra stopwords	-hard to tune. started with params for model 7, but it wasn't even close - accuracy was initially awful, but with very high max_depth got close to model 7 -predicted Rome 67% of the time
9	2nd deg Poly SVM, (countvec)	0.94	0.80	max_feat:7500 C:1 coef0:2	extra stopwords	-big difference between train and test - indicative of overfitting - no amount of parameter tuning could reduce overfitting -predicted greece 57% outcomes
10	2nd deg Poly SVM, (tfidf)	0.99	0.83	0.901	only post with 8+ words, and extra stopwords	-even worse overfitting than model 9 - adding greek stopwords made the model balanced in its prediction (50:50)



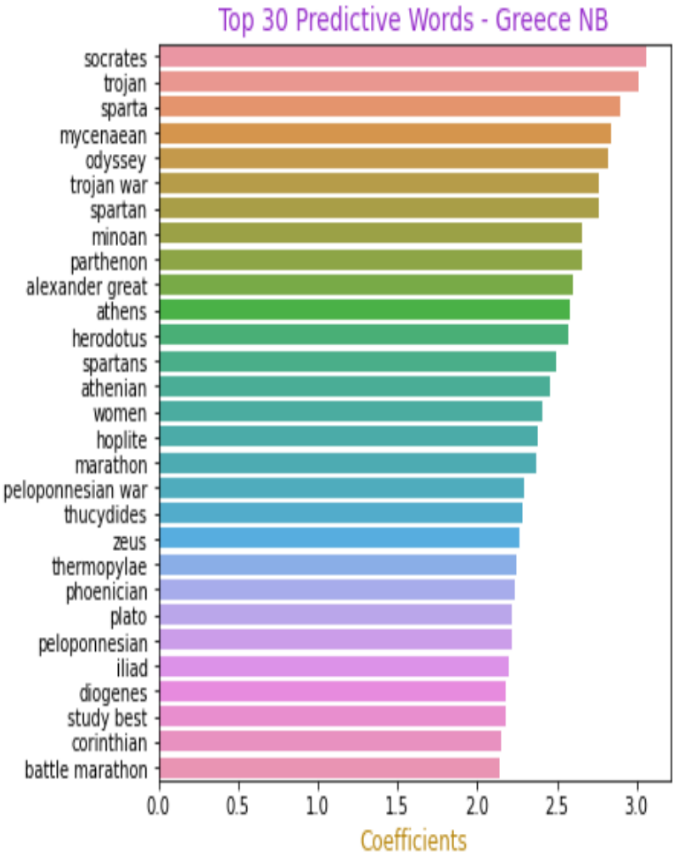
Key Influencer:

Length of posts

BEFORE



AFTER



Naïve Bayes

- Generally best performing group of models
- NB no 6 was chosen as our production model

BEFORE

	Predicted Rome	Predicted Greece
Actual Rome	476	225
Actual Greece	79	605

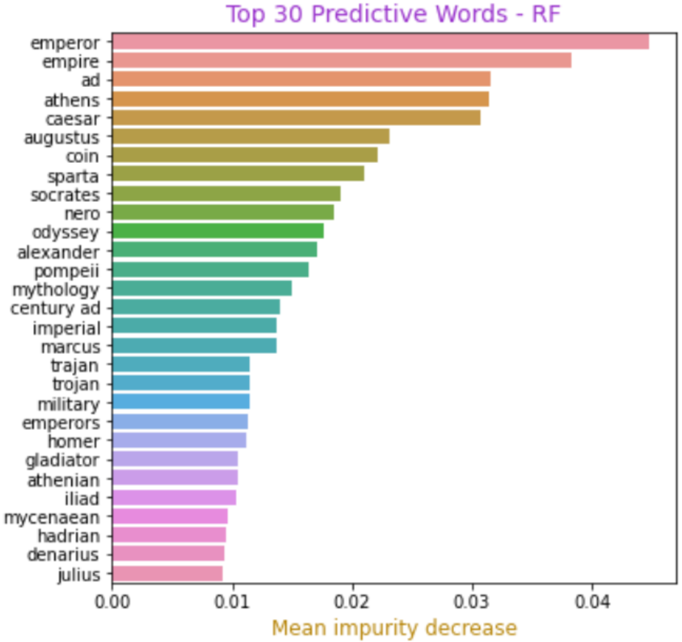
AFTER

	Predicted Rome	Predicted Greece
Actual Rome	476	41
Actual Greece	195	276

Random Forest

- Scores not as high as NB, but less overfitting (0.81 accuracy on the test set, and 0.78 on train)
- Confusion matrix is completely changed after adding more stop words

Random Forest Cont'd



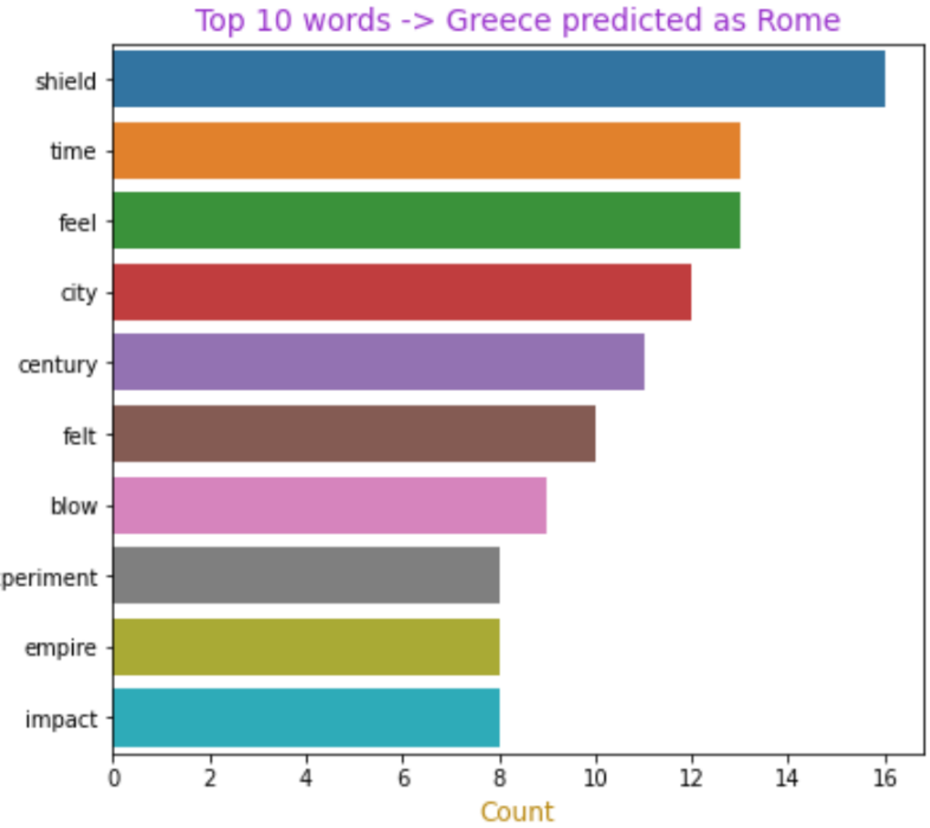
- Words with the highest predictive value within the model

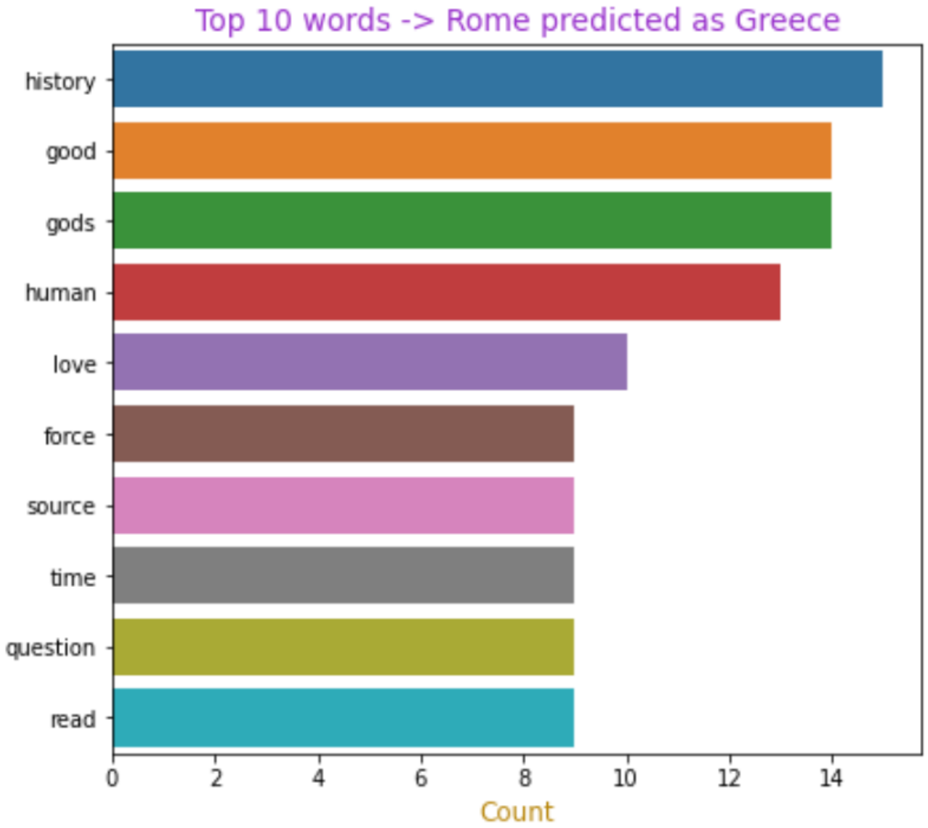
Support Vector Machine

```
==============================
==== SVM/tfidf & 8+words =====
==============================

Scores:
------------------------------

train score: 0.99
test score: 0.83
cross-validated score: 0.8
```

- Very big difference between testing and training data
- Require lots of tuning. Unreliable

GREECE AND ROME MISCLASSIFIED





CONCLUSIONS

- Proper nouns are 75% of our most import words across models
- Rome and Greece were similar periods in human history, but they had different political structures, which lend themselves to different words used to describe them
- Choosing stop words can make or break model. Input from experts ought to be sought before composing the list
- Regularly check for the words that cause predictions to be wrong. Exclude them if they hurt the model performance

MODEL IMPROVEMENTS

- Recognition of different alphabets and characters
- Have an expert-approved list of stop words
- Automate searching for most common words causing wrong predictions, taking them out of predictors list and re-run the model

NEXT STEPS

We envision this entire project to be just the first step in developing internet search for any material that relates to any of the University's academic departments. The next steps are:

- Expand the model to more historical topics, work on multiple topics at one time
- Configure the model to recognize ‘others’ category
- Repeat the process for other academic fields