Reddit Classification

r/dating vs. r/datingoverforty

Problem Statement

- Client: weird, reclusive dude
- Thinks all dating/relationship related posts sound the same
- Wants to see if a machine could tell the difference
- Asked for the most accurate classifier of r/dating and r/datingoverforty posts

Data

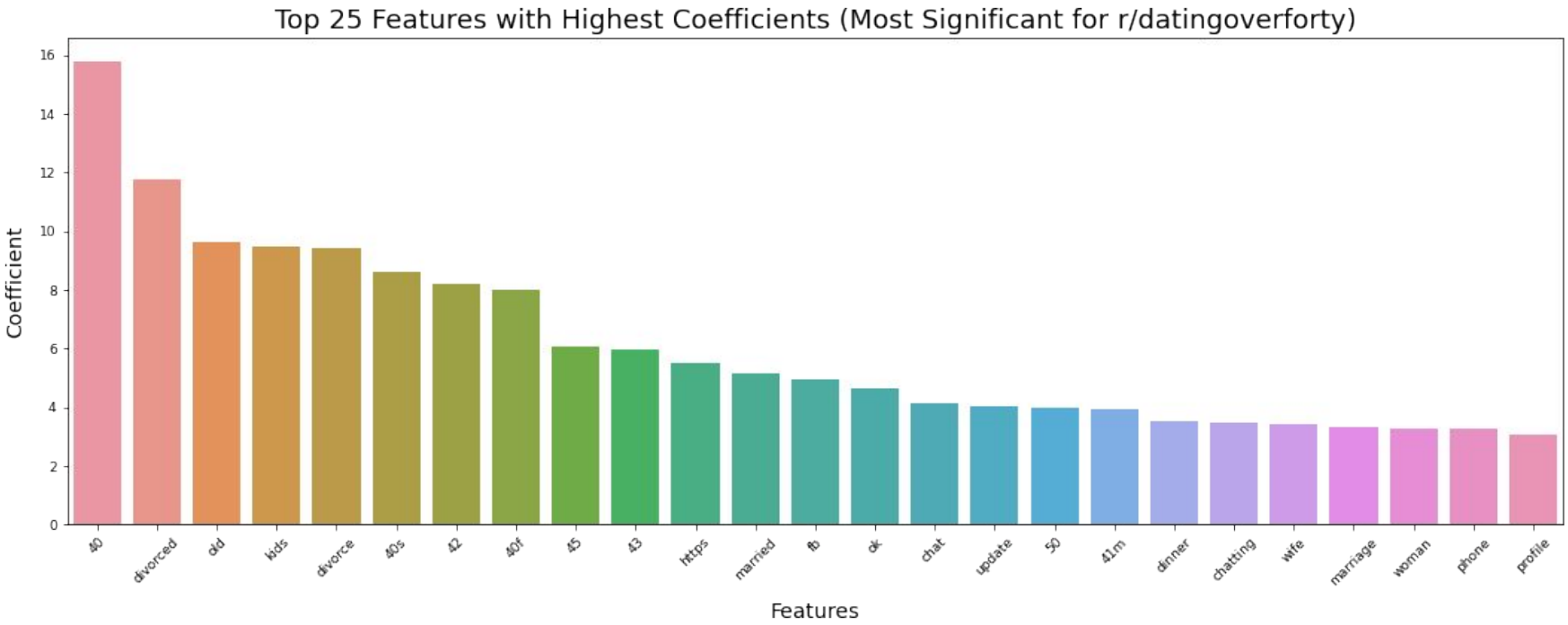
- Gathered posts from both subreddits starting from the end of March 2021 to late March 2022
- Posts way more frequent on r/dating
  - 92% of combined dataset
- Downsampled the majority class

Exploratory Data Analysis

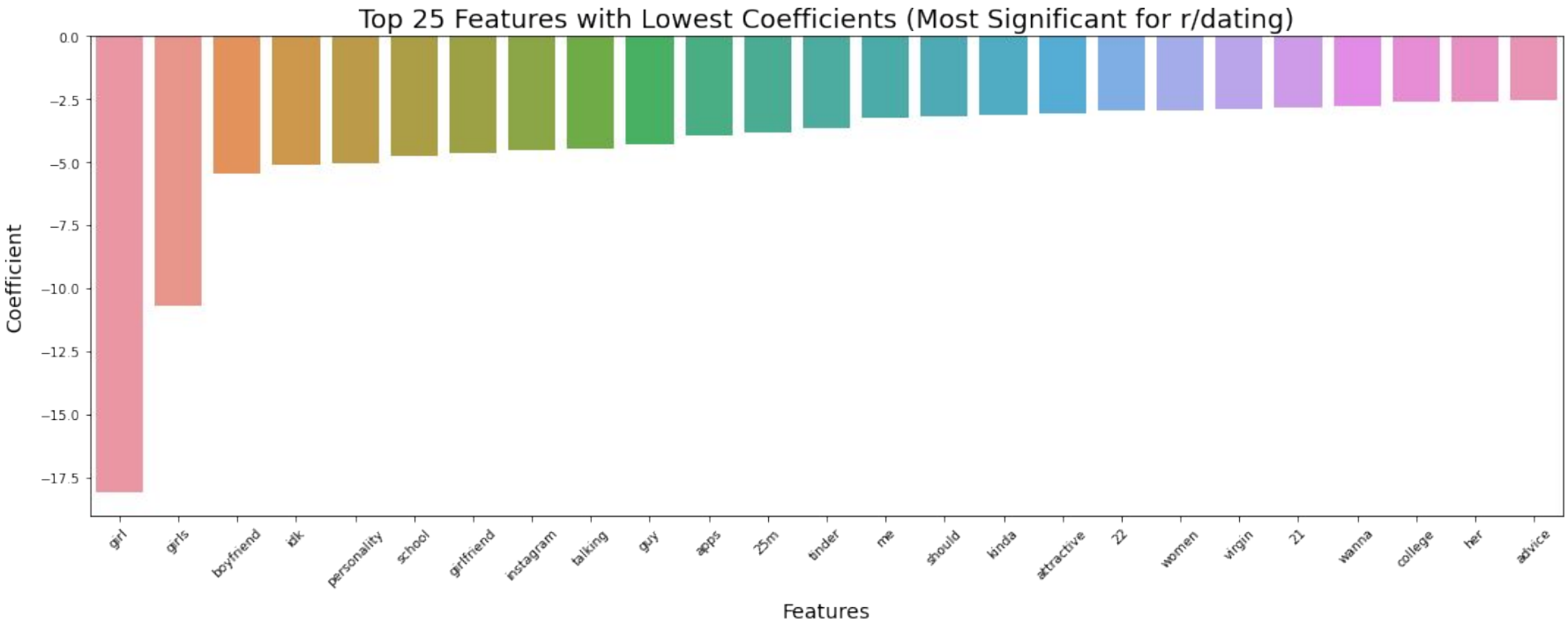
---

1. Used Logistic Regression with lasso regularization to see which words had highest coefficients
2. Looked at most common words shared by both subreddits to find stop word candidates

Significant Words in r/datingoverforty



Significant Words in r/dating



Topics/Words

---

r/dating

- Using the word “girl”
- Numbers in the 20s
- School/College

r/datingoverforty

- Numbers over 40
- Divorce
- Children
- Marriage

Most common words in both

just	like	date	dating	ve
want	time	relationship	think	feel
people	don	really	know	said
didn	things	going	good	person

Feature Engineering/Preprocessing

- Function replacing different variations of topic words with one word
- Used RegEx to match different variations
- Done before text vectorization in pipeline

Before	After
girl, girls, girlfriend, girly . . .	girl
marriage, marry, marrying . . .	marriage
divorce, divorced, divorcing . . .	divorce
40, 43m, f53, 50s, 47yr . . .	over_forty
20, 23m, 28f, 30s . . .	under_forty
kids, kiddo, child, grandson . . .	kids
school, college, university	school

Modeling

- Logistic Regression and Multinomial Naive Bayes
- 4 Separate Grid Searches for Each
  - Replacing Topic Words
  - CountVectorizer / TfidfVectorizer
- Tested different stop words combinations in Grid Search
  - No stop words
  - Default stop words
  - Default with additional common words

Logistic Regression and Multinomial Naive Bayes Results

Model	Consolidated Topic Words	Vectorizer	Best Score (train)	Best Score (test)
Logistic Regression	Yes	Count	0.918557	0.810209
Logistic Regression	Yes	Tf-idf	0.909395	0.820244
Logistic Regression	No	Count	0.918994	0.811518
Logistic Regression	No	Tf-idf	0.911576	0.818499
Multinomial Naive Bayes	Yes	Count	0.863583	0.805846
Multinomial Naive Bayes	Yes	Tf-idf	0.879290	0.799738
Multinomial Naive Bayes	No	Count	0.866492	0.806283
Multinomial Naive Bayes	No	Tf-idf	0.880454	0.800175

More Modeling

- Gridsearch using:
  - Ada Boost Classifier (w/ Decision Tree Classifier base)
  - Random Forest Classifier
  - Kernel Support Vector Machine Classifier
- Consolidate topic words and used Tf-idf Vectorizer for all
- Still Gridsearched over different sets of stop words

Ada Boost, Random Forest, SVM Best Scores

Model	Best Score (train)	Best Score (test)
Ada Boost Classifier	0.999709	0.745201
Random Forest Classifier	0.895870	0.799738
Kernel Support Vector Machine Classifier	0.982403	0.696771

Best Model?

Model	Consolidated Topic Words	Vectorizer	Best Score (train)	Best Score (test)
Logistic Regression	Yes	Count	0.918557	0.810209
Logistic Regression	Yes	Tf-idf	0.909395	0.820244
Logistic Regression	No	Count	0.918994	0.811518
Logistic Regression	No	Tf-idf	0.911576	0.818499
Multinomial Naive Bayes	Yes	Count	0.863583	0.805846
Multinomial Naive Bayes	Yes	Tf-idf	0.879290	0.799738
Multinomial Naive Bayes	No	Count	0.866492	0.806283
Multinomial Naive Bayes	No	Tf-idf	0.880454	0.800175

Next Steps

- Explore adding more features about text
  - Sentiment analysis
  - Word counts/sentence counts
- More model tuning/exploration
- Look into other ways to analyze text besides just vectorization