Data 410 Project Rough Draft

Daniel Krasnov, Keiran Malott, Ross Cooper

2023-04-07

Contents

Daniel's Analysis (better title later)

Base Model

The dimension of the DTM is too large by default. Can't use bestGLM as too many predictors. Do form of best subset by checking quantities of TF-IDF.

The 95% percentile gave the best AIC so this is the base model I will select.

There are several NAs, not good. Let's get results.

High VIFs, bad need to reduce collineairty. We try PCA now.

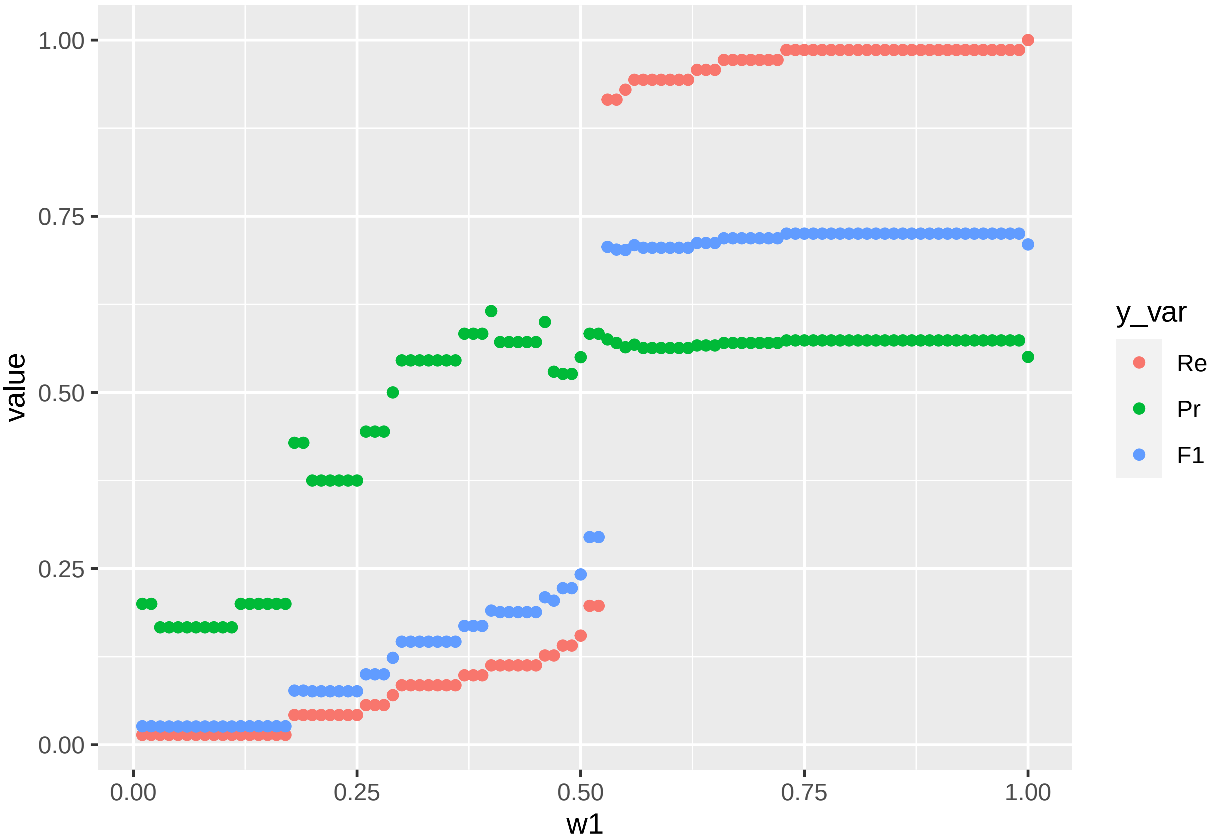
PCA

We keep up to 90th percentile

Very not good predicts everything as not sar.

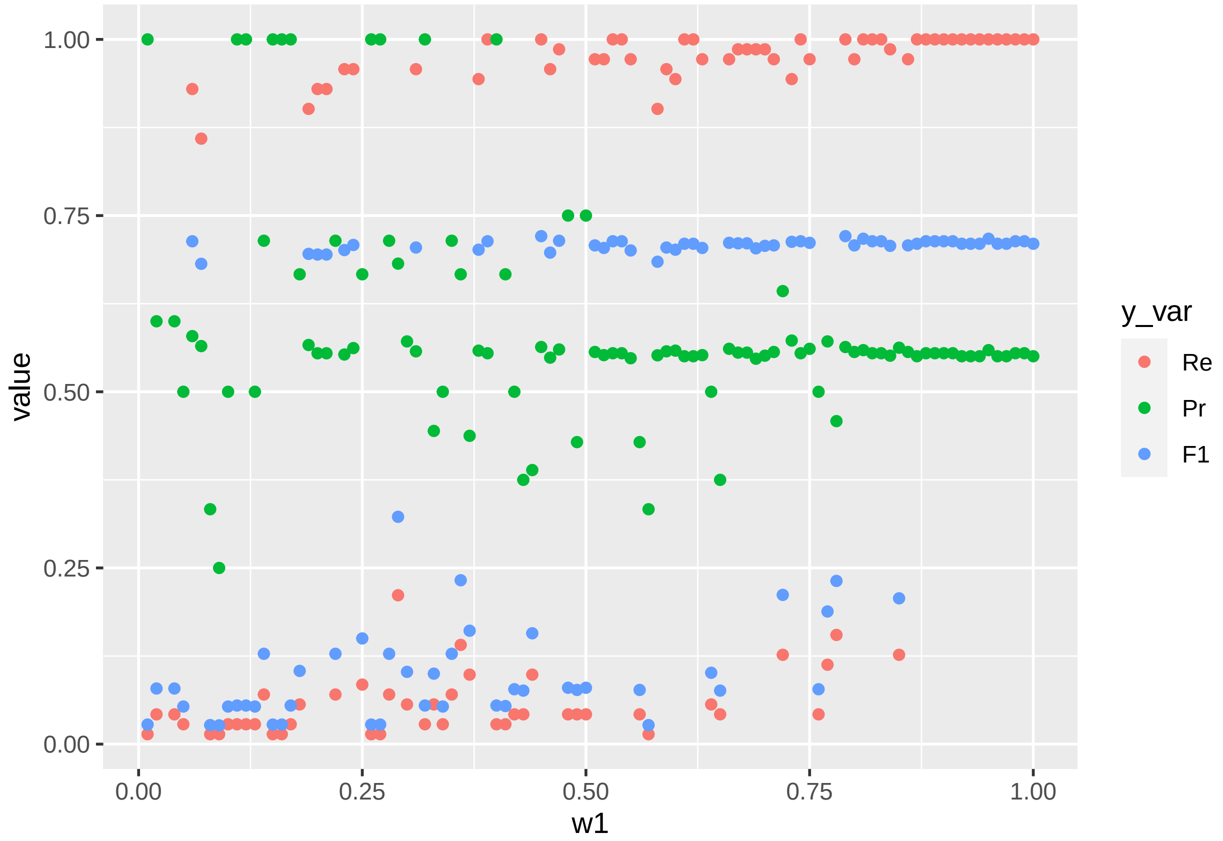
LASSO

Weighted Base Model



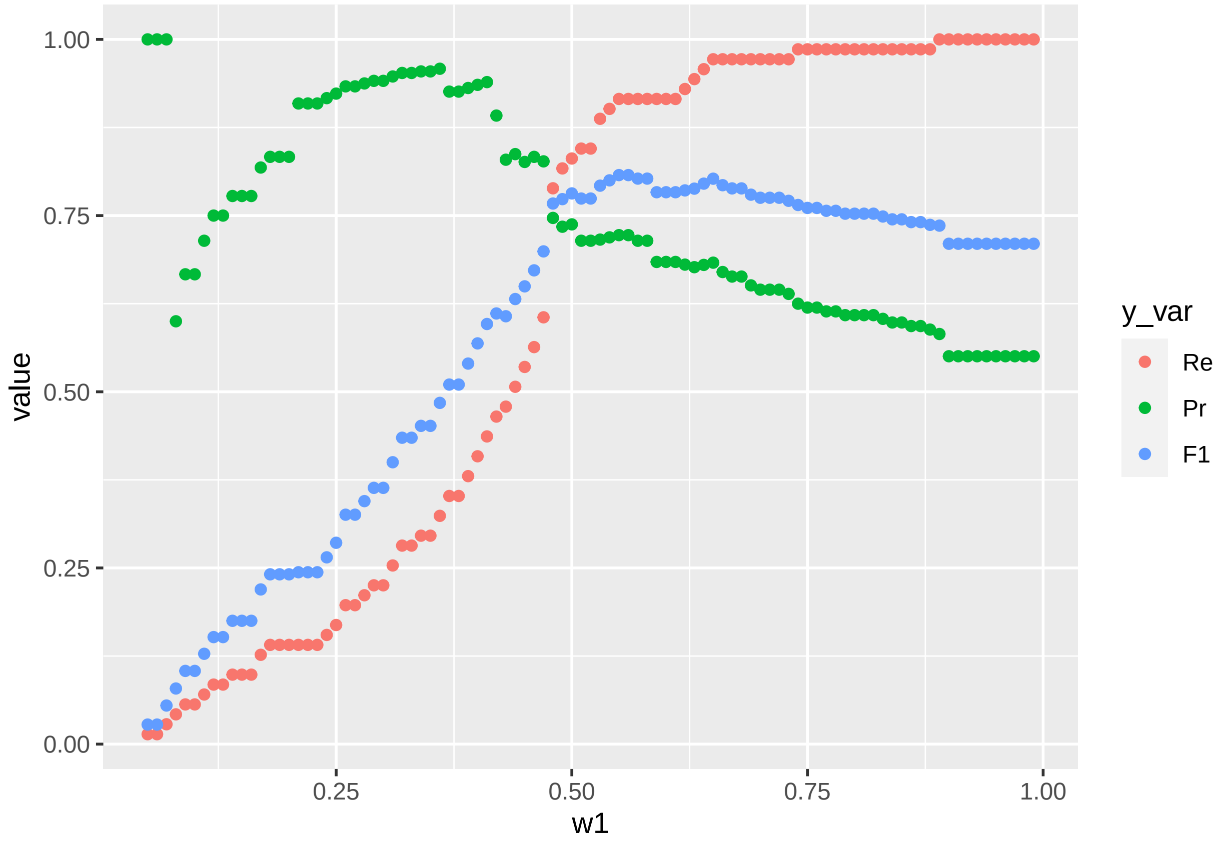
We want a F1 that balances Recall and Precision but still focuses on Recall. Anything around 0.75 is good.

Weighted PCA



F1 holds pretty steady. We go for the one that has the highest Recall, a reasonable precision, and w good F1. 0.62 has

Weighted LASSO



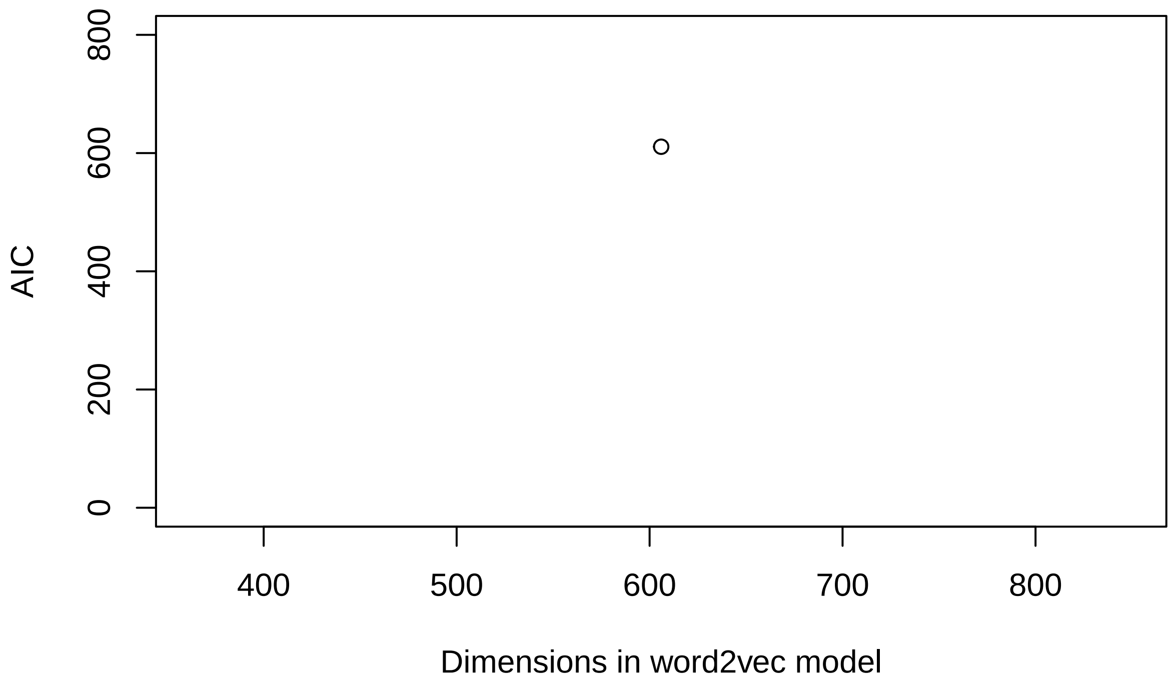
0.55 is best

Keiran's Analysis (better title later)

Ross' Analysis (better title later)

```
## [1] FALSE
```

```
## INFO  [16:00:26.160] epoch 1, loss 0.2108
## INFO  [16:00:26.191] epoch 2, loss 0.0799
## INFO  [16:00:26.201] epoch 3, loss 0.0410
## INFO  [16:00:26.206] epoch 4, loss 0.0225
## INFO  [16:00:26.211] epoch 5, loss 0.0149
## INFO  [16:00:26.216] epoch 6, loss 0.0105
## INFO  [16:00:26.220] epoch 7, loss 0.0077
## INFO  [16:00:26.224] epoch 8, loss 0.0057
## INFO  [16:00:26.228] epoch 9, loss 0.0044
## INFO  [16:00:26.234] epoch 10, loss 0.0034
```



140 dimensions gives the best model for logistic regression

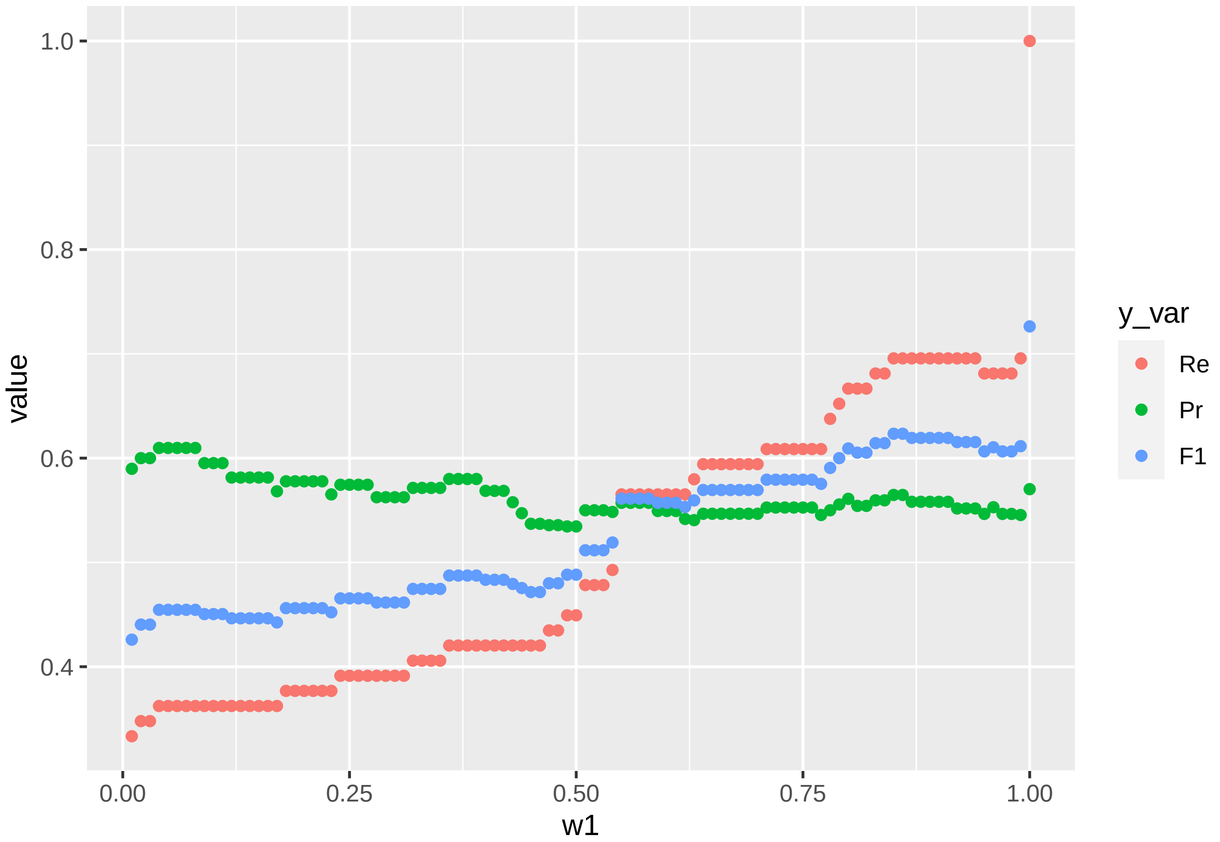
Base Model

```
## INFO  [16:00:28.426] epoch 1, loss 0.2152
## INFO  [16:00:28.431] epoch 2, loss 0.0831
## INFO  [16:00:28.436] epoch 3, loss 0.0427
## INFO  [16:00:28.441] epoch 4, loss 0.0223
## INFO  [16:00:28.446] epoch 5, loss 0.0151
## INFO  [16:00:28.451] epoch 6, loss 0.0107
## INFO  [16:00:28.456] epoch 7, loss 0.0078
## INFO  [16:00:28.462] epoch 8, loss 0.0059
## INFO  [16:00:28.466] epoch 9, loss 0.0045
## INFO  [16:00:28.470] epoch 10, loss 0.0035
```

PCA

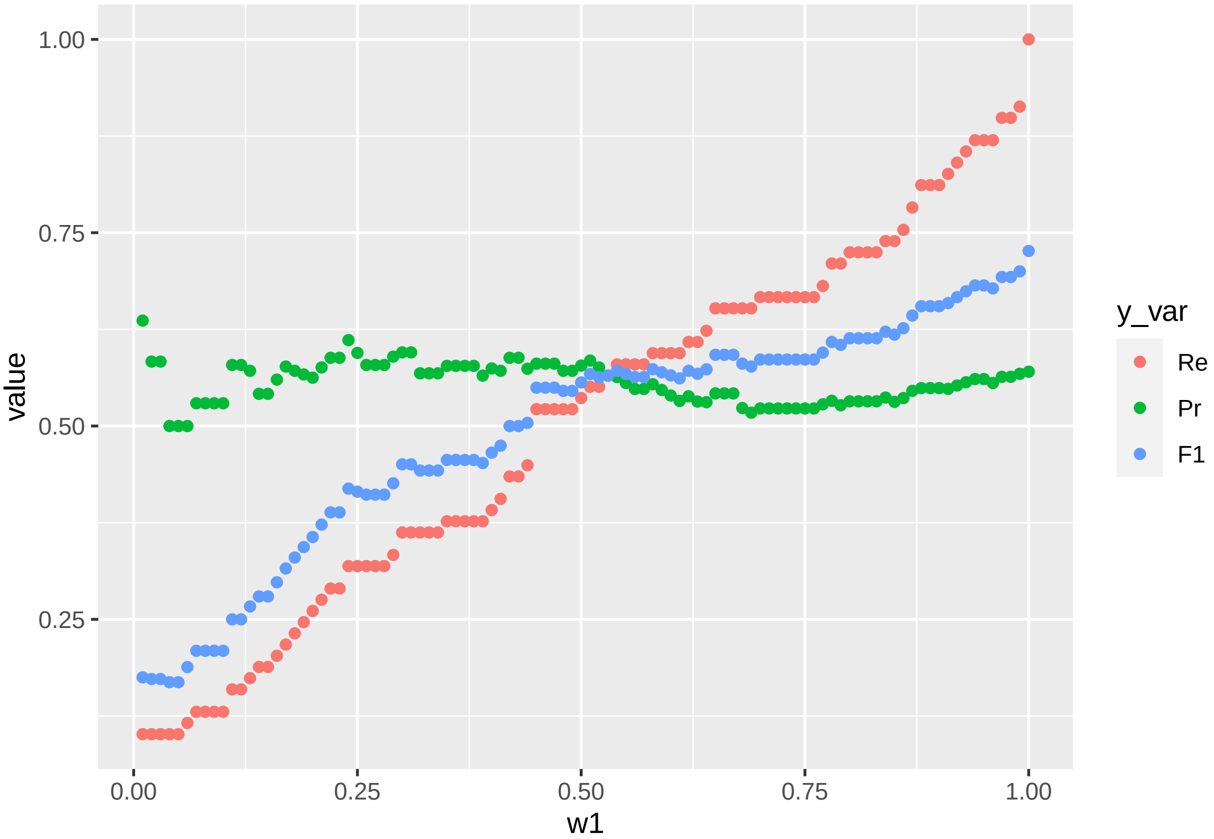
LASSO

Weighted Base Model



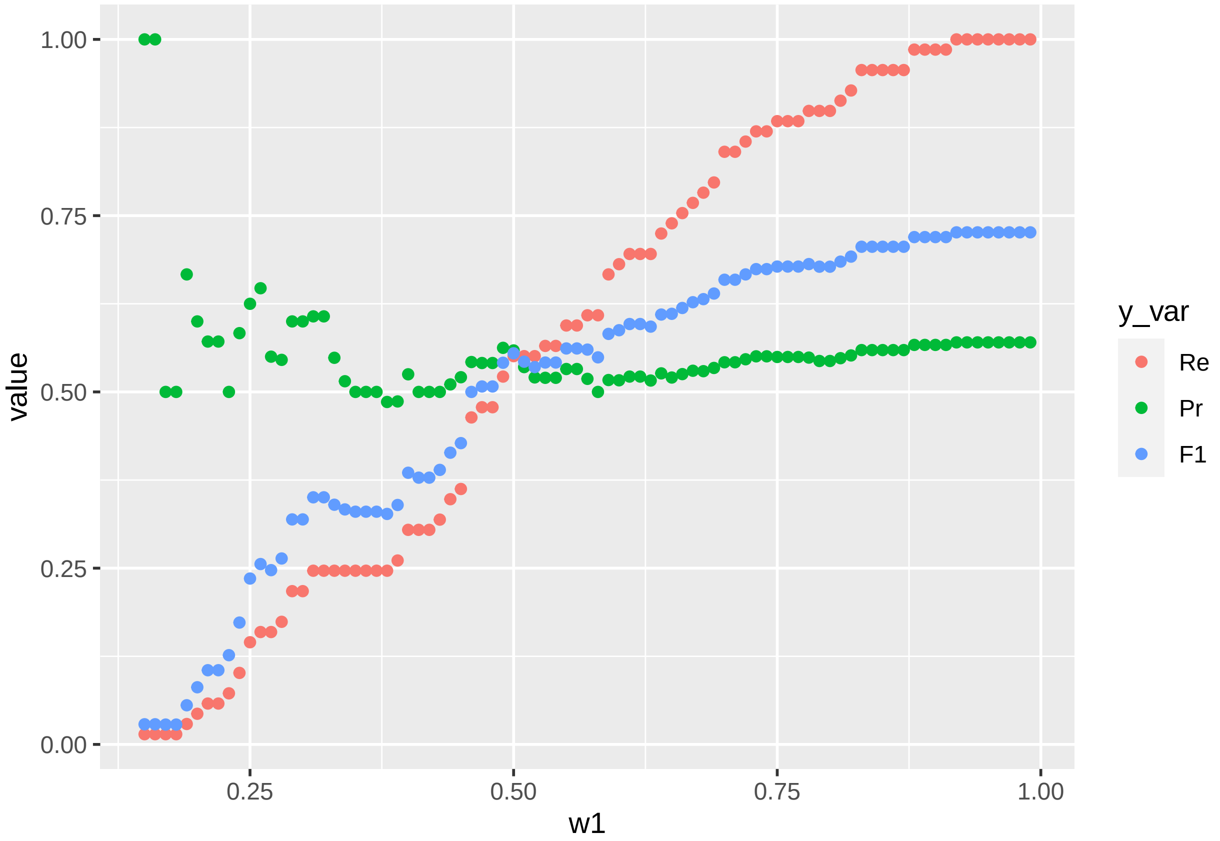
We want a F1 that balances Recall and Precision but still focuses on Recall. Anything around 0.7 is good.

Weighted PCA



Precision holds pretty steady while recall and F1 increase with weighting. We want a weighting with a good balance of all three where the recall plateaus. All scores are good at 0.9, which is also the beginning of the plateau.

Weighted LASSO



Recall and F1 plateau at 0.9

Introduction

Reddit is an American social news website that hosts discussion boards where users can share, comment and vote on various posts (Reddit wikipedia). These posts are housed in subreddits which are communities on Reddit focused on a specific topic.

When writing comments on Reddit users will often write /s at the end of their post to indicate their comment is Sarcastic. This, coupled with Reddit's web scrapping Python API, provides a self labeled data set of sarcatic comments.

The goal our analysis will be to use the /s as a binary indicator of a comment being sarcastic and fit a Logistic regression model. We can then explore this model's efficacy and optimize it for prediction.

Data Collection Method

On the subreddit dataisbeautiful one user posted the following figure (figure citation):

We began by scrapping the top 10,000 posts from each of the above subreddits. We found that all the subbreddits had approximately a 1:100 ratio for sarcastic to non-sarcastic comments. We constructed our first data set by sampling from all the above subbreddits however, we found the data to be too 0 heavy and no model specification could learn an underlying relationship between words and sarcasm. We then attempted to fit models to various ratios of sarcastic to non-sarcastic comments. We found that Logistic regression began to perform reasonably well at a ratio of 1:2 sarcastic to non-sarcastic. We also found that models tended to perform far better if all comments came from a single subreddit as opposed to multiple. As per

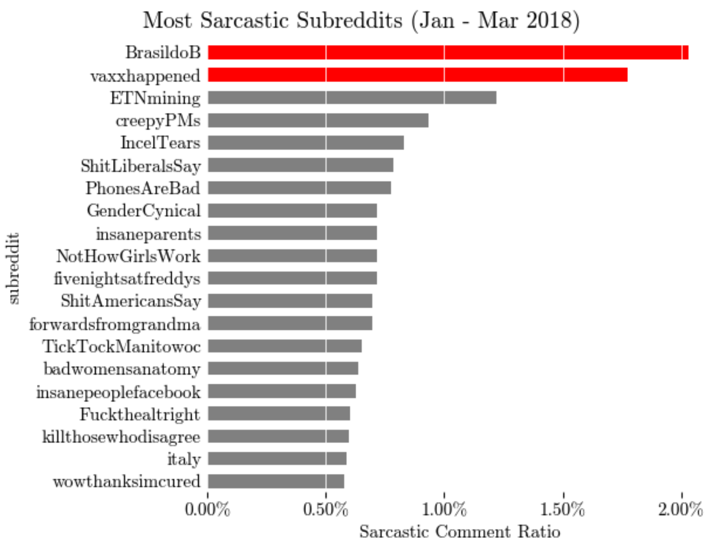
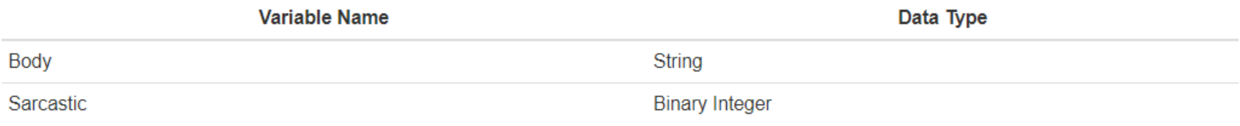


Figure 1: sarcastic_subreddits

our prelimnary results we opted for a single subreddit at a ratio of 1:2 sarcastic to non sarcastic comments. NotHowGirlsWork was found to have the largest count of Sarcastic comments at 321 therefore we selected this subreddit for our data set.

Variable Description

Our data set is constructed as follows:



Variable Name	Data Type
Body	String
Sarcastic	Binary Integer

Figure 2: data set table

where Body is the raw comment string scrapped from the comment and Sarcastic is 1 when /s is present in the comment and 0 when it is not.

Data Preprocessing

In Natural Language Processing there are various text preprocessing steps that are common to employ (text as data citation):

- Punctuation, whitespace, and number removal - any punctuation characters such as !, @, #, etc. as well as empty space and numbers are removed.
- Stopword Removal - removal of words that fail to provide much contextual information, e.g., articles such as ‘a’ or ‘the’.
- Stemming - identifying roots in tokens and truncating words to their root, e.g., fishing and fisher transformed to fish.

In our data set we first removed the /s from every sarcastic comment and preforemd the above preprocessing steps.

Feature Extraction Methods

In order to use text as data in a Logistic regression we must numerically encode our strings. There are a plethora of feature extraction methods in NLP. For our analysis we compare TF-IDF, Word2Vec, and GloVe.

TF-IDF

Term frequency inverse document frequency (TFIDF) is a heuristic to identify term importance (text mining in R citation). It calculate the frequency with which a term appears and adjusts it for its rarity. Rare terms are given increased values and common terms are given decreased values (text as data citation).

TFIDF is given by

$$\text{TFIDF}(t) = \text{TF}(t) \times \text{IDF}(t)$$

where

$$\text{TF}(t) = \frac{\text{\# of times term t appears in a document}}{\text{\# of terms in the document}}$$

and

$$\text{IDF}(t) = \ln\left(\frac{\text{\# total number of documents}}{\text{\# number of documents where t appears}}\right)$$

In our analysis a document is a Reddit comment. After being preprocessed the text of each comment is separated into individual terms and has its TFIDF calculated. From there the TFIDF values are placed in a Document Term Matrix (DTM). This matrix has rows as the documents and each column in a term. It is therefore a sparse matrix where entries are the TFIDF score for a specified term in a specified document.

The DTM acts as the design matrix for our Logistic Regression model: **TO DO: ask if john wants this level of explanantion**

```
## <<DocumentTermMatrix (documents: 6, terms: 8)>>
## Non-/sparse entries: 1/47
## Sparsity           : 98%
## Maximal term length: 10
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
## Sample             :
##     Terms
## Docs common forevaaaaa husband lost potenti surviv two      will
##   10      0          0       0    0       0      0   0 0.0000000
##   5       0          0       0    0       0      0   0 0.0000000
##   6       0          0       0    0       0      0   0 0.2352558
##   7       0          0       0    0       0      0   0 0.0000000
##   8       0          0       0    0       0      0   0 0.0000000
##   9       0          0       0    0       0      0   0 0.0000000
```

Word2Vec

Explain word2vec. Show what final result is for design matrix and explain how you got there.

Glove

Explain Glove. Show what final result is for design matrix and explain how you got there.

Evaluation Metrics

Describe Precision, Recall, F1.

Regression Analysis

This analysis is a comparison of logistic model performance when using 3 types of feature extraction.

Variable Selection

TF-IDF Selection

Word2Vec Selectopm

Glove Selection

Metric Evaluation

TF-IDF Evaluation

Word2Vec Evaluation

Glove Evaluation

Violations

TF-IDF Violations

Word2Vec Violations

Glove Violations

Conclusion

Model Comparison

Limitations

Final Remarks

References

https://en.wikipedia.org/wiki/Reddit#References

https://www.reddit.com/r/dataisbeautiful/comments/9q7meu/most_sarcastic_subreddits_oc/

Text as Data Barry DeVille, Gurpreet Singh Bawa