Relationship advice & Petty revenge

GA DSI Project 3
April 2021

Problem

A disgruntled moderator left his forums in ruins as you wrested the moderator badge from his chest.

We want to come up with methods for classifying text blocks from these two subreddits back into their proper homes. These methods can be applied to other NLP tasks in the future .

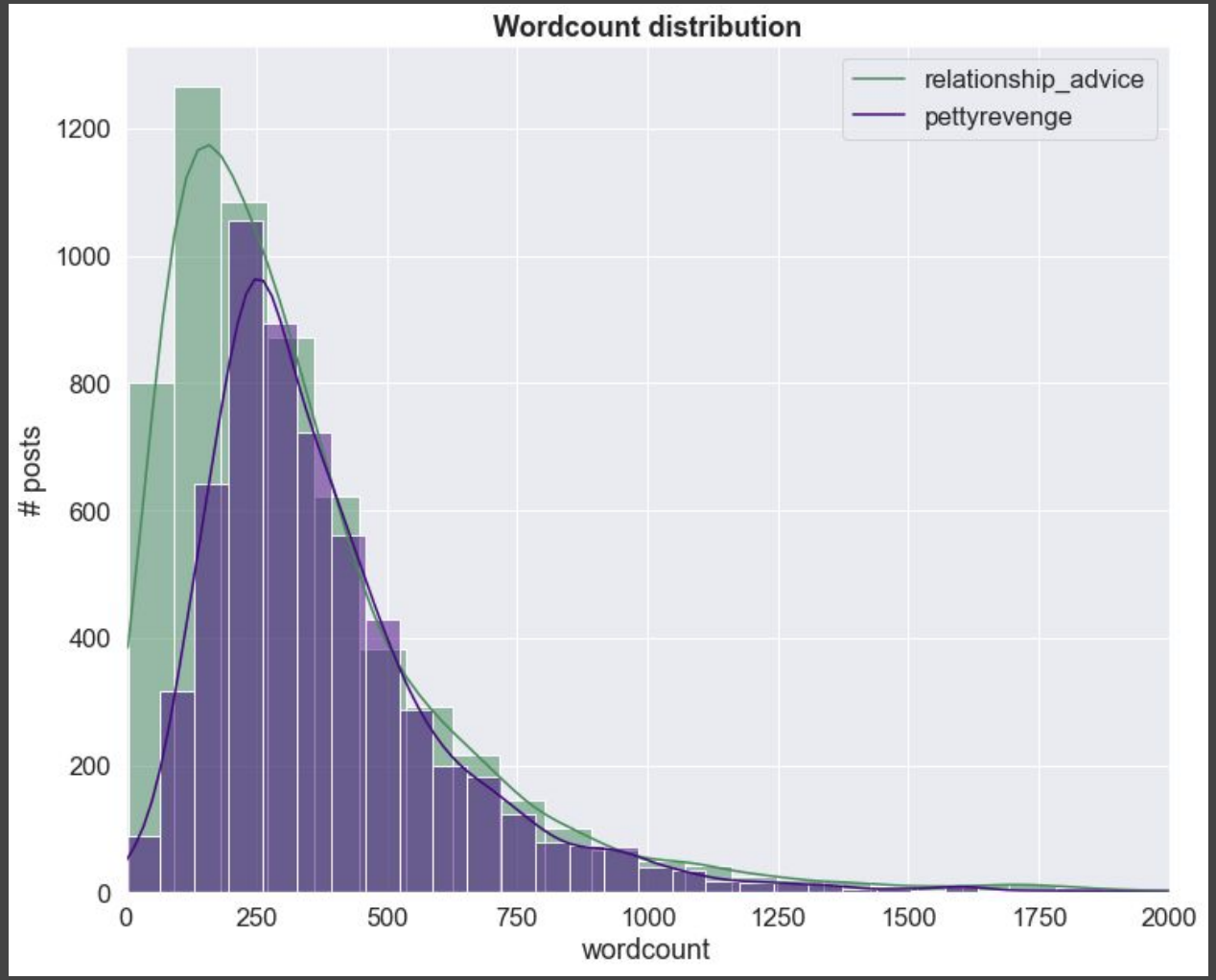
Problem

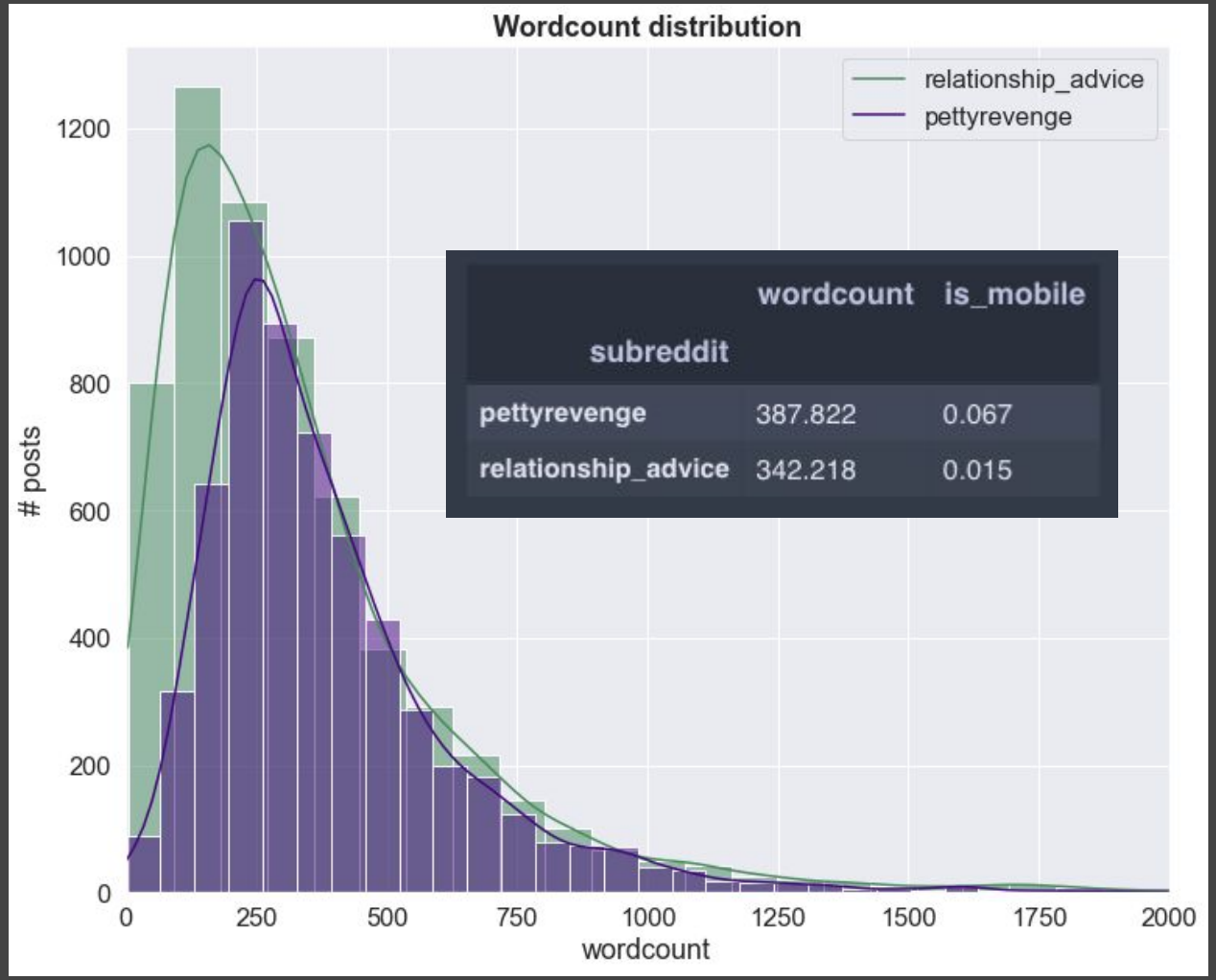
A disgruntled moderator left his forums in ruins as you wrested the moderator badge from his chest.

We want to come up with methods for classifying text blocks from these two subreddits back into their proper homes. These methods can be applied to other NLP tasks in the future.

Data description

- **Source: Reddit API pushshift.io**
- **Relationship_advice: 5M members → 6,093 (51%)**
- **Pettyrevenge: 1M members → 5,907 (49%)**
- **Sampled over same timeframe → 2016 to present**


Wordcount distribution
relationship_advice
pettyrevenge
posts
0
200
400
600
800
1000
1200
0
250
500
750
1000
1250
1500
1750
2000
wordcount



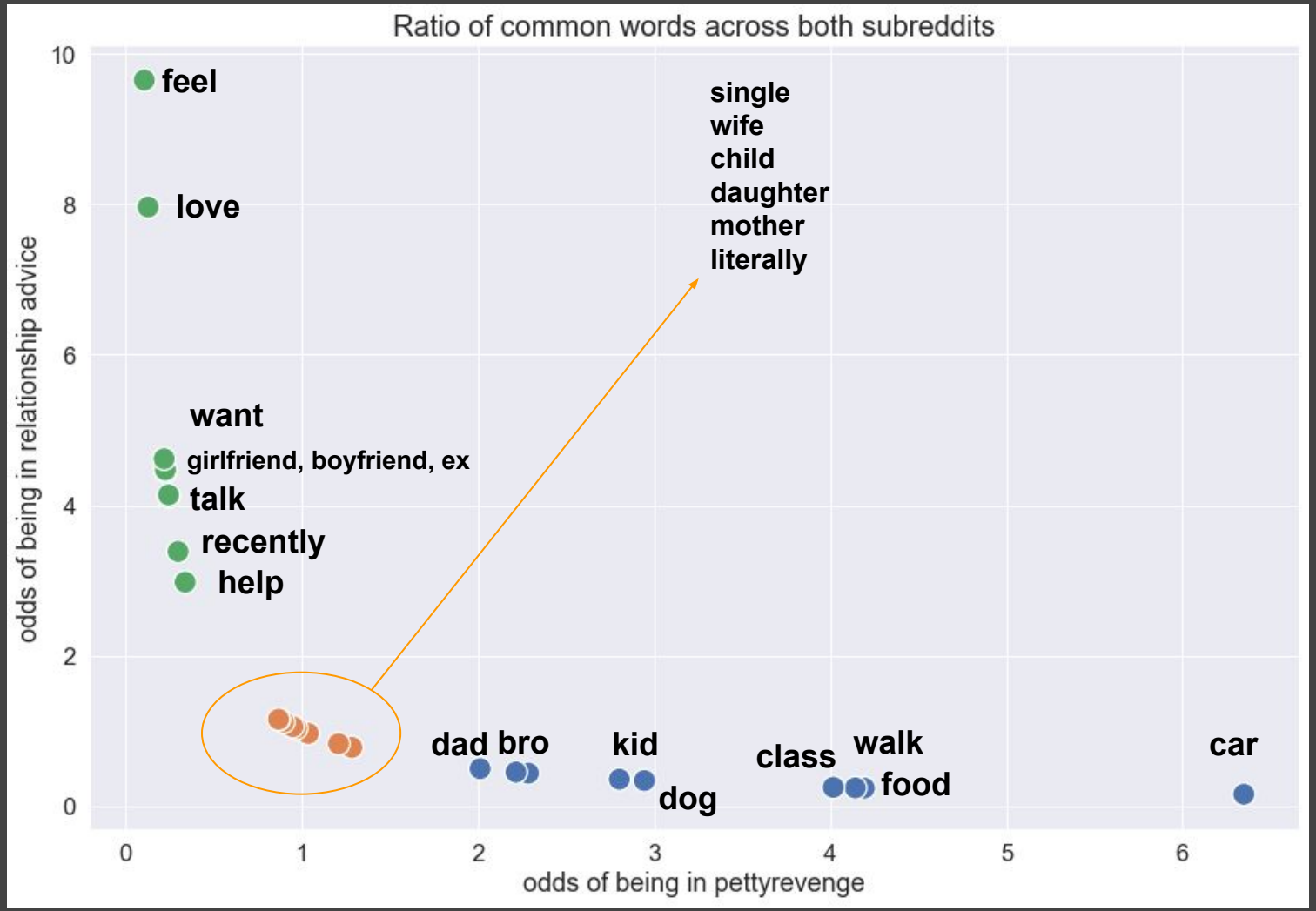
subreddit	wordcount	is_mobile
pettyrevenge	387.822	0.067
relationship_advice	342.218	0.015

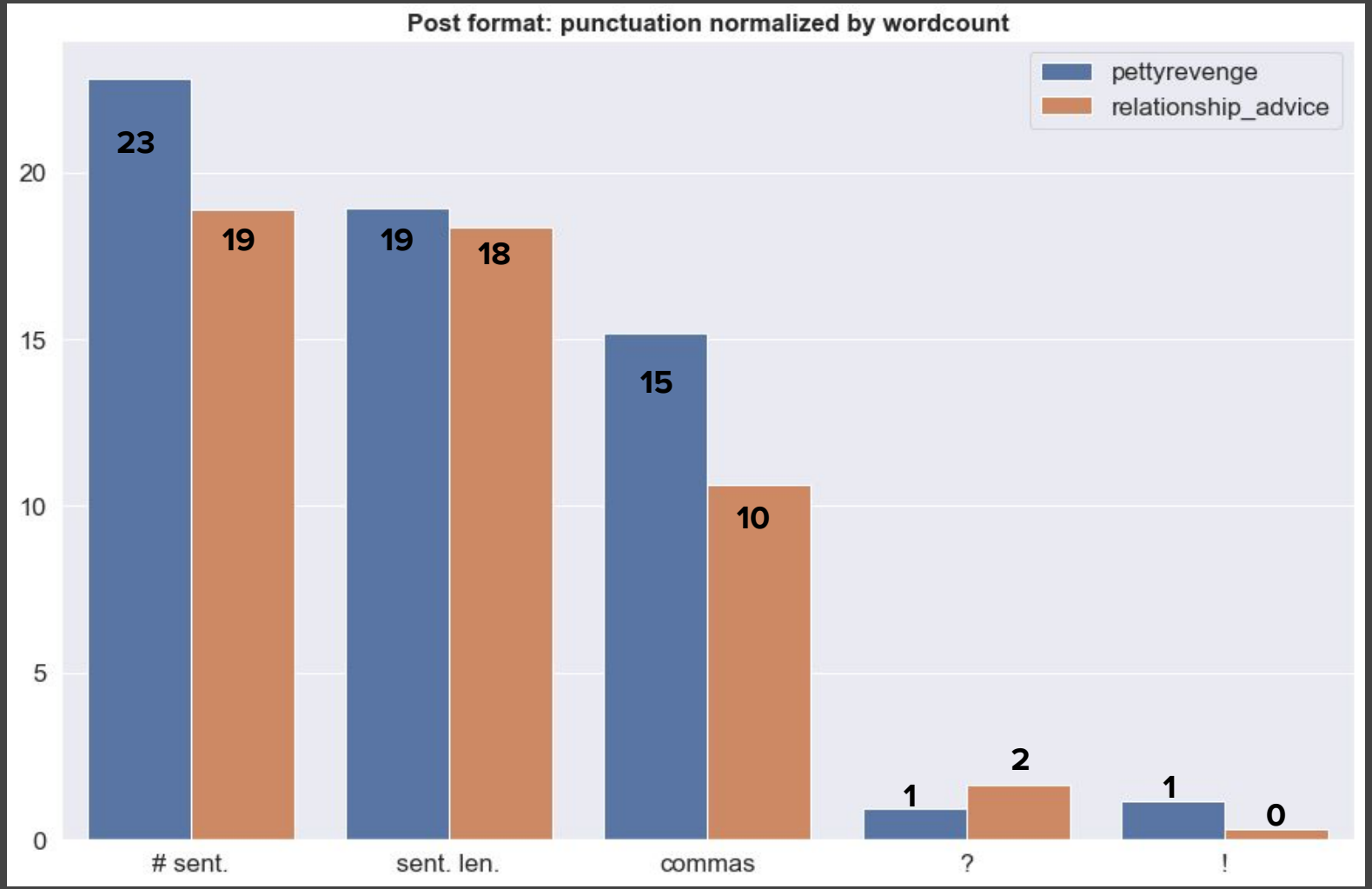
PR average wordcount: 388

RA average wordcount: 342

A mobile post is 4 .5x more likely to be from PR than RA


know
didnt
day
love
say
like
just
feel
make
told
want
work
said
thing
think


Ratio of common words across both subreddits
odds of being in relationship advice
odds of being in pettyrevenge
0
2
4
6
8
10
0
1
2
3
4
5
6
feel
love
want
girlfriend, boyfriend, ex
talk
recently
help
single
wife
child
daughter
mother
literally
dad
bro
kid
dog
class
walk
food
car

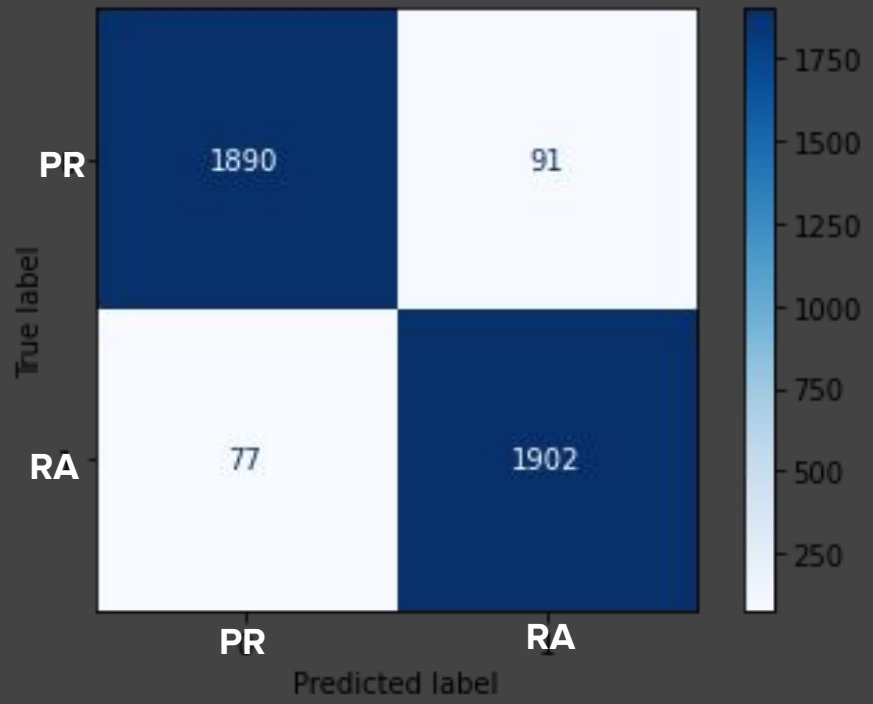

Post format: punctuation normalized by wordcount
pettyrevenge
relationship_advice
0
5
10
15
20
23
19
19
18
15
10
1
2
1
0
sent.
sent. len.
commas
?
!

Pre-Processing

1. Remove all URLs
2. Remove all punctuation
3. Remove all Chinese characters which is a thing that passes isalpha()...
4. Lemmatize the text
5. Custom stop_words
   a. ‘English’
   b. Lemmatized ‘english’
   c. ‘Relationship’, ‘advice’, ‘petty’, ‘revenge’

Model 1: Multinomial Bayes

with CountVectorizer



- RandomSearchCV
- Best parameters:
  - max_df=100
  - Max_features = 6112
  - Ngram_range (1,3)
  - Mnb_alpha = .3077

Scores

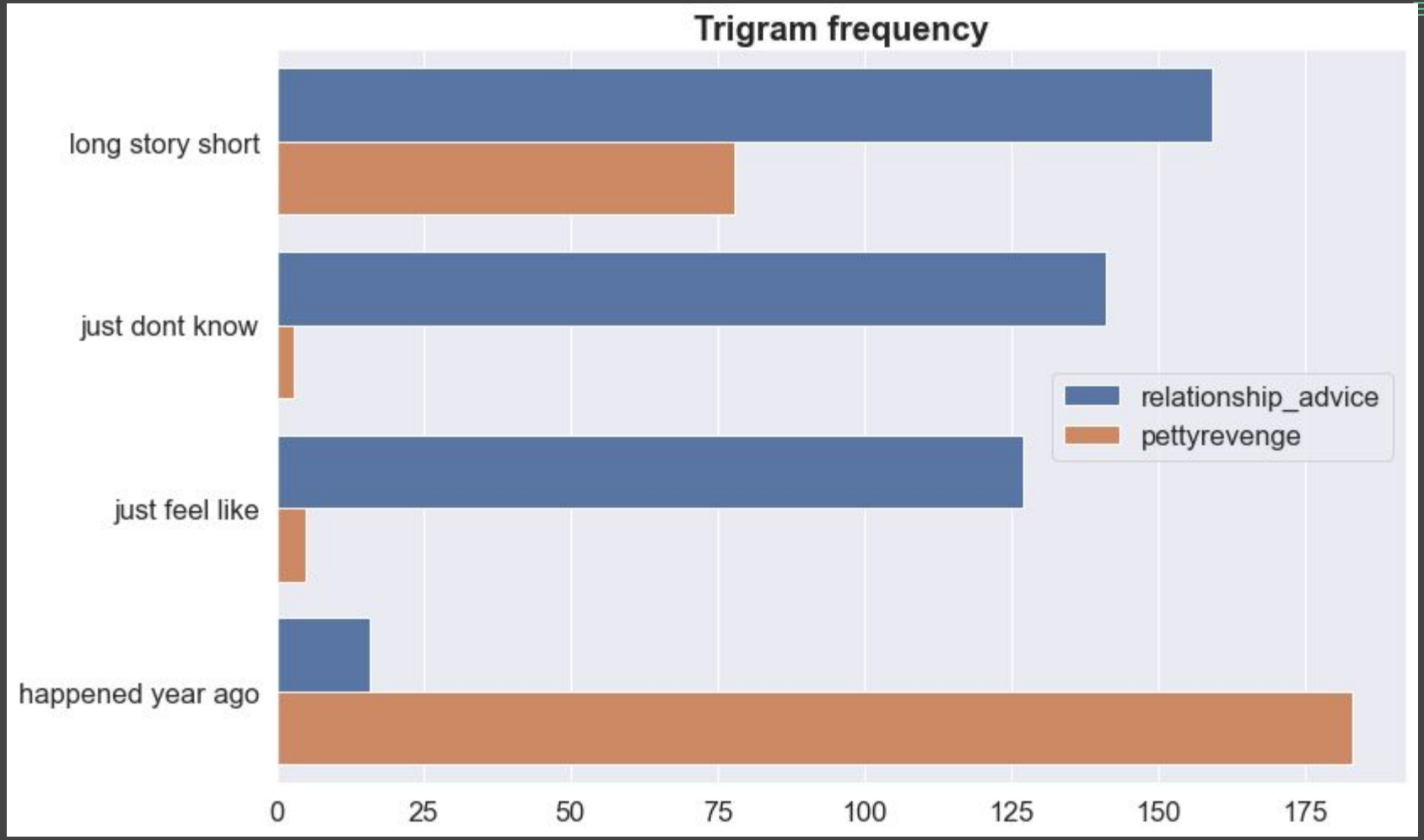
F1: 0.9577039274924471

Accuracy: 0.957576

Sensitivity: 0.961091

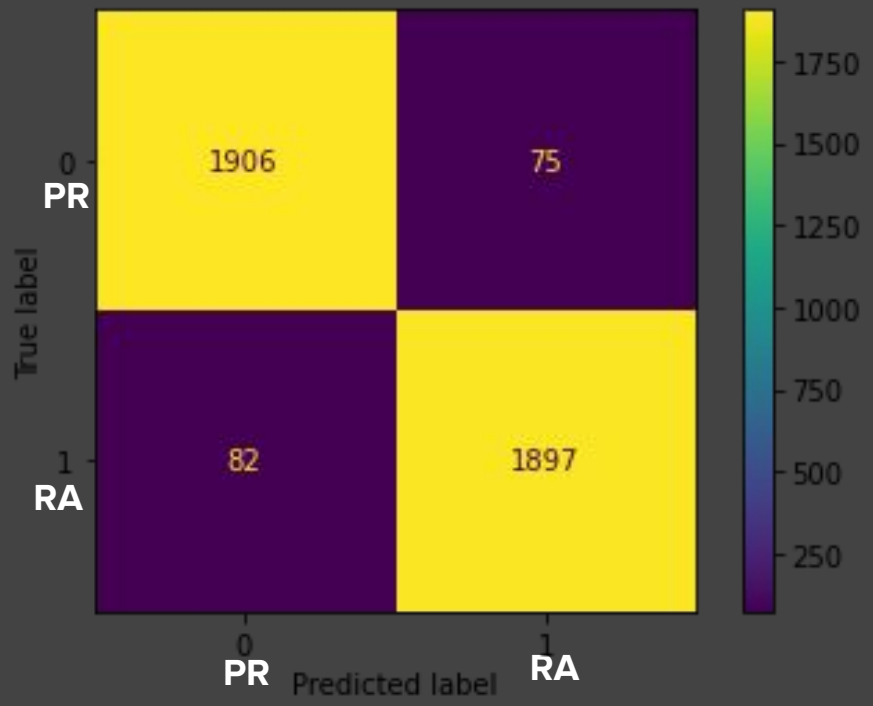
Precision: 0.9543401906673357

Specificity: 0.954064


Trigram frequency
long story short
just dont know
just feel like
happened year ago
relationship_advice
pettyrevenge
0
25
50
75
100
125
150
175

Model 2: Support Vector Classification

with Tf-Idf



- RandomSearchCV
- Best parameters:
  - max_df=0.93
  - Max_features = 4739
  - Ngram_range (1,3)
  - C = 0.63

Scores

```
Accuracy: 0.960354

Sensitivity: 0.958565

Precision:
0.9619675456389453

Specificity: 0.96214

F1: 0.9602632245001266
```

Conclusions

- Word frequencies even among ubiquitous words can help guide classification even from rudimentary plotting
- There are minor structural differences (wordcount, comma usage), but not with enough differentiation to base decisions on it
- SVC and MNB both yield competitive models, with SVC having the superior F1 score (and specificity and sensitivity)
- Coefficients for the best model (SVC) are not interpretable. No idea what's going on in there. But our word frequency EDA can give us an idea of feature importance.