Write-Up Notes

Description of the data

- The training data and test data both have 10,000 observations. There are more comments labeled "toxic" than labeled "obscene" Note that categories are not mutually exclusive and that comments can be labelled both obscene ***and*** toxic.
- There's almost a perfect 50/50 split between observations who were labeled either "toxic" or "obscene" and those who were not labeled as either.
- 99% of the comments labeled "obscene" were also labeled "toxic". Of the 5018 comments labeled "toxic", approximately 51% of them were also labeled "obscene".
- ADD SOMETHING ABOUT WORDCLOUD
- KMEANS MIGHT NOT BE GOOD CHOICE BC DATA SEEMS TO BE SPARSE IN THE TOXIC / OBSCENE SPACE – LEFT WITH ONE DENSE CLUSTER WHERE A LOT OF COMMENTS ARE LABELED TOXIC OR TOXIC & OBSCENE
- CREATE WORDCLOUD FOR WORDS WITH HIGHEST COEFF VALUES FOR TOXIC AND OBSCENE

Starting point for modeling

- The first model I created was a ridge regression model using un-pre-processed data.
- This model serves as a sort of benchmark for my future attempts – I expect that a "good" model will outperform this one.
- As such, I then had two major considerations as I developed "better" models: pre-processing choices and model selection.

Features: Pre-processing choices

- The texts classified as either toxic or obscene seemed, while browsing the data, to include coarse language (i.e. words like "fuck", "shit", "ass") and also used capital letters and punctuation liberally (e.g.)
- I decided to start by creating models that from texts that had only a single type of pre-processing performed at a time, taking advantage of the "control" settings offered by `tm`. I expected that removing whitespace, stopwords, and numbers would help, but was unsure if completely removing punctuation would be helpful since it seemed to be common in the target comments. However, its widespread use in non-target comments (i.e. comments not labeled either "toxic" or "obscene") could also imply that punctuation, broadly speaking, would not consitute informative features.
- I also tried four different weighting approaches with no other pre-processing: tf, tf-idf, SMART weighting, and bin weighting.
- When tested individually (i.e. one pre-processing decision applied at a time), the following pre-processing and/or weighting choices yielded the lowest CV error in the training data: - *For "toxic" classification: numbers removed, text tokenized, stopwords removed* - For "obscene" classification: text stemmed and punctuation removed

- Once I identified the top 3 single processing choices for each classifier (as measured by CV error of the resulting ridge regression model), I created two models for each classifier based on the best weighting methods. I then tried permutations of the different pre-processing choices for each weighting method and select the 3 models (per classifier) with the lowest CV error for further investigation.
- The following pre-processing/weighting choices yielded the lowest CV error for the training data:

  For "toxic" classification:
  - ZZ - XX - YY

  For "obscene" classification:
  - ZZ - XX - YY
- After determining the top three models based on standard `tm` package options, I went on to build my own features and include them in the models.

Custom features

- Reading through some comments, it becomes obvious that all of the ones marked either toxic or obscene contain curse words. As such, I wanted to explore ways of making those words classification criteria.
- remove terms that occur in fewer than 5 documents dtm <- DocumentTermMatrix(corpus, control = list(bounds = list(global = c(5, 3000))))
- Proportion of capital letters: Given that both toxic and obscene classifications denote aggression, I decided to use the proportion of capital letters as a proxy of indicator of aggression and incorporate it into my best GLM model.
- Number of exclamation or question marks: Another proxy for aggression could potentially be reflected in the use of question of exclamation marks. Usually, many question marks in a row indicate a rhetorical or sarcastic attitude (e.g. "Oh really??????", "You are an idiot!!!!!") which could be employed in an otherwise obscene or toxic comment to emphasize the aggressive undertone that the writer is trying to convey.

Model Selection

- I started modeling using a relatively un-complex model: ridge regression. This alone resulted in good predictive ability, with models hovering around a score of 0.88 in Moodle
- After running exploratory models (with different pre-processing options as described above), I transition to a lasso model given the theoretical advantage of a smaller feature space and, thus, reducing the risk of overfitting (though simultaneously reducing model interpretability).
- Since our objective in this exercise is predictive rather than related to inference, I felt comfortable making the decision to sacrifice model interpretability in this case.
- Random forest
- Bagging
- Boosting
- Look at coeffs for each model

Ridge Regression

- I started my analysis using **ridge regression** – a regularized form of the general linear model. Building a simple model gave me a computationally efficient way to start my model selection.
- Using a ridge regression model, I tried different pre-processing choices to identify those that yielded models with the lowest CV error. I then calculated the test error and F1 score for the best three models for each classifier (i.e. six models total).

Lasso

- I then decided to use the same top 3 pre-processing combinations in **lasso** models since a lasso model would further shrink the feature space.

Estimating generalization error

- I used F1 to estimate the generalization error. I calculated the F1 score on a test set built from the training data provided.
- NEED TO DO: Discuss your choice with respect to bias/variance of estimates of generalization error, computational cost, and the number of feature selection and model selection choices at your disposal.
-

Concerns re: overfitting

Submissions

Misleading Submissions

Please note that my first four submissions (below) were created using training data that was then bound to test data identifiers (thus explaining their poor performance). They are essentially nonsensical and resulted in poor model performance (~60%). * 1 - Model: Ridge regression, Preprocessing: None, Result: Labeled all as toxic, none as obscene * 2 - Model: Ridge regression, Preprocessing: only removed punct (used this model bc it seemed to give best F1 results across toxic and obscene) * 3 - Model: Ridge regression, Preprocessing: "toxic" classification - number removal, text tokenization, and stopword removal; "obscene" classification - stemming, punctuation removal, and text tokenization. * 4 - Model: Ridge regression, Preprocessing: "toxic" classification - remove numbers; "obscene" classification - stemming, tokenize

Real Submissions

The below submissions correctly used test data. * 5 - Model: Ridge regression, Preprocessing: "obscene" - bin weighting, remove punctuation, remove whitespace, tokenization; "toxic" - whitespace, punct, token, bin weighting

Cool resources:

https://developers.google.com/machine-learning/guides/text-classification/step-2-5 * Suggests doing n-gram vectors * Suggests that accuracy peaks at 20k terms – anything more than that, tend to overfit * Discard terms that occur less than 2 times * Use f-classif to determine feature importance? (DIDN'T DO THIS IN CLASS?) * Feature reduction here means getting rid of uninformative words (can reduce using tf-idf) * Suggests to look @ log reg or SVM or MLP (multi-layer perceptrons – but we didn't learn about this in class)

Instructions

- A description of the data using plots, tables, and your written observations of any patterns you have uncovered.
- An outline of potential features that you may use in the analysis and your thoughts about their respective merits for each classification problem.
- A description of custom features you have constructed for this classification problem (e.g. comment length, count of consecutive capital letters, etc.) and why you think they might be informative.
- A writeup of any classification, dimension reduction, clustering, and feature learning algorithms you will use and why you expect them to perform well.
- A write up of your strategy for estimating generalization error (F1 in this case). Discuss your choice with respect to bias/variance of estimates of generalization error, computational cost, and the number of feature selection and model selection choices at your disposal.

- A discussion of your strategy for model selection and tables recording CV error for each approach and test error for models submitted to Kaggle
- Code should be included in a separate Rmd file (or any other file type if you wish to use a language other than R).