Classification Of Subreddit Posts

Nicholas Lim
Irwin Wei
Michelle Ng

Outline

- Problem Statement
- Data Information
- Data Cleaning
- Nicholas’ Models
- Irwin’s Models
- Michelle’s Models

Data Cleaning

We generally formatted the data using the following steps:

- Filled posts that only contained images/video clips with blank quotes (‘’)
- Title and content of posts were merged to one single string of text
- Removed non-letters (numerics, new line separators, punctuations)
- Stop words removed using nltk’s library of stopwords
- Lemmatization and stemming were experimented with the data
- Removed keywords found in the subreddit title from posts
- Removed Reddit links from posts using regex

Problem Statement

To classify posts from 2 different subreddits, /r/depression and /r/anxiety.

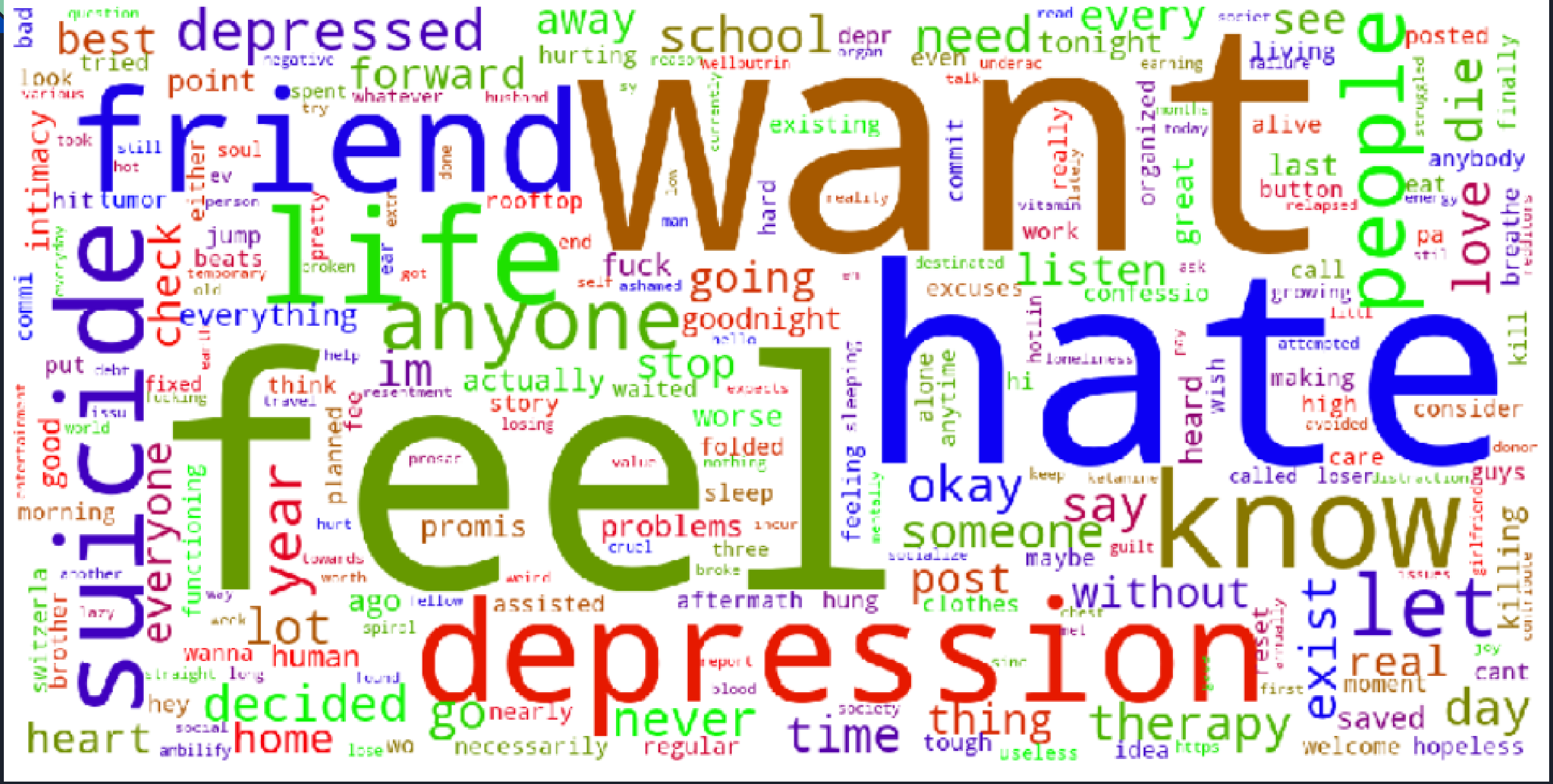
Motivation: Help reddit users who may be suffering from either depression or anxiety issues but not properly diagnosed to get the correct advice through posting at proper channels

About the data

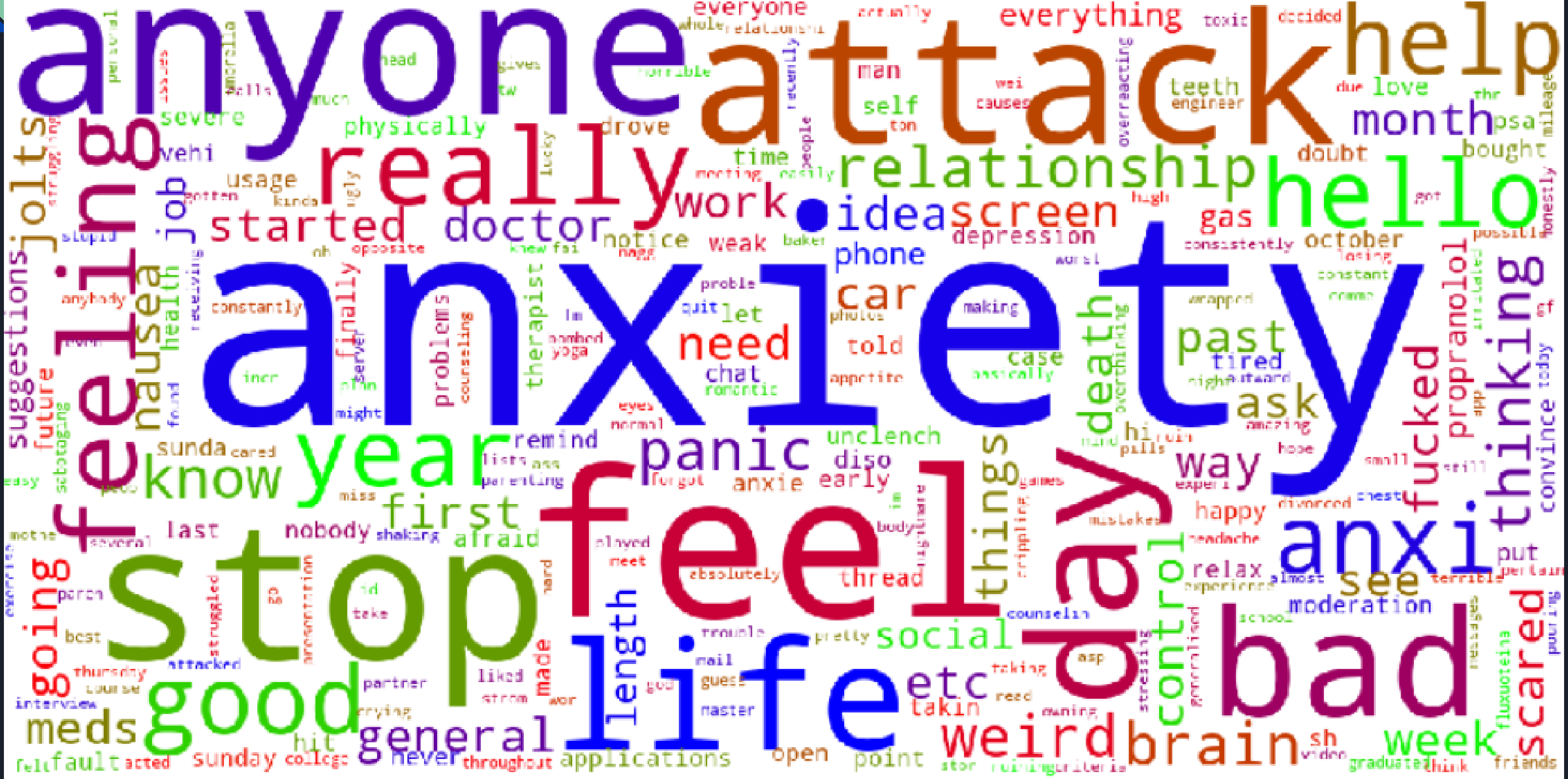
- Total of 1943 rows of data
  - 998 from r/Anxiety
  - 945 from r/depression

Source:
www.reddit.com/r/depression
www.reddit.com/r/anxiety

Frequent Words in r/depression



Frequent Words in r/Anxiety



Modeling

Model	Train Score (CountVectorized)	Test Score (CountVectorized)	Train Score (TfidfVectorized)	Test Score (TfidfVectorized)
Logistic Regression	0.8277	0.821	0.8538	0.8416
K-Nearest Neighbours	0.7076	0.6749	0.7804	0.7737
Naive Bayes (Multinomial)	0.8476	0.8313	0.8428	0.8025
Decision Tree	0.7955	0.7922	0.8016	0.7984
Random Forest	0.8627	0.8374	0.86	0.8477
Extra Tree	0.8298	0.7963	0.8469	0.8004

Model Evaluation

Model	Correct /r/depression posts predicted	Correct /r/anxiety posts predicted
Logistic Regression with TfidfVectorization	197/236	212/250
Random Forest with TfidfVectorization	198/236	214/250

Conclusion

- Random Forest with TfidfVectorizer worked fairly well with an accuracy score of close to 85%, even though both subreddits were fairly similar in nature.
- Logistic Regression with TfidfVectorizer also works equally well as well with an accuracy score of 84%
- Scope can be expanded to include the following to further improve the models:
  - Include lemmatization, stemming and spell checks to have a general feel of the posts
  - Include more subreddits (eg. bipolar) in our classification model. This may be further extended to be used as an initial diagnosis of any mental issues that the user might be suffering from.
  - Tuning of parameters for random forest to get a better score. However, this requires a longer amount of time to tune to get the perfect parameters.
  - Consider either boosting or bagging to get a more optimal outcome.

Irwin: r/TalesFromTheCustomer r/TalesFromYourServer

r/TalesFromTheCustomer:
Accounts of poor customer service encountered by contributors. **997 posts**

r/TalesFromYourServer:
Comprises contributions from people who work(ed) as waiters/waitresses regarding unreasonable customers they encountered at work. **996 posts**

Problem Statement:

To differentiate between both types of posts so that a service provider can obtain insights on pain points experienced by customers and their staff, so as to bolster staff training and psychological preparedness.

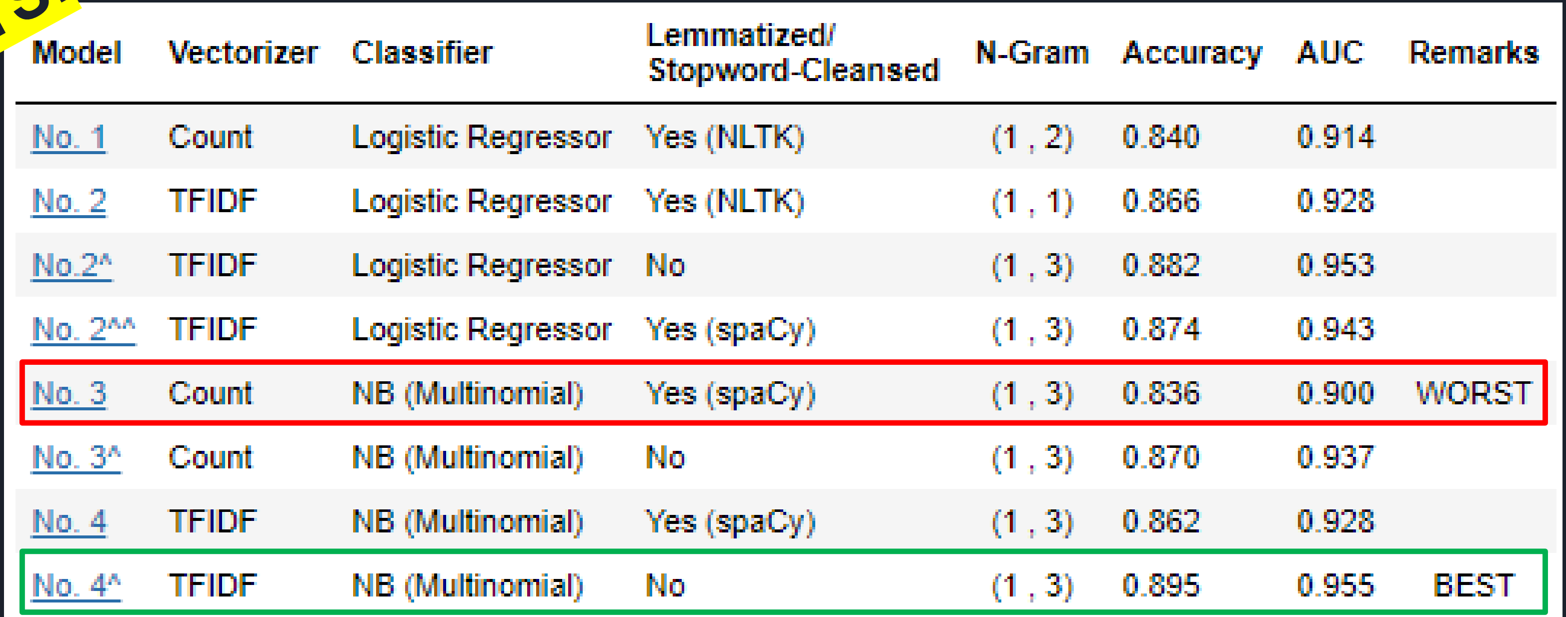
Main Challenge:

Very similar words (nouns, adjectives etc.)

General Approach:

- - Combinations of text pre-processing, vectorization & classifier
- - For each combination, iteratively apply GridSearch to find optimum parameters

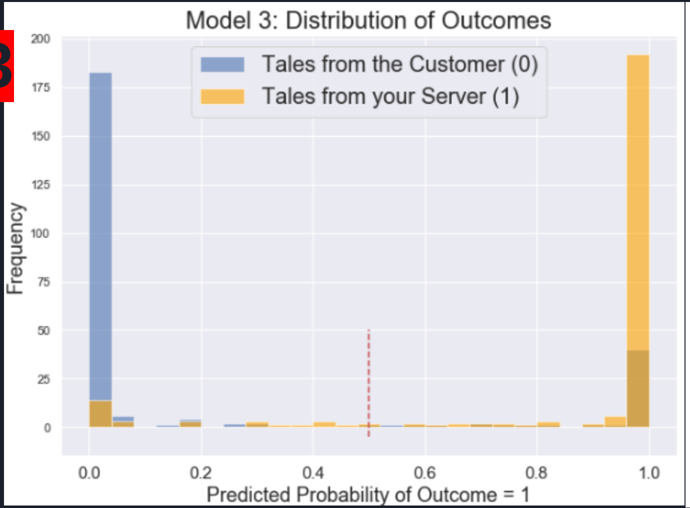
RESULTS!

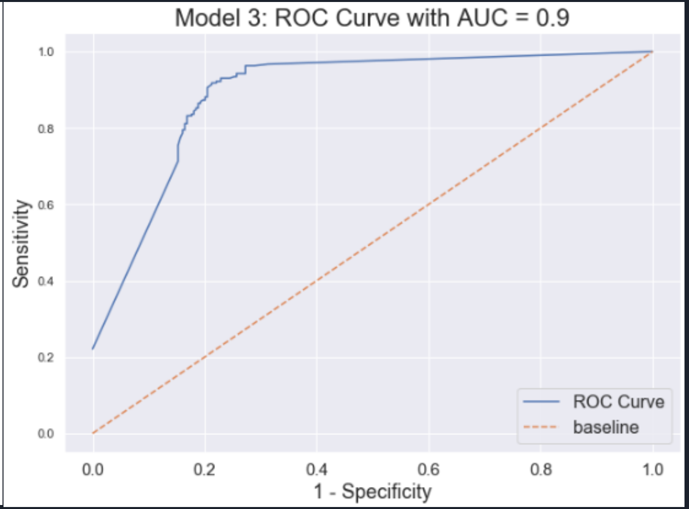


Model	Vectorizer	Classifier	Lemmatized/ Stopword-Cleansed	N-Gram	Accuracy	AUC	Remarks
No. 1	Count	Logistic Regressor	Yes (NLTK)	(1 , 2)	0.840	0.914	
No. 2	TFIDF	Logistic Regressor	Yes (NLTK)	(1 , 1)	0.866	0.928	
No.2^	TFIDF	Logistic Regressor	No	(1 , 3)	0.882	0.953	
No. 2^^	TFIDF	Logistic Regressor	Yes (spaCy)	(1 , 3)	0.874	0.943	
No. 3	Count	NB (Multinomial)	Yes (spaCy)	(1 , 3)	0.836	0.900	WORST
No. 3^	Count	NB (Multinomial)	No	(1 , 3)	0.870	0.937	
No. 4	TFIDF	NB (Multinomial)	Yes (spaCy)	(1 , 3)	0.862	0.928	
No. 4^	TFIDF	NB (Multinomial)	No	(1 , 3)	0.895	0.955	BEST

Model 3

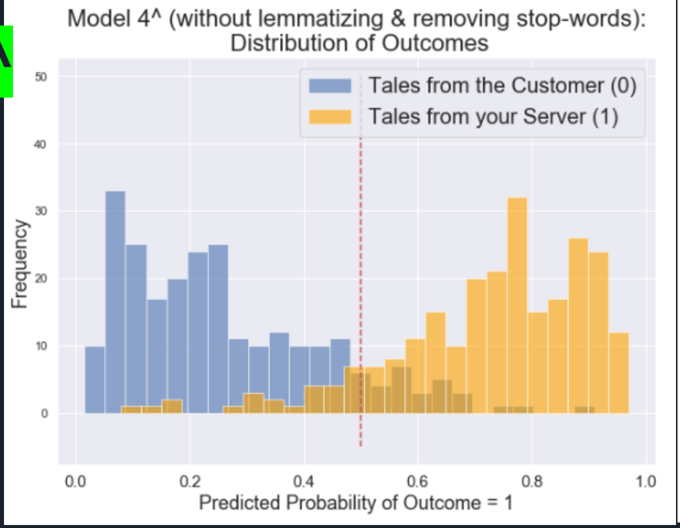
Ngram = (1,3)

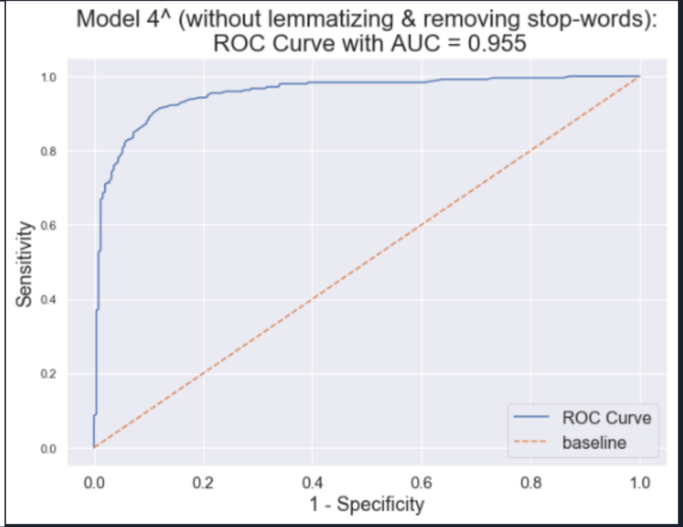

Model 3: Distribution of Outcomes


Model 3: ROC Curve with AUC = 0.9

Model 4^

Ngram = (1,3)


Model 4^ (without lemmatizing & removing stop-words): Distribution of Outcomes


Model 4^ (without lemmatizing & removing stop-words): ROC Curve with AUC = 0.955

Michelle: R/Relationship_advice and R/JUSTNOMIL

Problem Statement:

How can we accurately predict whether a post is from R/relationship_advice or R/JUSTNOMIL?

Importance:

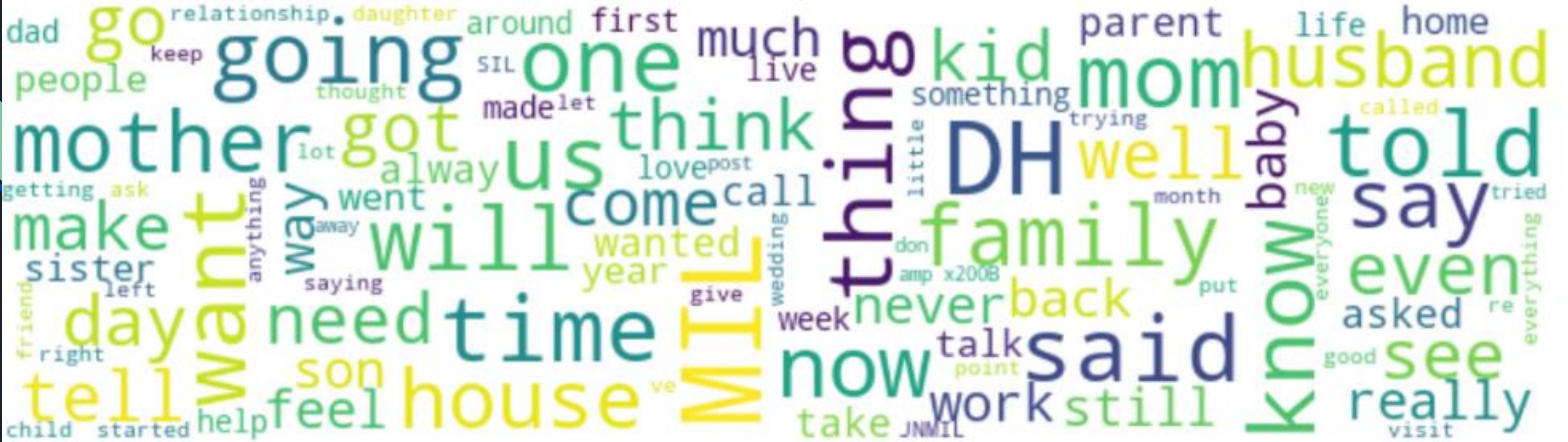
To provide a ‘triage’ for advice as R/JUSTNOMIL may need more time sensitive, immediate responses (emotional support and legal advice).

Data:

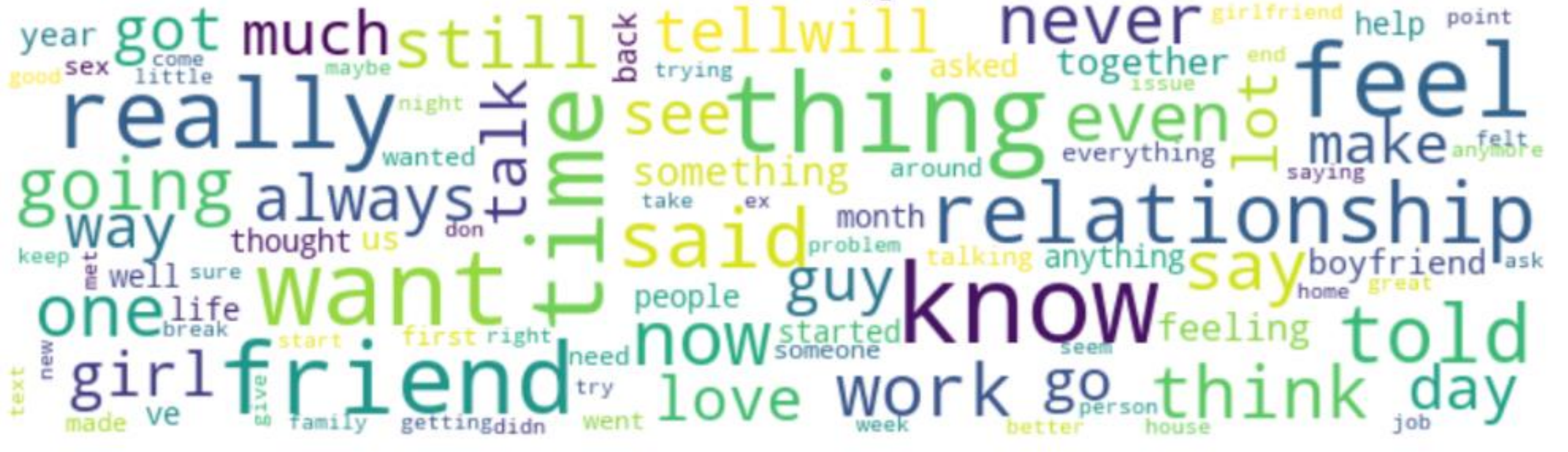
981 Relationship advice

990 JUSTNOMIL

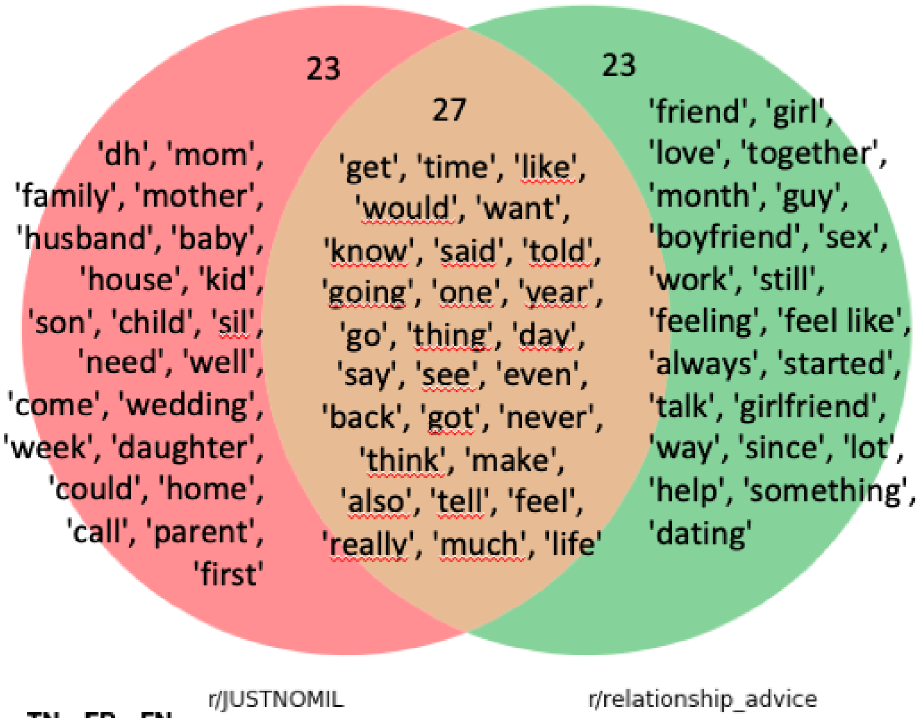
Wordcloud of r/JUSTNOMIL



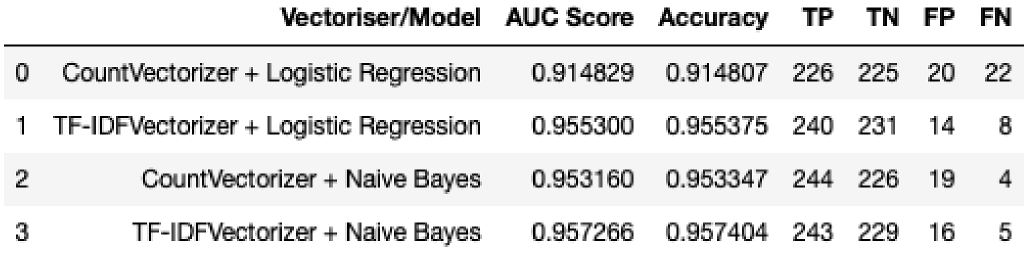
Wordcloud of r/relationship_advice



Venn Diagram of overlapping high probability words from r/JUSTNOMIL and r/relationship_advice (T-Vec)


23
'dh', 'mom', 'family', 'mother', 'husband', 'baby', 'house', 'kid', 'son', 'child', 'sil', 'need', 'well', 'come', 'wedding', 'week', 'daughter', 'could', 'home', 'call', 'parent', 'first'
27
'get', 'time', 'like', 'would', 'want', 'know', 'said', 'told', 'going', 'one', 'year', 'go', 'thing', 'day', 'say', 'see', 'even', 'back', 'got', 'never', 'think', 'make', 'also', 'tell', 'feel', 'really', 'much', 'life'
23
'friend', 'girl', 'love', 'together', 'month', 'guy', 'boyfriend', 'sex', 'work', 'still', 'feeling', 'feel like', 'always', 'started', 'talk', 'girlfriend', 'way', 'since', 'lot', 'help', 'something', 'dating'
r/JUSTNOMIL
r/relationship_advice

Scores and Details of the Models



	Vectoriser/Model	AUC Score	Accuracy	TP	TN	FP	FN
0	CountVectorizer + Logistic Regression	0.914829	0.914807	226	225	20	22
1	TF-IDFVectorizer + Logistic Regression	0.955300	0.955375	240	231	14	8
2	CountVectorizer + Naive Bayes	0.953160	0.953347	244	226	19	4
3	TF-IDFVectorizer + Naive Bayes	0.957266	0.957404	243	229	16	5