Classifying Toxic Comments

USING MACHINE LEARNING TO FIND THE BAD ONES

A DATA SCIENCE CAPSTONE PROJECT BY DOVID BURNS

Main Problem and Client

- Toxic comments posted in public forums online are common, and they are so corrosive that they rapidly shut down otherwise engaging discussions
- Many Platforms, e.g. Facebook, YouTube, Twitter, Wikipedia, Yelp and Instagram
- Conversation AI team have models already but these make too many errors

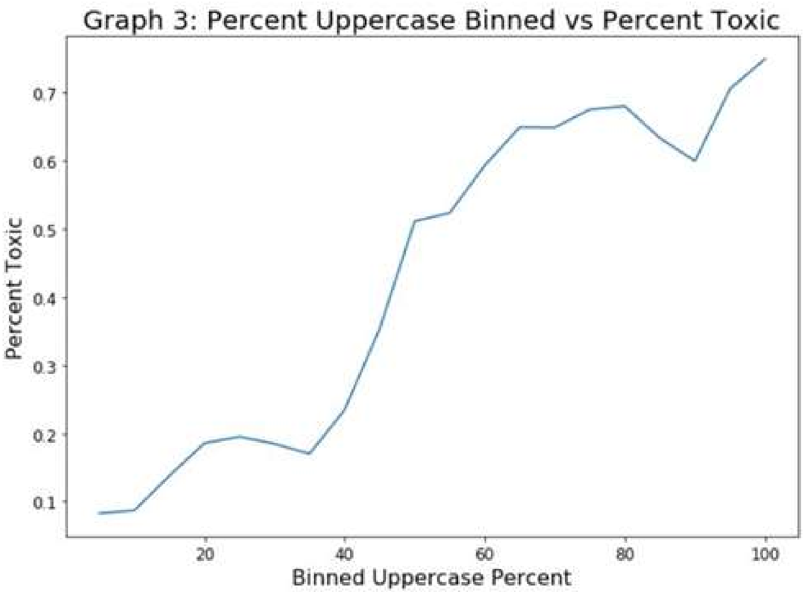
Proposed Solution

- Build a classifier using dataset of comments from Wikipedia's talk page edits: classify comments as toxic or benign
- The model can be used in many platforms for automatic detection—and removal—of toxic comments
- Model can be used to track toxic behavior through time across multiple platforms to flag chronically problematic users

Awesome Dataset

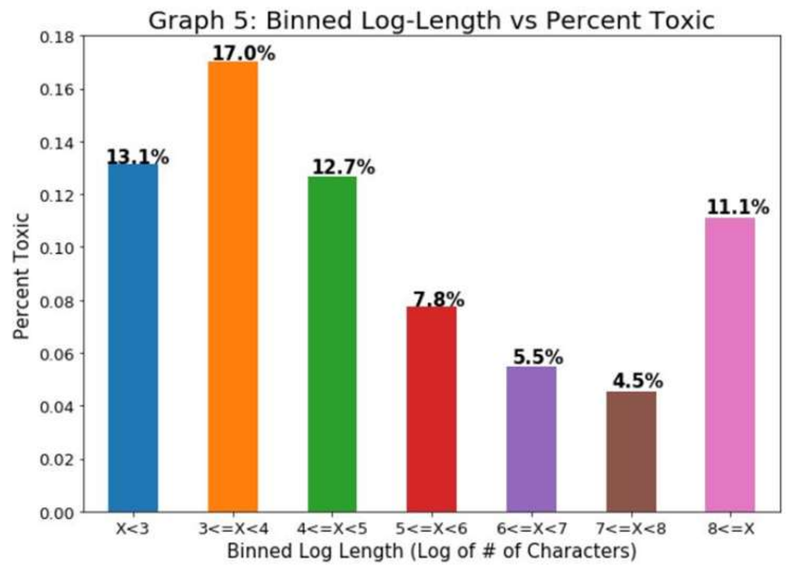
- 159,571 comments taken from Wikipedia talk pages
- 15,294 were manually tagged as toxic by human graders

Binning: Log-Length + Percent Uppercase



- Binned starting at 0-5%, increasing by 5% each time to see the trends in percent of toxic
- Percent Uppercase Bins: X < 0.1, 0.1 <= X <0.45, 0.45<= X < 0.55, 0.55 <= X
- Examined Length vs Percent Toxic and Log-Length vs percent toxic
- Log-Length Bins: X<3, 3<=X<4, 4<=X<5, 5<=X<6, 6<=X<7, 7<=X<8, 8<=X

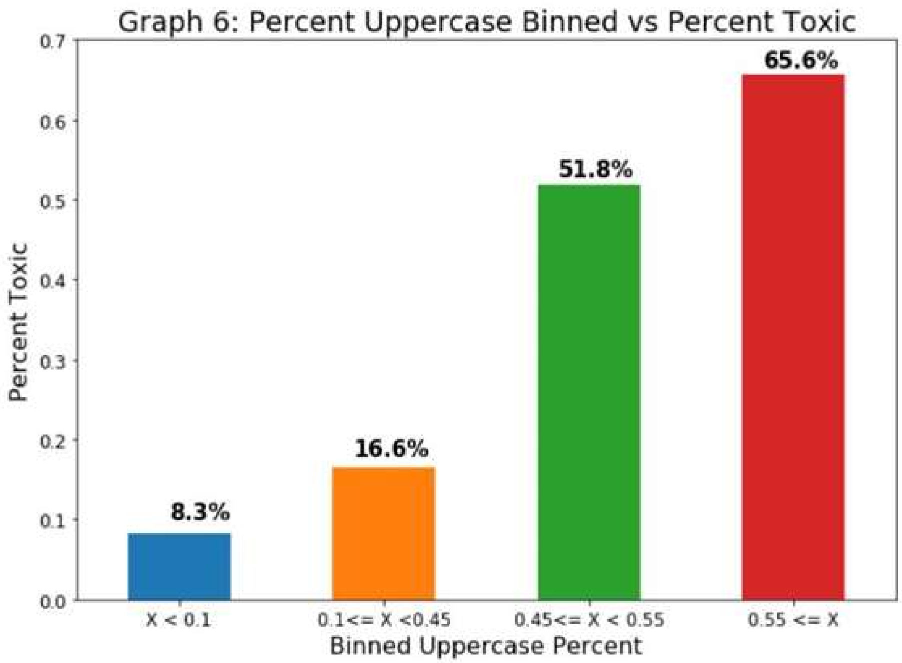
Binned Log-Length Variable



- General trend is a negative correlation between Log-Length and Percent toxic
- This is not true for the very short and very long comments
- Chi-Squared test had p-value <0.001

Percent Uppercase Variable

- The more uppercase characters in a comment the more likely to be toxic
- Makes sense – uppercase letters connote shouting
- Chi-Squared test had p-value <0.001



Most important Features From Random Forest

- Created sub-groups of comments that contain these words
- Analyzed them for percent toxic
- Chi-Squared test showed statistical significance

WORD	IMPORTANCE	PERCENT_TOXIC	P-VALUE
f*ck	0.1	94.1	< 0.001
f*cking	0.091	95	< 0.001
sh*t	0.053	78.6	< 0.001
b*tch	0.048	90.1	< 0.001
stupid	0.033	61.2	< 0.001
suck	0.028	85.3	< 0.001
a*s	0.027	14.5	< 0.001
f*ggot	0.024	93.7	< 0.001
idiot	0.021	67.5	< 0.001
d*ck	0.019	73.7	< 0.001
as*hole	0.016	90.3	< 0.001
gay	0.016	54.7	< 0.001
c*ck	0.012	68.6	< 0.001
c*nt	0.012	87.5	< 0.001
bastard	0.012	81.8	< 0.001
hell	0.012	14	< 0.001
p*nis	0.01	68.9	< 0.001
n*gger	0.01	81.7	< 0.001
loser	0.008	43.6	< 0.001
f*g	0.008	88.6	< 0.001

Machine Learning Overview

Data Pre-Processing

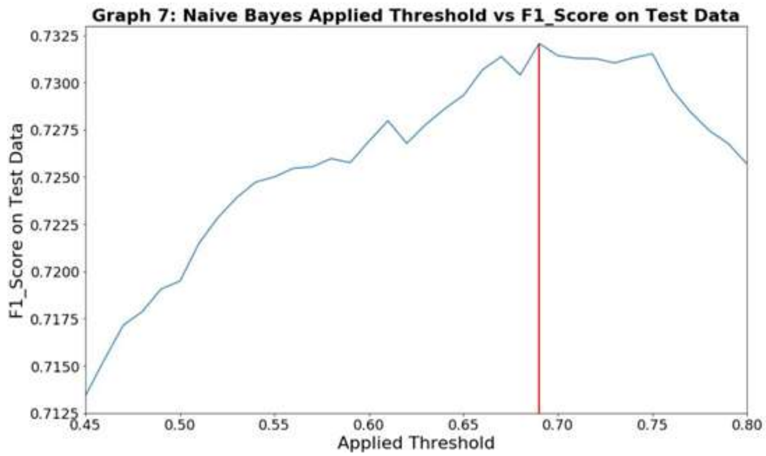
- Clean text data and tokenize
- Stem the words
- Create dummy variables from the binned variables
- Combine the two together
- Split into test and train groups for machine learning evaluation

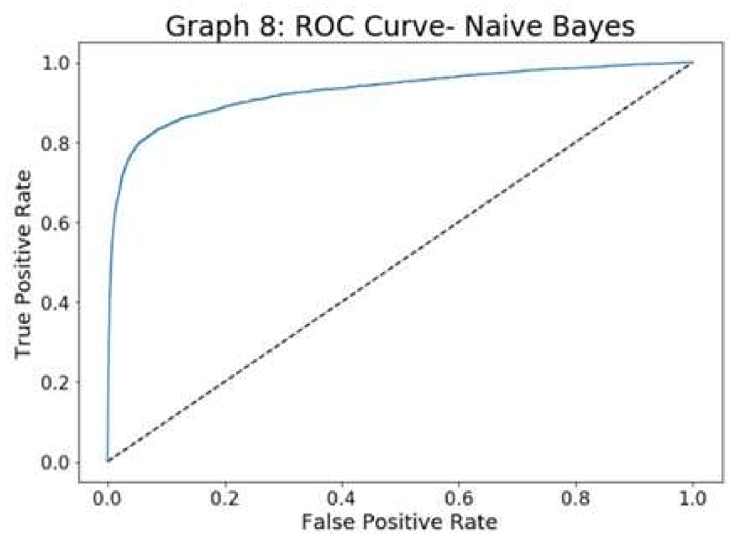
Machine Learning Workflow

- Use grid search to optimize hyperparameters
- Create three models using, Naïve Bayes, Random Forest and AdaBoost
- Find AUC scores and ROC curves for each model
- Use a custom threshold to improve F1-Score
- Find the best model based on AUC Score on test data

Multinomial Naïve Bayes

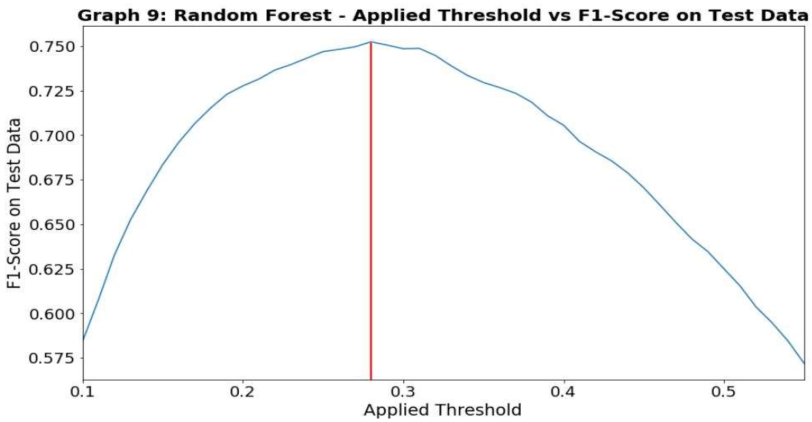
- Optimized hyperparameters with grid search
- CountVectorizer: min_df = 15, max_df = 0.2 and no max features
- Alpha =1
- AUC on Test Data = 0.928
- F1-Score with threshold 0.732

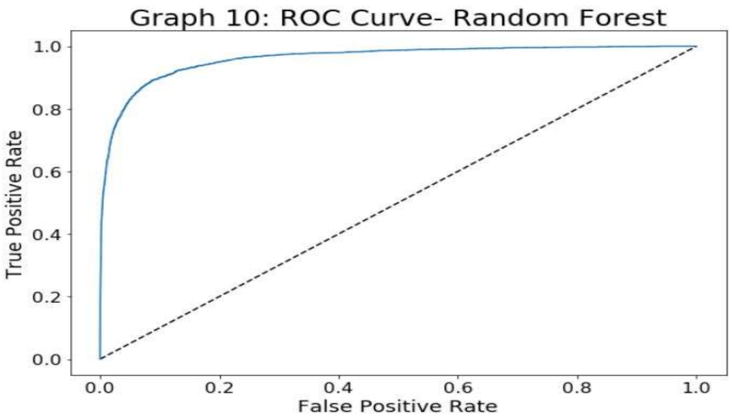




Random Forest

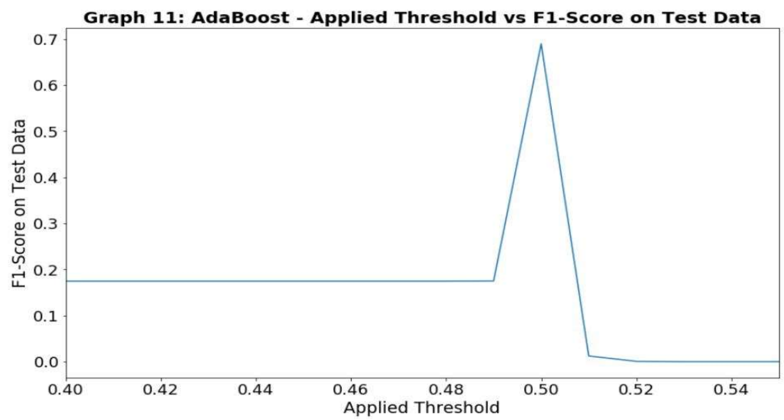
- Optimized hyperparameters with grid search
- CountVectorizer: min_df = 10, max_df = 0.2 and no max features
- Random Forest: bootstrap= False, min_samples_leaf=10
- AUC on test data = 0.961
- F1-Score with threshold 0.752

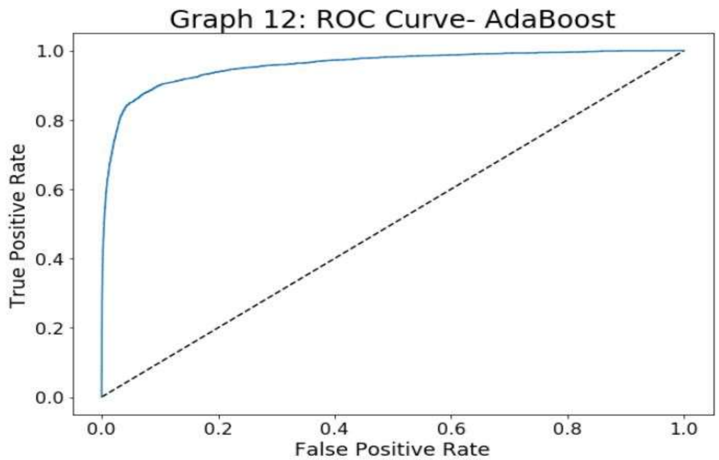




AdaBoost

- Optimized hyperparameters with grid search
- CountVectorizer: min_df = 0, max_df = 0.4 and max features = 10,000
- AUC on test data = 0.957
- F1-Score 0.69
- Changing threshold did not increase F1-Score





Additional Data to Improve the Models

- Broader comment base: Including YouTube, Facebook or Quora
- Time elapsed from initial post to the comment response
- Amount of time the user was part of the community
- Past comment history, lots of toxic comment or few/ no toxic comments

Conclusion: Advice to Client

- We successfully built a very strong Random Forest Classifier that can improve Conversation AI team's current models
- We recommend that public platforms implement a system of blocked or flagged words based on the Random Forest model's top predictors
- Score users retroactively using model to flag for extra monitoring or ban from making future comments based on individual history of posting toxic comments