Natural Language Processing with reddit

By Bruno Santos
Data Scientist

Our Data Problem

- Considered “the front page of the internet”
- Broken down by “subreddits” - separate pages for categories or topics
- Pick two subreddits and use NLP to predict the origin of a post
- How many models will we use?
- How many features are optimal?
- How accurate are our models in their predictions?

Behind the Scenes

- Scrape reddit API and gather pertinent data
- Subreddits of choice: LegalAdvice, PersonalFinance
  - LegalAdvice - “A place to ask simple legal questions.” - 54.65% of total
  - PersonalFinance - “Get your financial house in order, learn how to better manage your money, and invest for your future.” 45.35% of total
- Clean Data
- Build / Tune Models
- Score Models!

Basic Logistic Regression - GridSearch

CountVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 2

Max Features: 3000

Ngram Range: Bigrams

Stop Words: None

Accuracy: 96.90% Train / 83.88% Test

TfidfVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 2

Max Features: 1500

Ngram Range: Unigrams

Stop Words: None

Accuracy: 91.26% Train / 84.67% Test

Multinomial Naive Bayes - GridSearch

CountVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 2

Max Features: 1500

Ngram Range: Unigrams

Stop Words: None

Accuracy: 91.17% Train / 86.25% Test

TfidfVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 2

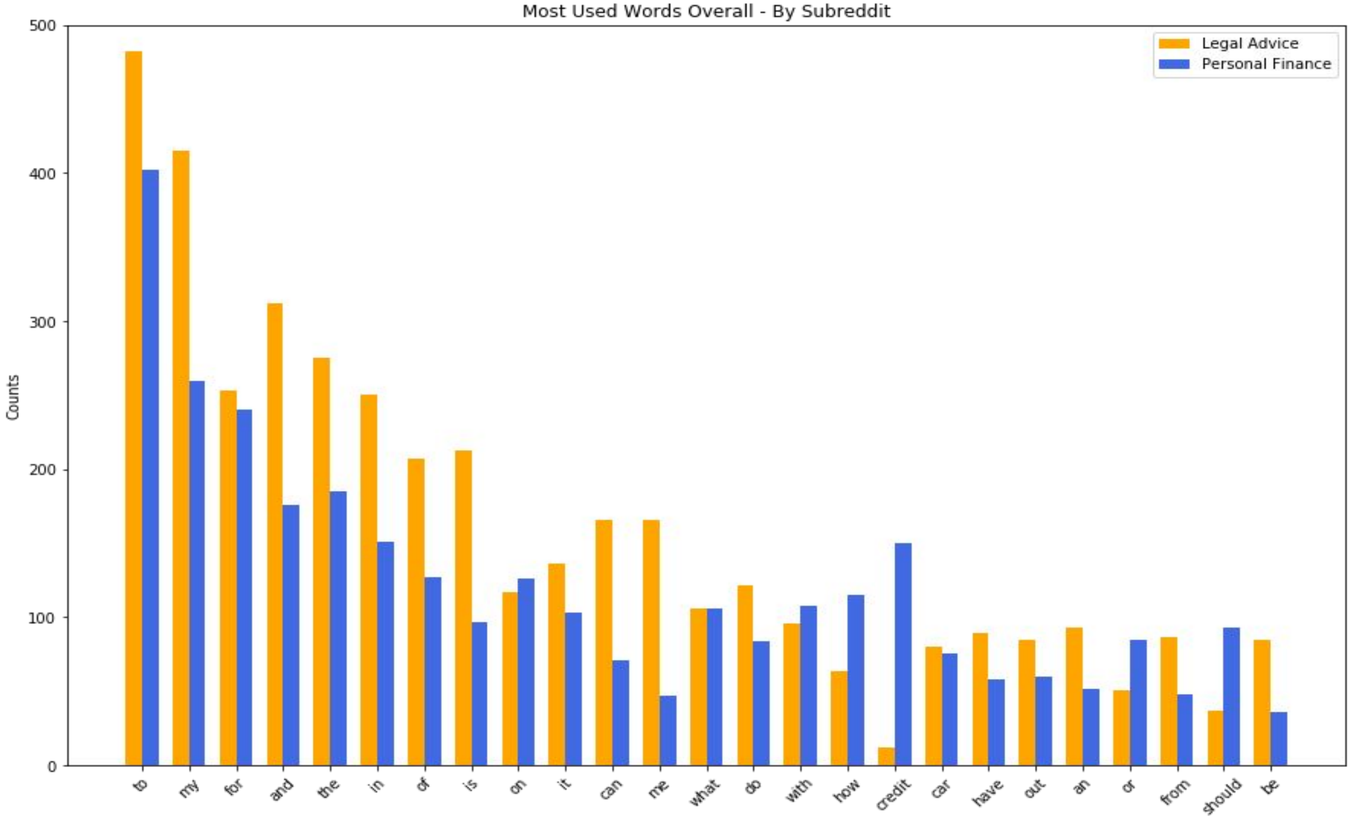
Max Features: 1500

Ngram Range: Unigrams

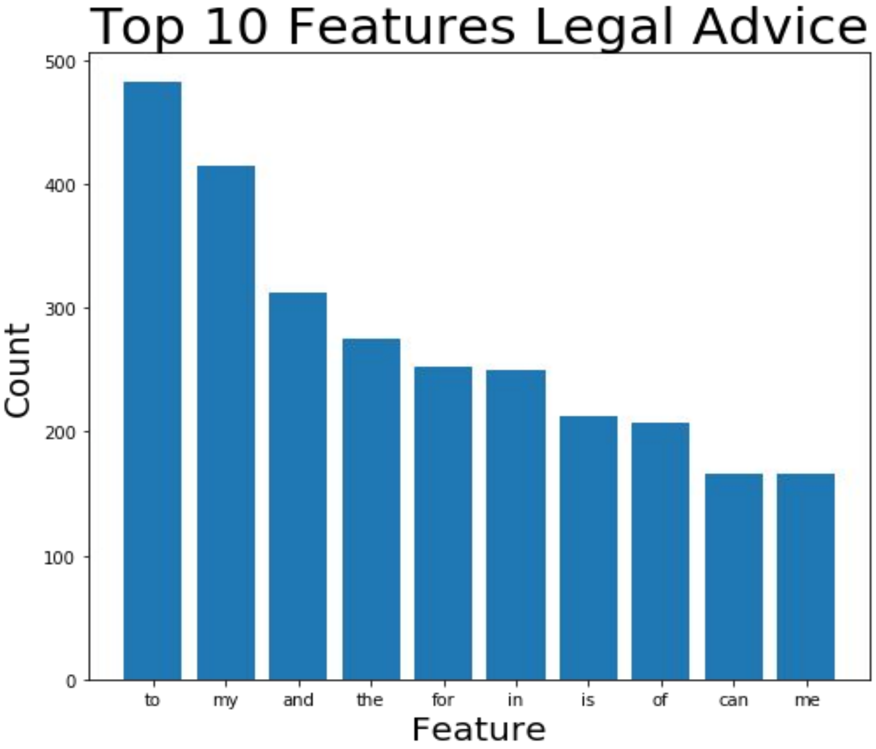
Stop Words: None

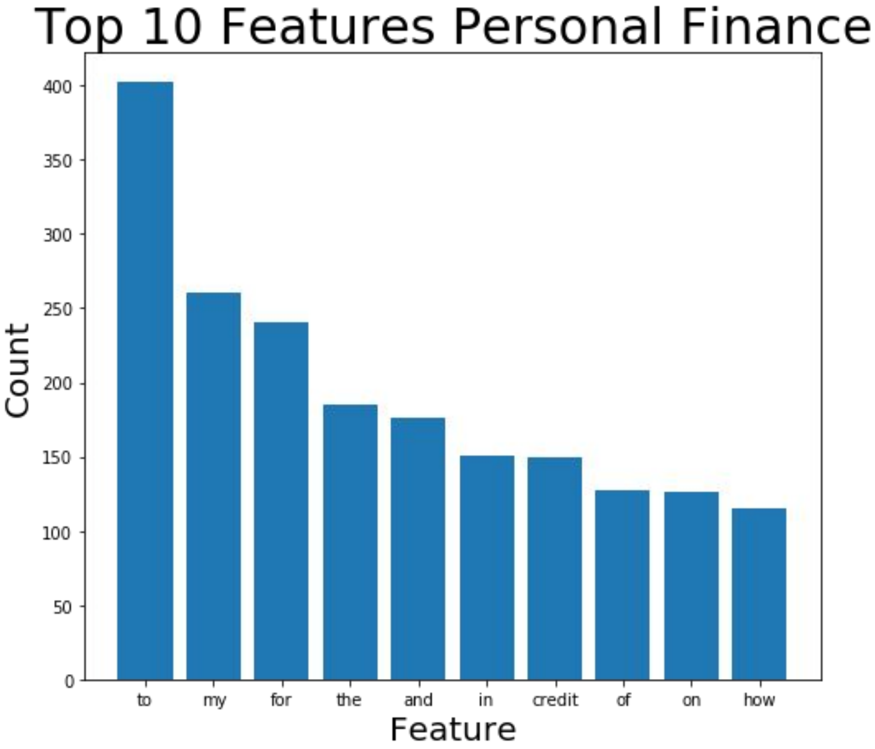
Accuracy: 92.02% Train / 85.66% Test

Top 25 Word Count w/o StopWords Removed



CountVectorizer Top Features w/o StopWords Removed





Multinomial Naive Bayes

CountVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 2

Max Features: 1000

Ngram Range: Unigrams

Stop Words: English

Accuracy: 88.92% Train / 83.68% Test

TfidfVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 3

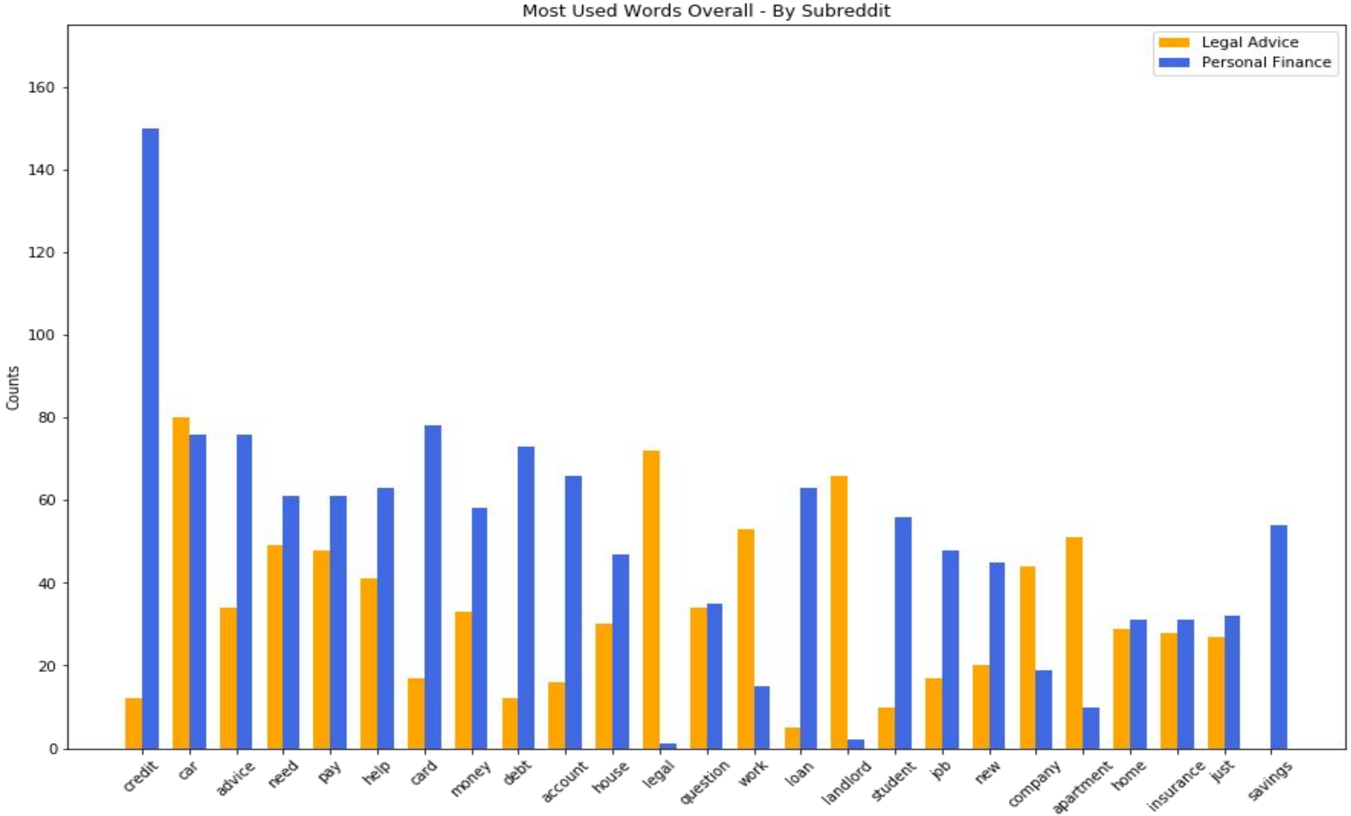
Max Features: 1000

Ngram Range: Unigrams

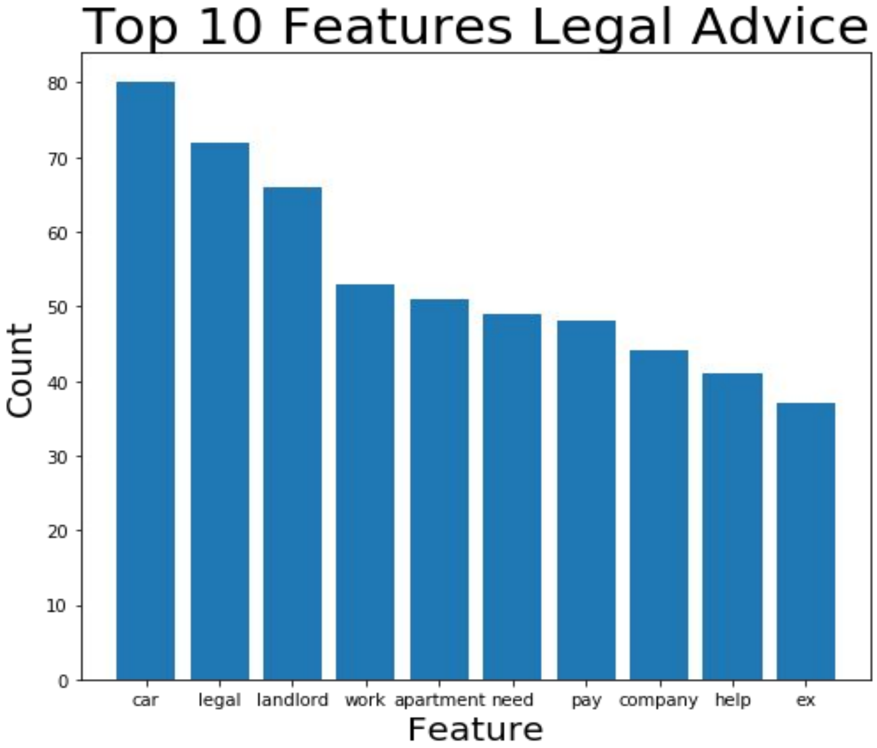
Stop Words: English

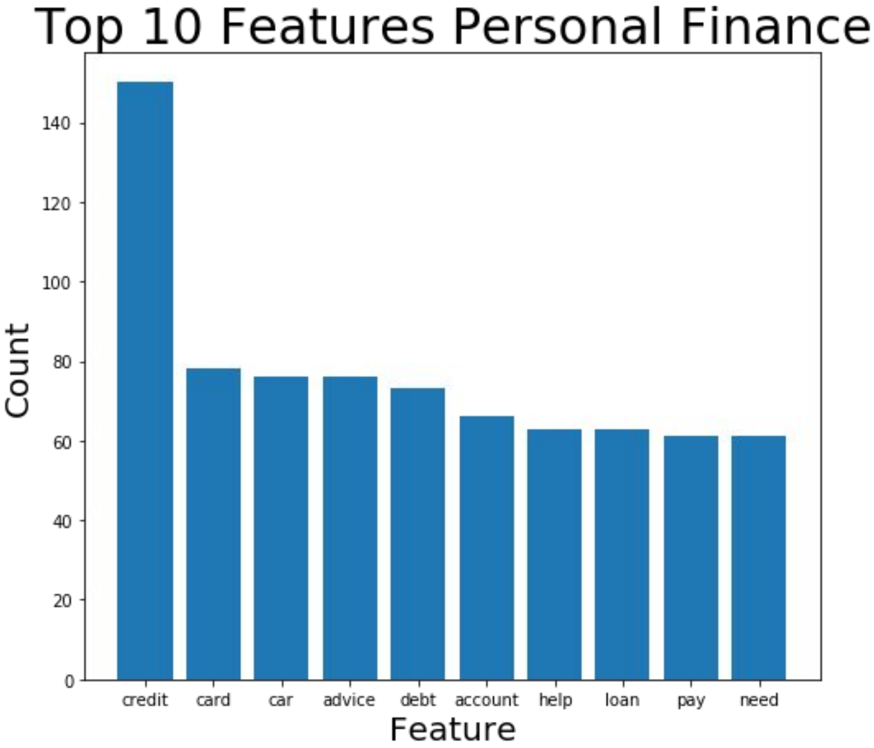
Accuracy: 90.24% Train / 84.17% Test

Top 25 Word Count w/o English StopWords

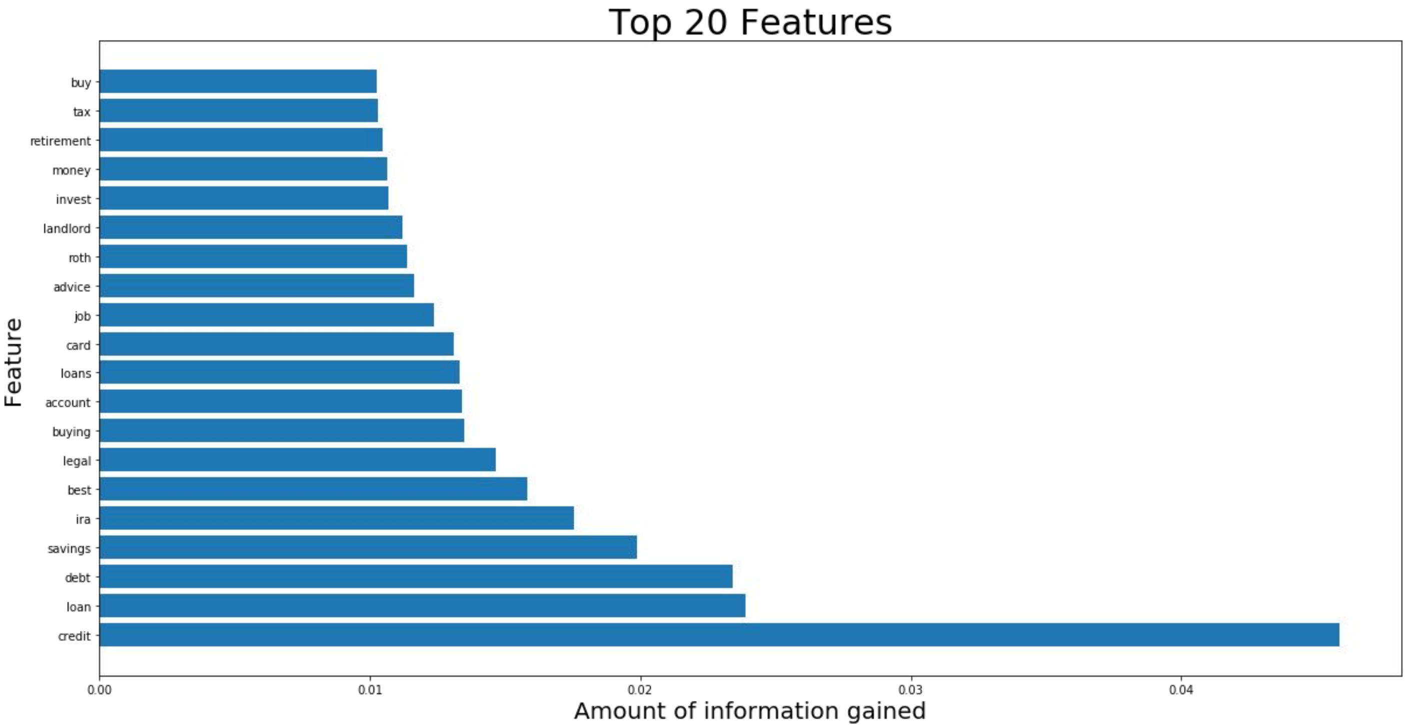


CountVectorizer Top Features w/o English StopWords





TfidfVectorizer Feature Importance



RandomForestClassifier

Parameters

Max Depth: None

Min Samples Leaf: 1

Max Samples Split: 50

Number Estimators:

CV - 50

Tfidf - 100

CountVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 2

Max Features: 1000

Ngram Range: Unigrams

Stop Words: English

Accuracy: 89.90% Train / 80.12% Test

TfidfVectorizer

Max Document Frequency: 0.75

Min Document Frequency: 3

Max Features: 1000

Ngram Range: Unigrams

Stop Words: English

Accuracy: 94.27% Train / 80.61% Test

Conclusions / Recommendations

- Our best performing model was Multinomial Naive Bayes.
  - No StopWords removed, GridSearched hyperparameters
- Removing StopWords lowered model performance but gave a more clear insight into the language of subreddits.
- Significantly increasing features causes overfitting.
- In future tests - would want to pick more closely related subreddits, try different models, and more hyperparameters.

Questions?