Modeling Hate Speech

Nikky Yu

11/27/2020

Section1: Introduction

With nowadays rapid development of the Internet and a strong support for free speech, negative online behaviors, such as toxic comments, appear frequently. The threat from these rude and disrespectful comments may be huge, making people feel depressed and be afraid of expressing themselves. Especially for children and teenagers who have not developed settled personalities, such threat might mislead their lives to the wrong way. Therefore, I feel that it is important to create a tool to help to clean nasty comments and improve online conversations. Once we create a model to identify nasty comments with high accuracy, these comments can then be filtered by programmers in order to protect the whole general group's mental and physical health. The data set we are using in this project is taken from Kaggle competition. All comments in this data set come from Wikipedia talk page edits and they have been labeled by human raters for toxic behavior. Please note that some of the words contained in this paper may be offensive. There are a total of 159571 observations in this data set and observations are divided into six types of toxicity: toxic, severe toxic, obscene, threat, insult, and identity hate. Our goal in this project is to create a model that is capable of detecting different types of toxicity.

Section2: Data Cleaning and EDA

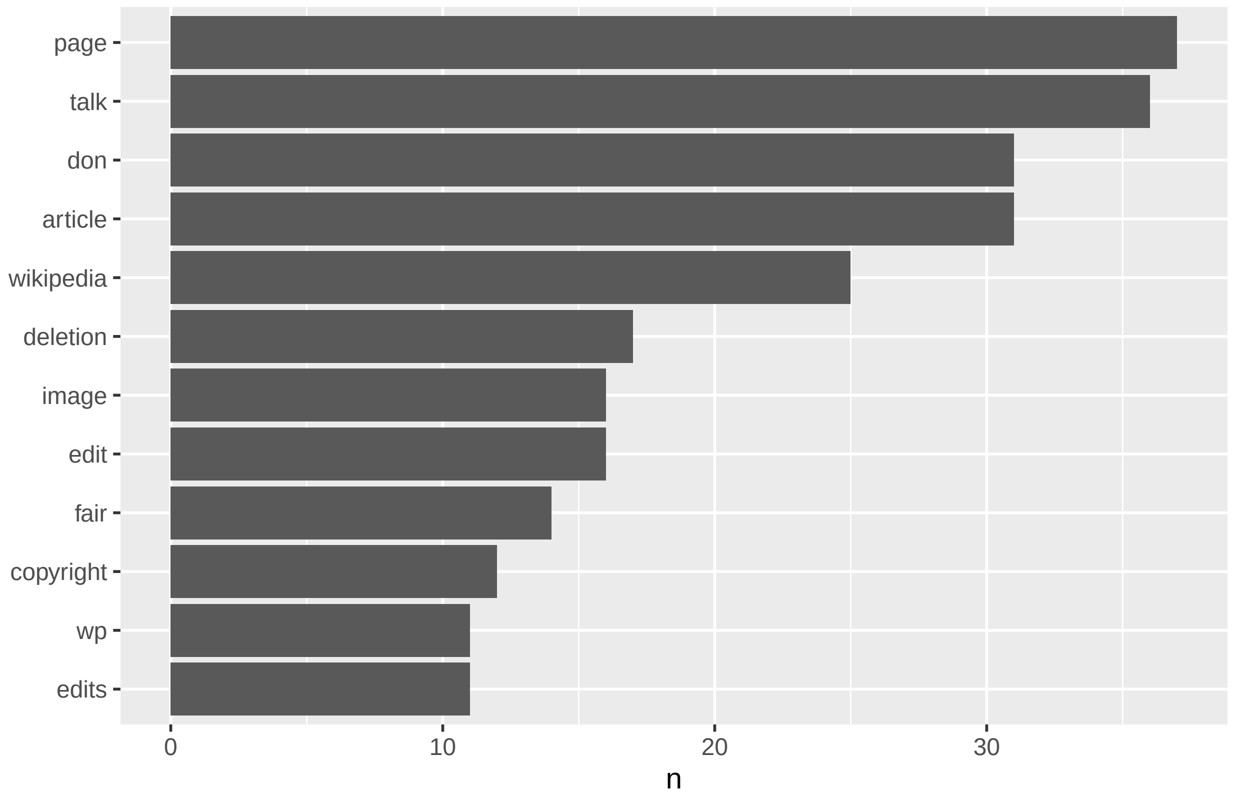
Data Cleaning

In this section, we need to detect and remove inconsistencies from data to get a tidy version of the data set. By taking a look of few comments, we find that there are some weird notations, sequence of numbers, and stop words (words that are not useful for an analysis) such as "I" and "they". Our goal is to clean these unrelated and weird texts. To do so, we replace all punctuation and numbers in comments with space, then create a data frame containing text, and convert this so that it has one-token-per-document-per-row. For instance, if the original comment is ("This is really good.10086"), then our output data set will be ("This", "is", "really", "good"). Formatting data into this form, we are able to remove extremely common words such as "the", "of", "to" and export the tidy version of the data set. In this new data set, viewers can see every single word with its number of line appearing in the data set.

EDA

After having a tidy version of the data set, we have to identify some patterns and trends of it and gain insights about which variables should be included in our first model. By using tidy tools to store word counts in a tidy data frame, we create a visualization of the most common words among all comments.

Figure 1.1



We find that words page, talk, don, article, wikipedia, deletion, image, edit, fair, copyright, and wp appear the most frequent. To further explore different types of toxicity in nasty comments, we want to gain insights about each category(toxic, severe toxic, obscene, threat, insult, and identity hate).We decide to divide the data set into two subsets: Neutral Data (if the comment is not classified as any one category of toxicity) and Nasty Data (if the comment falls into at least one of the six categories of toxicity) by creating and adding a new binary variable IsNasty. We find that there are 16225 comments in the Nasty Data and 143346 comments in the Neutral Data. In each subset, visualizations of most common words are created. There will be vulgar and offensive words appearing in the following graphs.

Figure 1.2

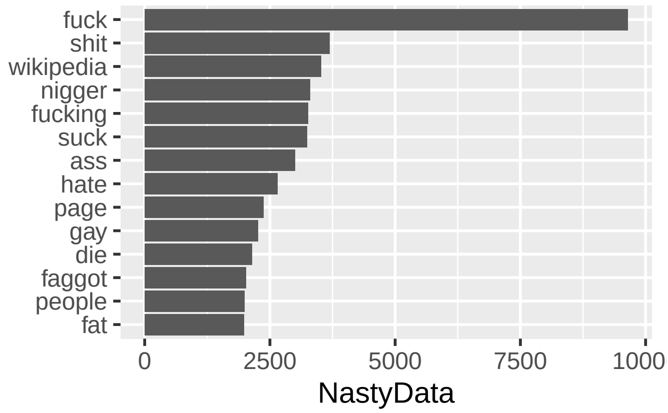
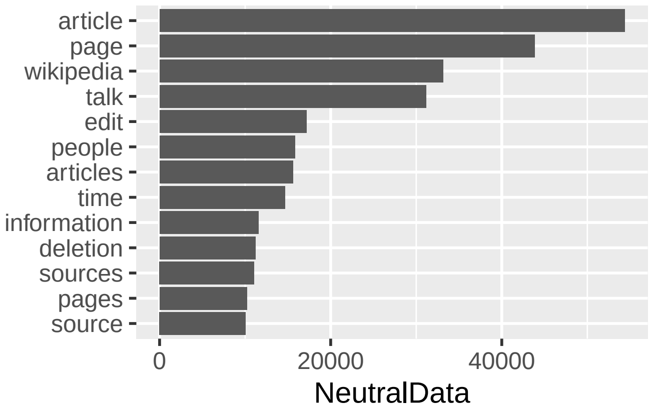


Figure 1.3



Nasty Data contains mostly offensive and vulgar words, whereas Neutral Data contains neutral and common used words. However, there are some overlapping words such as “Wikipedia” and “page” appearing in both data sets. From here, words that are unique to each data set can be used to build our first model in differentiating Nasty and Neutral comments. We also plot the most common words in each category of toxicity in the Nasty Data and compare word frequencies between them.

Figure 1.4

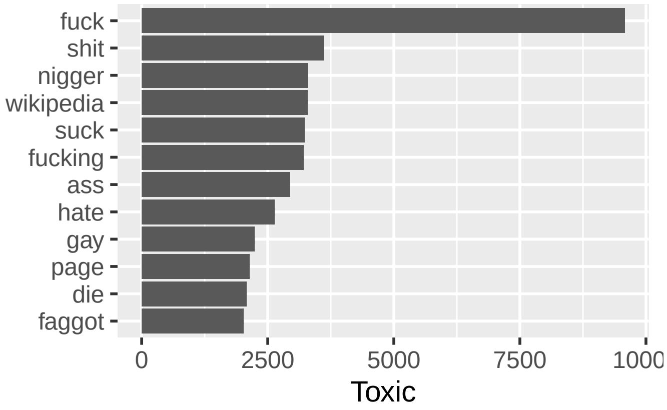


Figure 1.5

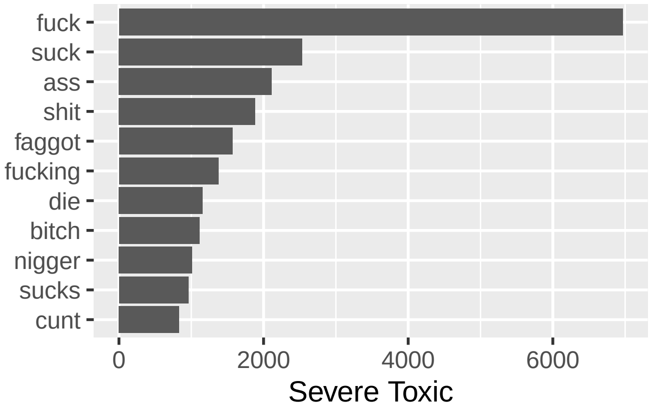


Figure 1.6

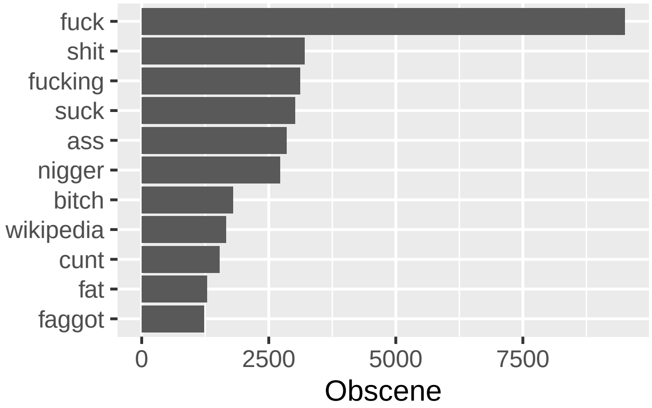


Figure 1.7

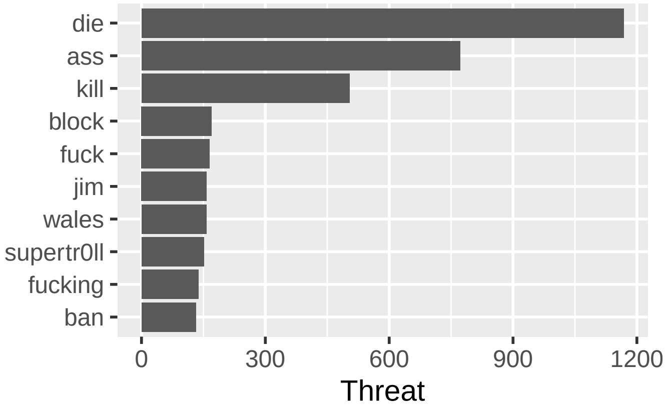


Figure 1.8

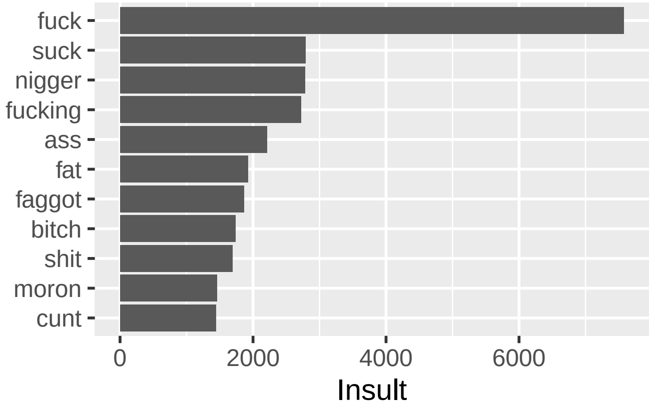
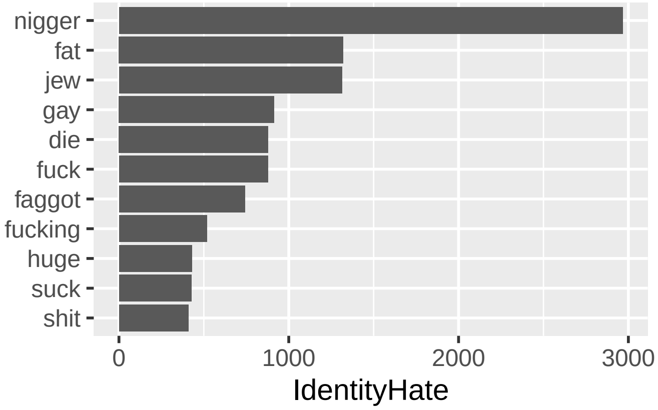


Figure 1.9



The top three common words in categories Toxic, Severe Toxic, Obscene, and Insult are overlapped with each other. Categories IdentityHate and Threat has some unique words that are different from others. To further compare types of toxicity in pairs, we create frequency comparison plots.

In sum, common words to all six categories should be ignored so that each type's unique characteristics will be disclosed. These common words should not be considered as identifiers in the following models since they are apply to any kind of toxic comment and are lack of representatives.

Section3: Methods

Model1: Logistic Regression

Our first goal is to create a model to differentiate nasty comments from neutral comments. To do so, we choose the Logistic Regression. From the last section, we have found ten most frequent words in Nasty Data and Neutral Data and decide to choose the most common one word in each subset as identifiers. By creating two binary variables named "F" and "Article" in the data set, we keep track of their appearances in each comment. Their values are 1 if such words appear in a comment and 0 otherwise. Below is the regression model we used:

Let Y_i, $i = 1, \ldots, 159571$, be a comment in the data set.

Let π_i be the probability that a comment Y_i is classified as nasty, i.e.,$P(Y_i) = 1$

Let $Articla_i$ be an indicator that comment Y_i contains the word "article" and F_i is an indicator that comment Y_i contains a specific word.

For a full list of features, see the Appendix.

$$Y_i \sim Bernoulli(\pi_i)$$

$$logit(\pi_i) = log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1(F_i) + \beta_2(Article_i)$$

The fitted line we get is:

$$log\left(\frac{\hat{\pi_i}}{1-\hat{\pi_i}}\right) = -2.143 + 5.588 * F_i - 1.344 * Article_i$$

To see how our model behaves, we create the confusion matrix and calculate the Classification Error Rate:

LogisticModel	ActualNasty	ActualNeutral
PredictedNasty	2856	13281
PredictedNeutral	13369	130065

This is the resulted confusion matrix we get. From the matrix, 17.6% of Nasty Data is identified as nasty, and 90.7% of neutral data is identified as neutral. The CER is then calculated by dividing sum of values on the off diagonal by the total number of observations in our data set. CER turns out to be about 16.78% which shows that the logistic model does a good job in differentiating nasty comments from neutral comments. Our next goal is to identify specific types of toxicity by fitting classification trees on nasty comments.

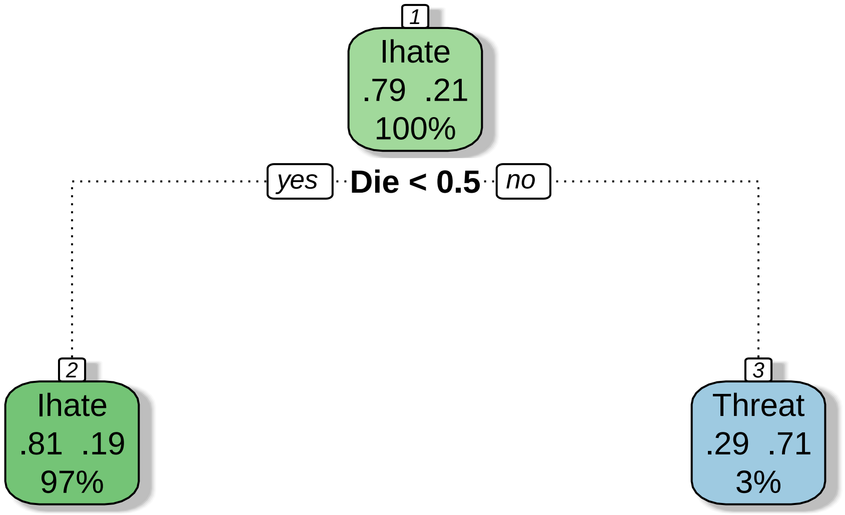
Model2: First ClassificationTree(IdetityHate vs. Threat)

CombinedCategories	Numbers
Toxic ann Nasty	15294
Toxic and Obscene	7926
Toxic and Insult	7344

When we explored comments that are classified as more than one category in Nasty Data, we found that there are 15294(about 94.3%) comments are toxic, 7926(about 48.9%) comments are categorized as toxic and Obscene, and 7344(about 45.3%) comments are categorized as toxic and Insult. Therefore, we decide to merge categories Toxic, Obscene, and Insult because of the high percentage of overlapped categorizations. To do so, we create another categorical variable named “categories”" to classify each comment to one of the three types of toxicity(TOI, IdentityHate, or Threat). In our new data set, there are 1405 IdentityHate comments, 380 Threat comments, and 14440 TOI comments.

To further split these three types of toxicity, we want to fit a classification tree by using most common words in these sub-categories as identifiers. Since there are much more comments being classified as TOI than as IdentityHate and Threat and the Classification Tree requires a balanced number of observations in each category, we discard category TOI and want to focus on the classification of IdentityHate and Threat first. From previous frequency plots, we pick a total of four unique words in these two categories and create some potential identifiers. Before actually fitting the first classification, we randomly split the nasty comments into two smaller data sets: the Training data and Test data. The Training data is used to fit the model, whereas the Test data serves as the proxy for new data. Speaking of our NastyData, we randomly choose 90% observations in NastyData(14600) as Training and the remaining 10% observations in NastyData as Test. We then extract all comments that have classification as Ihate and Threat from the training data and use these potential identifiers to create the first classification tree.

IdentityHate vs. Threat



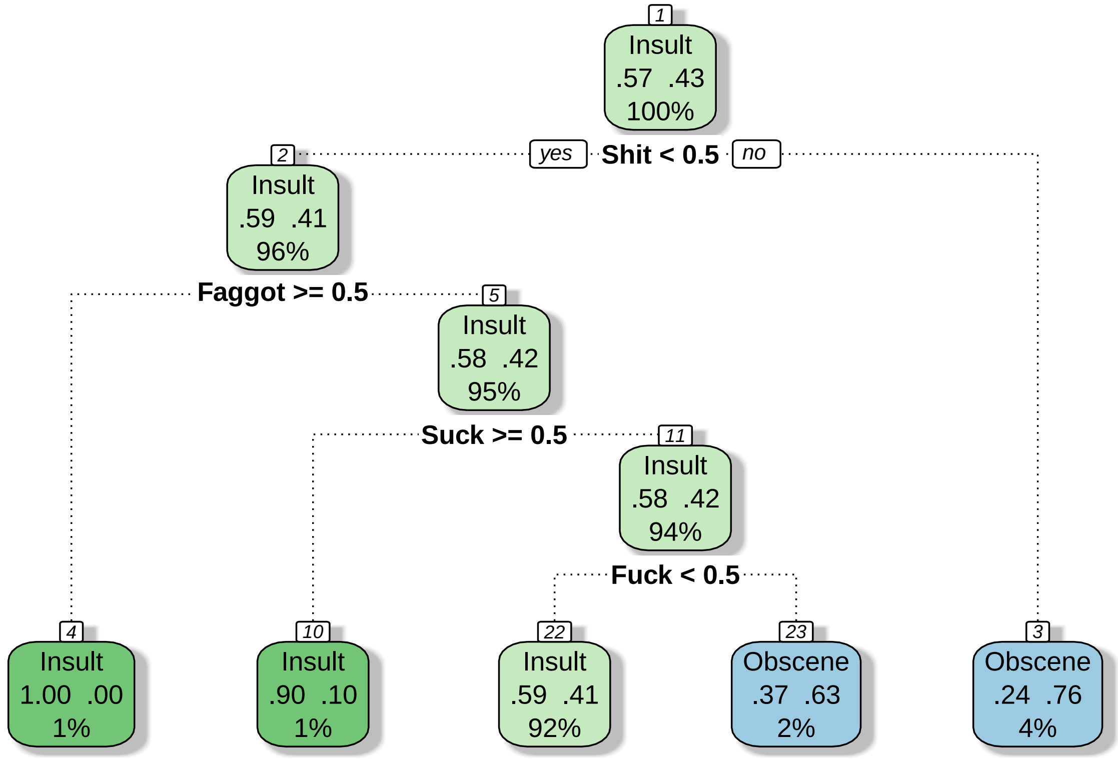
Rattle 2020−Nov−26 17:00:23 yus17

According to our tree, the identifier “D” is used to split the first giant cluster into two smaller ones: Ihate(stands for IdentityHate) and Threat. If “D” is smaller than 0.5, which means the specific word does not appear in the comment, the comment is more likely to be classified as IdentityHate. If “D” is bigger than 0.5, which means the specific word does appear in the comment, the comment is more likely to be classified as Threat. In the IdentityHate cluster, there are 81% of IdentityHate comments and 19% of Threat comments. In the Threat cluster, there are 29% of IdentityHate comments and 71% of Threat comments. The CER of training data and test data are 19.4% and 26.5% respectively. By using just one identifier “D”, the classification tree actually gives out an acceptable accuracy although there are mis-classified comments. To further classify comments in the merged data set TOI, we create more identifiers and fit another classification tree.

Model2: Second ClassificationTree(Toxic vs. Obscene vs. Insult)

To build the second classification tree in classifying comments in TOI into three sub-categories, we decide to focus on Insult and Obscene since toxic comments in TOI data are much more than Insult and Obscene. Again, we pick some common words in these two categories, create potential identifiers, and fit our second classification tree.

Insult vs. Obscene



Rattle 2020−Nov−26 17:00:25 yus17

According to this classification tree, identifiers “S” and “F” are effective ones. If S is smaller than 0.5, which means the word does not appear in the comment, the comment is more likely to be classified as Insult. If S is bigger than 0.5, which means the word does appear in the comment, the comment is more likely to be classified as Obscene. To further split Insult cluster, identifier F is used. If F is smaller than 0.5, which means the word does not appear in the comment, the comment is more likely to be classified as Insult. If F is bigger than 0.5, which means the word does appear in the comment, the comment is more likely to be classified as Obscene. The resulted Insult cluster contains 59% Insult comments and 41% Obscene comments. The Obscene cluster contains 24% Insult comments and 76% Obscene comments. Because of lack of observations in both categories, there is a big mis-classification exists. We then calculate the CER for Training data and Test data again.

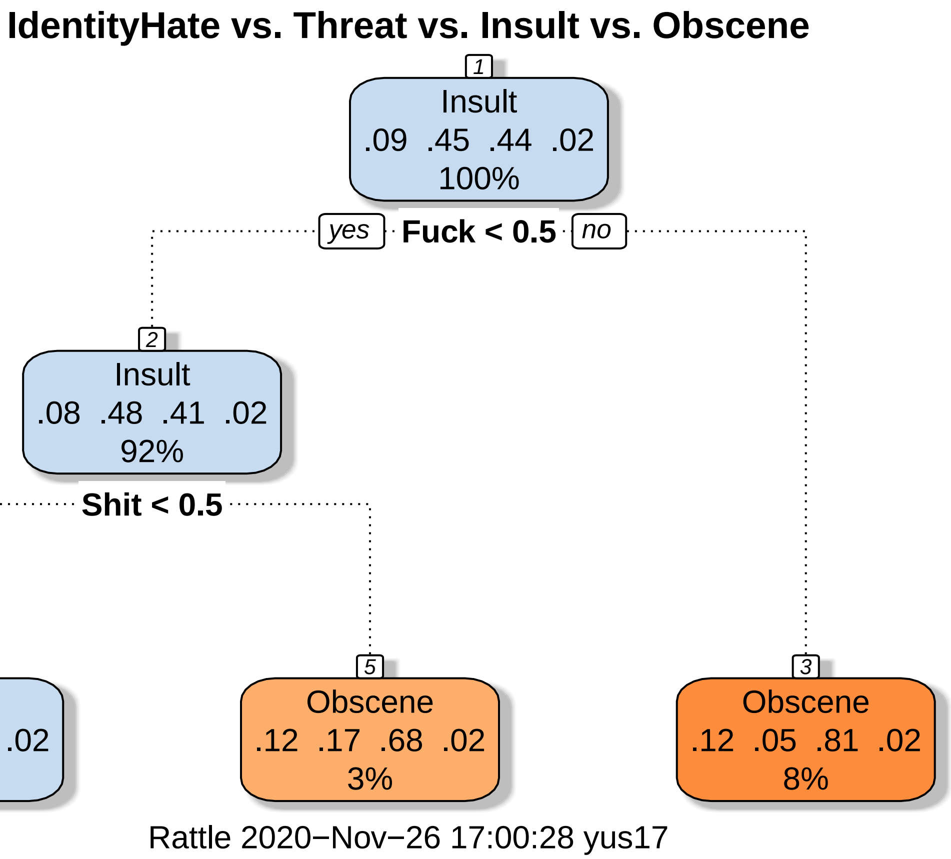
TrainData	ActualInsult	ActualObscene
PredictedInsult	469	321
PredictedObscene	15	38

TestData	ActualInsult	ActualObscene
PredictedInsult	49	39
PredictedObscene	0	0

From the Confusion Matrix table for training and test data, their CER are 39.86% and 44.32% respectively. In the test data table, we see that our model does not predict any comment as Obscene which is a big problem due to the lack of observations in Obscene comments.

We fit two classification trees to identify Threat vs. IdentityHate and Obscene vs. Insult. Now we want to combine these four categories together and fit a final classification tree to see how our model works.

Model2: Third Classification Tree(combine four chosen categories)



By combining four chosen categories(Obscene, Insult, Threat, and IdentityHate) together, there are only two categories shown in the tree plot due to an unproportional observations in these categories. According to this classification tree, identifiers "S" and "F" are effective ones. The resulted Insult cluster contains 8% IdentityHate comments and 50% Insult comments, 40% Obscene comments, and 2% Threat comments. The Obscene cluster contains 12% IdentityHate comments, 17% Insult comments, 68% Obscene comments, and 2% Threat comments. Because of lack of observations in Threat and IdentityHate, we need more observations to identify them. We then calculate the CER for Training data and Test data again.

TrainData	ActualIdentityHate	ActualInsult	ActualObscene	ActualThreat
PredictedIdentityHate	0	0	0	0
PredictedInsult	1070	6373	5124	197

TrainData	ActualIdentityHate	ActualInsult	ActualObscene	ActualThreat
PredictedObscene	213	150	1334	39
PredictedThreat	0	0	0	0

TestData	ActualIdentityHate	ActualInsult	ActualObscene	ActualThreat
PredictedIdentityHate	0	0	0	0
PredictedInsult	101	691	596	40
PredictedObscene	21	23	149	4
PredictedThreat	0	0	0	0

From the Confusion Matrix table for training and test data, their CER are 47.21% and 48.31% respectively. In the test data table, we see that our model does not predict any comment as IdentityHate and Threat. Again, we need more observations from these two categories to make the classification tree more effective.

Section4: Results

Model1

The summary of our first model shows that identifier F_i has a positive relationship with classifying a comment as nasty and identifier $Article_i$ has a negative relationship with nasty comments. Both of them are statistical significant and have p-values approximately 0. To see how our model behaves on classifying a random comment, we create a confusion matrix to compare our predictions to truths. The classification error rate turns out to be about 83.2%. Although the rate seems high at first glance, it is a good start since we just use include two identifiers in the model. If more identifiers are used, better results may be obtained.

Model2

We fitted three classification trees in the second model and found words that are responsible to identify the final four chosen categories. Due to the unproportion among observations in each category, we still have a high classification error rate. There are much fewer observations in Threat and IdentityHate comments. If more observations from these two categories are found, better results will be achieved. Or, we may try other models such as Random Forests in the future to reach better accuracy.