Data 410 Project Rough Draft

Daniel Krasnov, Keiran Malott, Ross Cooper

2023-04-07

Contents

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. We will fit increments of 5% for quantiles and see which produces best AIC

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.

We keep up to 90th percentile

Very not good predicts everything as not sar.

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

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

0.55 is best

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 sarcastic 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 using various feature extraction methods. 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 subreddits had approximately a 1:100 ratio for sarcastic to non-sarcastic comments. We constructed our first data set by sampling from all the above subreddits 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 1:1 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 our prelimnary results we opted for a single subreddit at a ratio of 1:1 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:

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

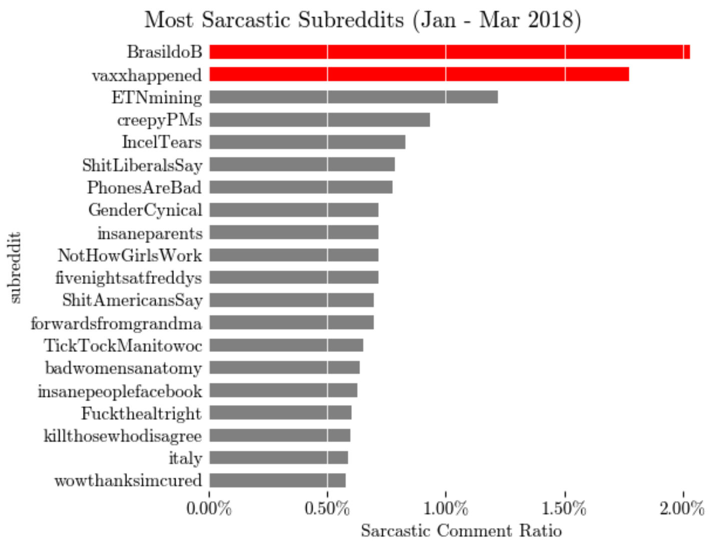
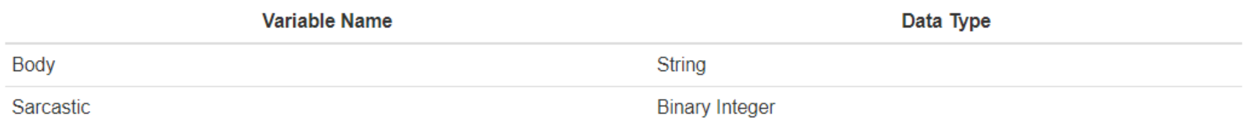


Figure 1: sarcastic_subreddits



Variable Name	Data Type
Body	String
Sarcastic	Binary Integer

Figure 2: data set table

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*, individual words, and truncating them 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 preformed 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 tokens and has its TFIDF calculated. From there the TFIDF values are placed in a *Document Term Matrix* (DTM). This matrix has document ids as rows and tokens as columns. It is therefore a sparse matrix where entries are the TFIDF scores for corresponding tokens.

The DTM acts as the design matrix for our Logistic Regression model:

```
## <<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

Word2Vec is a group of predictive models for learning vector representations of words from raw text. Word2Vec uses either the *continuous Bag-of-Words architecture* (CBOW) or the *continuous Skip-Gram architecture* (Skip-Gram) to compute the continuous vector representation of words. Both CBOW and Skip-Gram use shallow neural networks to achieve this, but CBOW predicts words based on the context and Skip-Gram predicts surrounding words given the current word (Efficient Estimation of Word Representations in Vector Space paper citation).

Each word is represented as a vector, and words that share common context are close together in vector space (Deep Learning Essentials textbook citation). Document vectors are representations of documents (Reddit comments) in vector space. A document vector can be constructed by summing the the word vectors from a common document and then standardizing them (word2Vec package citation). The design matrix for logistic regression can be constructed with the rows of the matrix as the document vectors. The resulting design matrix therefore has one row per Reddit comment and is as follows:

GloVe

Global vectors for word representation (GloVe) is an unsupervised learning algorithm which creates a vector representation for words by aggregating word co-occurrences from a corpus. The resulting co-occurrence matrix X contains elements $X_i j$ representing how often word i appears in the context of word j (citation).

Next, soft constraints for each word pair are defined by:

$$w_i^T w_j + b_i + b_j = log(X_{ij})$$

where w_i is the vector for the main word, w_j is the vector for the context word j, and b_i and b_j are scalar biases for the main and context words. Finally, a cost function is defined:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$$

Here f is a weighting function chosen by the GloVe authors to prevent solely learning on extremely common word pairs (citation):

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{x_m ax}\right) \alpha & \text{if } X_{ij} < XMAX \\ 1 & \text{otherwise} \end{cases}$$

To create the design matrix below, a vocabulary of the words in the corpus was created. Since this method creates a co-occurrence matrix, we prune all words which appear less than five times to reduce bias from less common words (citation). From there we constructed a term-co-occurrence matrix and factorized it via the GloVe algorithm. The resulting matrix consists of word vectors as rows, which are added together to create sentence vectors that are used to train the model:

Evaluation Metrics

Model performance is assessed based on classification performance. In sentiment analysis the most common metrics to tune model for performance are Precision, Recall, and F1 Score (citation).

Precision is the number of true positive divided by the number of true and false positives. Recall is the number of true positive divided by false negatives and true positive. It is the true positive rate. F1 Score is the harmonic mean of Recall and Precision (python learning citation) (add a CM and write the formulas).

For our analysis a true positive is a correctly predicting a comment is sarcastic.

Regression Analysis

This analysis is a comparison of logistic regression model performance when using 3 types of feature extraction. For each feature extraction we fit a base model. We then perform Princial Componenet Analysis (PCA) to reduced dimensionality and deal with multicollinearity. Finally we investigate LASSO models a means for dimensionality reduction.

After model fitting we perform weighting on all 3 model types and decide a best model for each feature extraction method. Weighting in done by multiply each predictor by w, where $w \in (0, 1)$, if a comment is sarcastic and by $1 - w$ if a comment is not sarcastic. This is done for every w starting from 0.01 to 0.99 in increments of 0.01. We record testing and training metrics foe each model and discuss our optimal selection.

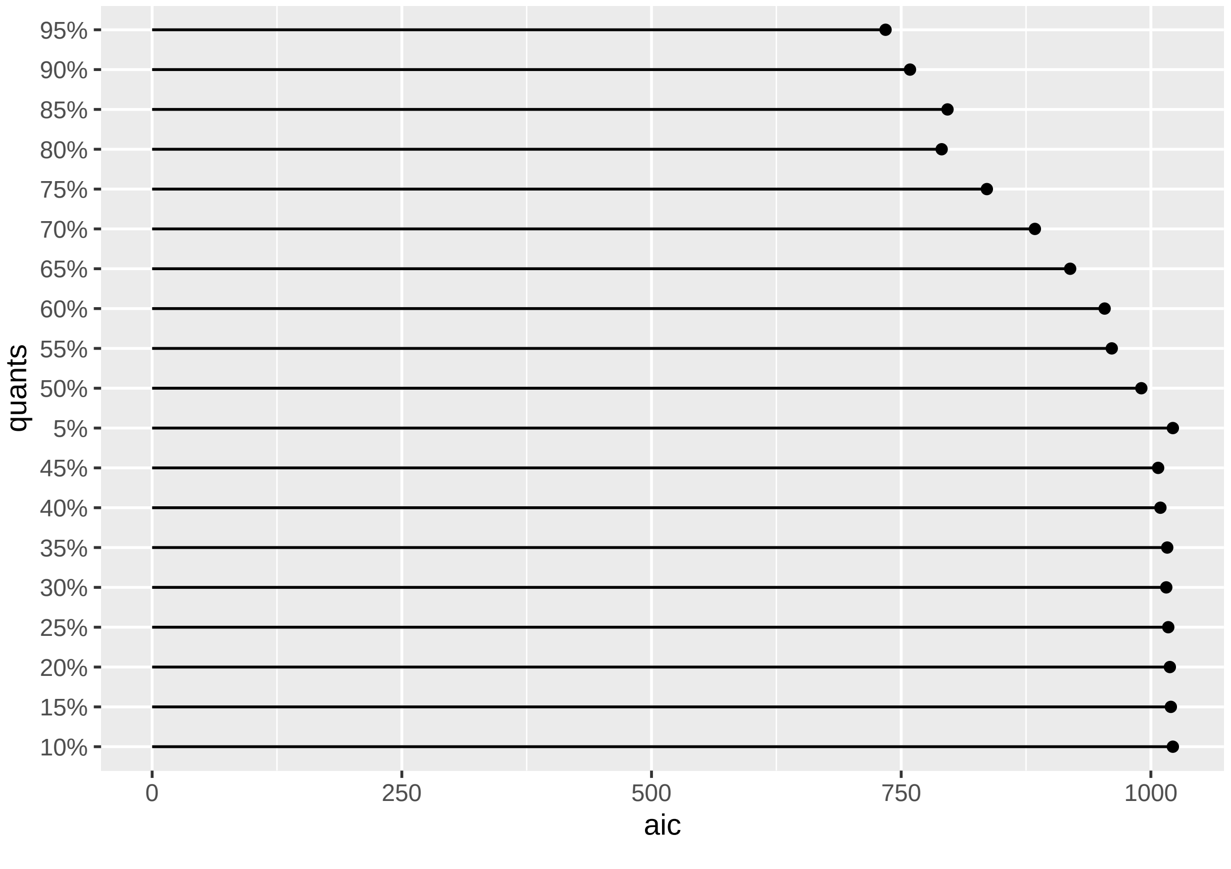
We hypothesis that models using GloVe as the feature extraction method will perform similarity to Word2Vec. TF-IDF will perform the worst. TF-IDF numerically encodes text based on rarity. We think this is too simple an approach to capture important sarcastic words. Word2Vec captures context and GloVe interprets word co-occurrences which we believe could both be suitable strategies to capture sarcastic structure in text.

Model Specification

TF-IDF Selection

Variable Selection

Base model The initial DTM after text processing was 642×2074. This matrix has far too many columns compared to rows and so some dimensionality reduction was required. One way to do so is to filter away unimportant terms. This can be decided by the quantiles of the TF-IDF. We can exclude any columns of the DTM based on whether or not their values fall within a certain percentile. If we do this for all percentile between 1 to 100 in steps of 5 we can record the AIC and choose the best subsetted model.



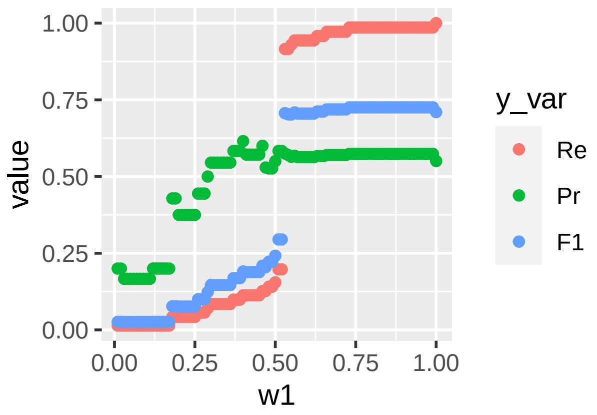
The minimum AIC found was 734 for a model not including any terms who had TF-IDF values below the 95th percentile. This resulted in a 14% reduction in deviance. Thew new dimensions of the DTM were 512×74. ##### PCA model

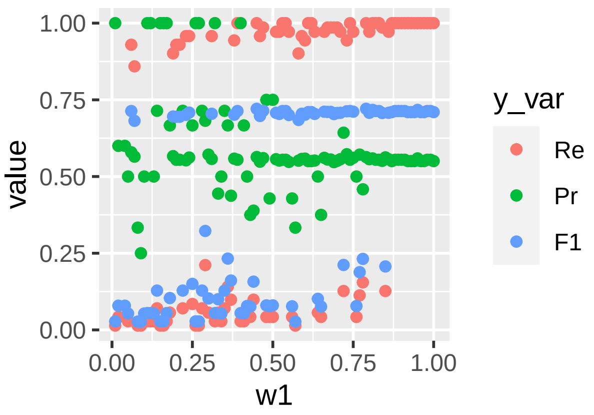
In an attempted to get around the multicollinearity in the model PCA was utilized on the filtered DTM. 90th cumulative proportion was kept. This resulted in an AIC of 17135 and actually increased the null deviance by a factor of 24 as opposed to reducing it.

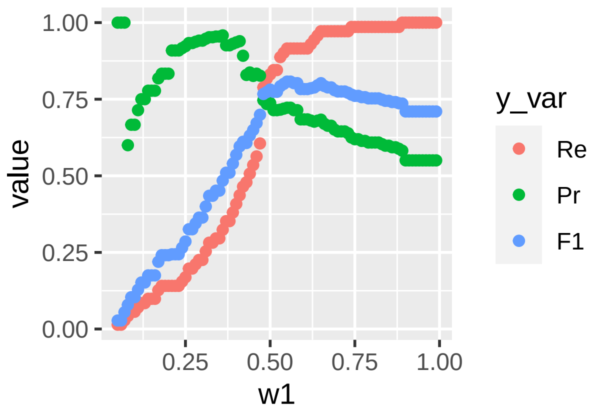
LASSO model corss validation was sued to fit a LASSO with a best lambda of 0.02539292. List of signifigant words was: tate,written, gold, final, companionship, realis, reduct, asexu, remov, sister, page, pictur, puberti, gal, act, crime, cop, polic, pedo, jail, report, dump, broke, depend, nowaday, iron, ignor

null deviance reduction: 0.3455114 null deviance of 710.8395

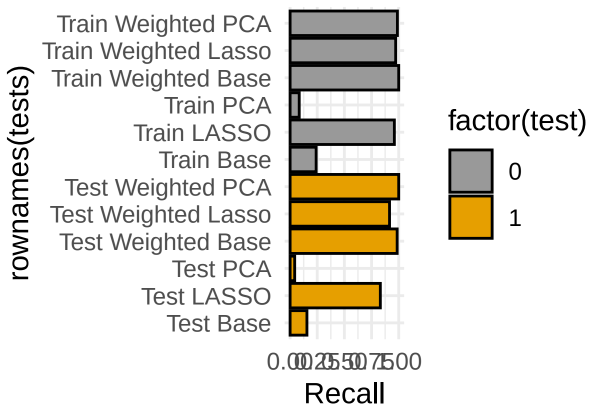
Weightings Weightings for each model were chosen as follows:

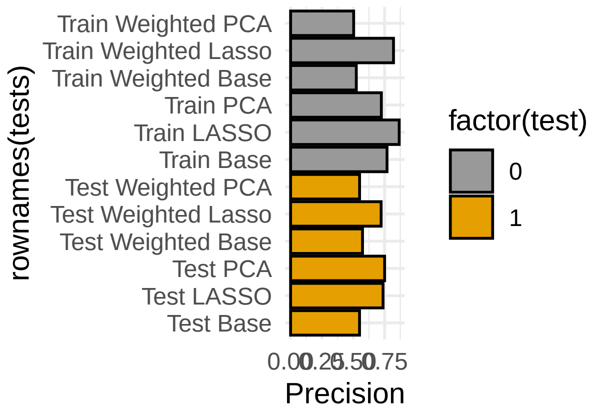


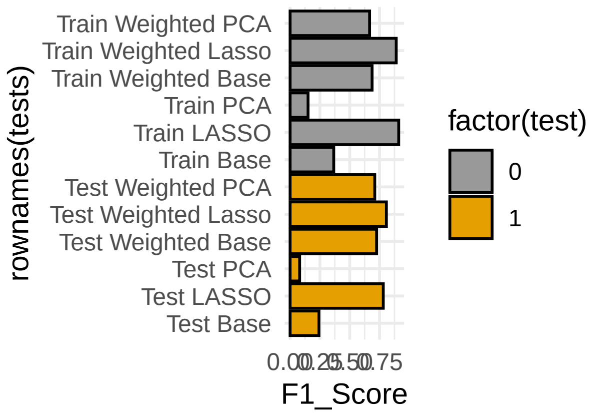




TF-IDF Evaluation





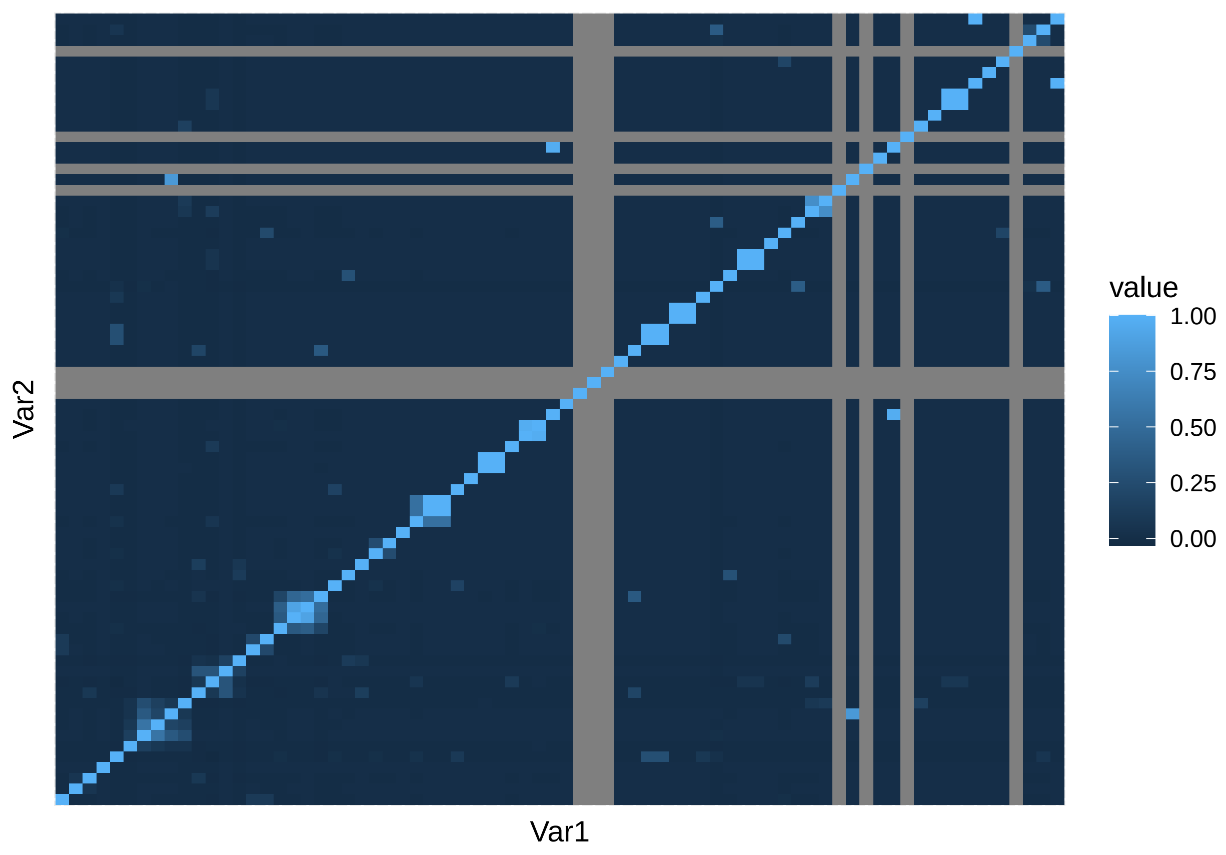


Diagnostics

TF-IDF Diagnostics

Base Serious multicollinearity issues. Many eastimates are NA. 6 VIFs were found to be over 10 indicating serious multicollinearity in the model and several of the coefficients for temrs in in the model were unable to be estiamted and assigned a NA

```
## Warning in cor(basemodelforcormap[, -1]): the standard deviation is zero
```



PCA PCA removed all collinearity no vifs. However deviance increased. This indicates potential information loss fr

LASSO

Base Model

PCA model

LASSO model

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

This paper shows we can use these metrics for this: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9320958&tag=1

for definition of recall, precision, and f1 use: Hands-On Ensemble Learning with Python