Toxic Comment Classification Using Logistic Regression

EC1

Harsha Vardhan, Khurdula.

CS59000- Natural Language Processing

Professor: Jon, Rusert.

March 5, 2023.

Task

The task at hand it so perform *text classification* for a set of comments which consists of both *toxic* and *nontoxic* text. This is has be achieved using *Logistic Regression*. Draw and contrast the results yielded by this *LR* model against a simple *Naïve Bayes classifier*.

Data Analysis

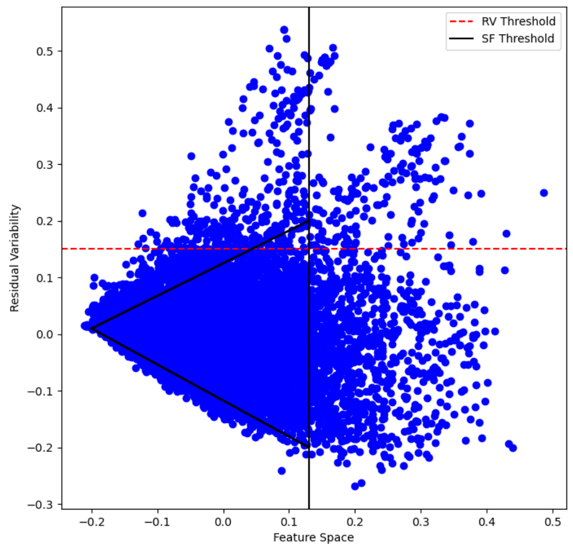
We shall use the same *trainset* used for Naïve Bayes, the goal is to provide consistent *trainsets* to both these models and then see how different the models are in terms of predictions made on ***trainingset***. This train set has to be normalized, as the comments contain a lot of special characters and numbers which do not weigh much in a comment being toxic, however one can still create slurs out of special characters, but that is out of scope for this simple analysis.

Normalization is performed by:

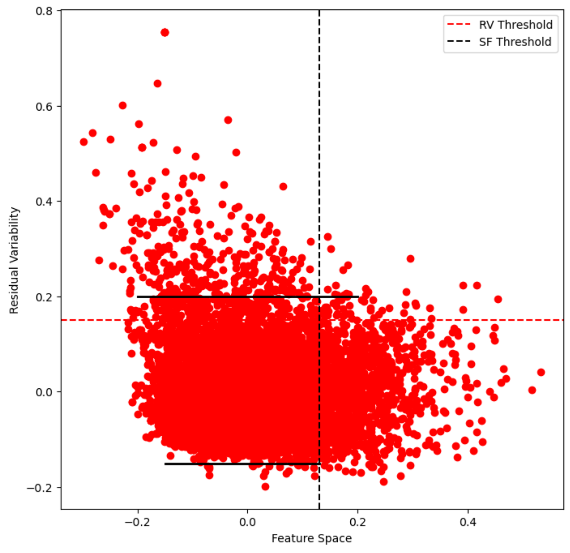
- Case conversion: converting the text into lower case for case consistency.
- Replacing Apostrophe: the character (apostrophe) can be used to fuse two words together e.x. can't: cannot, didn't: did not, we want to treat cant, and didnt as a single word after normalization.
- Capture Text only: we shall only extract words made out of alphabets within a sentence and discard the rest.
- Check for whitespace formed words: it is totally possible that a sentence can be just numerical, or empty, in that case we might return a string trailing with white spaces, and this has to be handled by discarding such strings resulting from above steps of preprocessing /normalization.

The model should be trained on word embeddings, these embeddings are create by using Tf-idf Vectorizer, that creates a sparse feature matrix for a set of comments provided and these features can be then used to train our LR model. One major difference is the fact that we shall be using Tf-idf Vectorizer, rather than using CountVectorizer that was used for the Naïve bayes classifier. It would be interesting to see how different these embeddings are when compared to the embeddings used for Naïve bayes classifier.

One of the most important things that I noticed, by a scatter plot is the following:



Plot: nontoxic n=10000



Plot: toxic n=10000

Observation:

The Residual Variability of *nontoxic* comments increases over time as we increase the count of total number of nontoxic comments passed, for *toxic* comments the Residual Variability remains close to the range -0.18 to 0.2 however, for nontoxic comments we notice a different range, starting from 0.01 to 0.2 or -0.18!

This major difference of data representation from the observations made within Naïve bayes classification could be due to use of a totally different vectorizer, but I thought this was an interesting observation.

Performance Analysis

Now, that we have our embeddings I'll try and compare the metrics for both of these models at different compositions of trian set, and note the observed metric values such as Accuracy, Precision, Recall and F1-score for each iteration.

The idea is to provide variable mix of Toxic and Nontoxic data for training, and see how they perform on testdata at once!

We shall keep the test_split seed constant with random_state=42 both both model and the split is 60-40% of total value of n for Trainning set and Test set respectively.

Training Iteration I:

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.7085	0.6528	0.8927	0.7543
LR	0.8632	0.8978	0.8284	0.8617

Table: Model metrics for trainset with toxic comments = 5000 & Nontoxic comments = 5000.

Training Iteration II:

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.7312	0.7138	0.9072	0.7989
LR	0.8718	0.8785	0.9138	0.8958

Table: Model metrics for trainset with toxic comments = 7500 & Nontoxic comments = 5000.

Training Iteration III:

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.7133	0.6570	0.8900	0.7560
LR	0.8765	0.8901	0.8556	0.8725

Table: Model metrics for trainset with toxic comments = 7500 & Nontoxic comments = 7500.

Training Iteration IV:

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.6984	0.6463	0.8787	0.7448
LR	0.8778	0.9019	0.8497	0.8750

Table: Model metrics for trainset with toxic comments = 10000 & Nontoxic comments = 10000

Observations:

Altough these are just some simple iterations of training, there are some consise observations which remain true of each of the above iterations!

- The Naïve Bayes model is either underfitted or overfitted with toxic class labels i.e. for the above iterations, Niave Bayes's performance is all over the place! The classifier either predicts everthing to be toxic, or nothing is toxic at all! There is no in between! But the LR model never has this issue throughout each iteration, not once.
- Logistic Regression model performed excellently consistent throughout each of these iterations, it had beautifully balanced metrics throughout! Totally acceptable numbers. Where as the Naïve Bayes model always tended to have a low recall score at all times.
-

Fine-tuned *Model Comparison*:

Now that we've seen how training iterations with mixed variable sets affected the performance of these models, I wanted to compare the fine-tuned models for both. For the overview of this, let us use the default thresholds for classification as 0.5 (for both).

When we try to train the LR model with 60% of entire data, we get an exception while fitting this model due to the extremely high row count of sparse matrix we obtained by fit_transform of Tfidf Vectorizer.

In order to overcome this, let's increase the max_iter limit of Logistic Regression upto 1000, from its default base value of 100.

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.9435	0.7789	0.5790	0.6643
LR	0.9544	0.9197	0.5678	0.7021

- ***Table: Model metrics for training done on entire trainset.***

One of the immediate observation that can be made is the fact that LR model has a very similar Recall Value as Naïve Bayes, we did not face this decay in Recall previously, this can be because the huge gap between the classes, i.e totalToxicComments (15500) < totalNonToxicComments (144277).

Inorder to improve this decay of performance let's reduce the totalNonToxicComments data that we fit the model with!

Training Iteration V:

Model	Accuracy	Precision	Recall	F1-Score
LR	0.9269	0.9176	0.6710	0.7752

Table: Model metrics for trainset with toxic comments = ALL & Nontoxic comments = 65000

We see an immediate improvement in Recall and F1 score, Let's further reduce the class nonToxicComments.

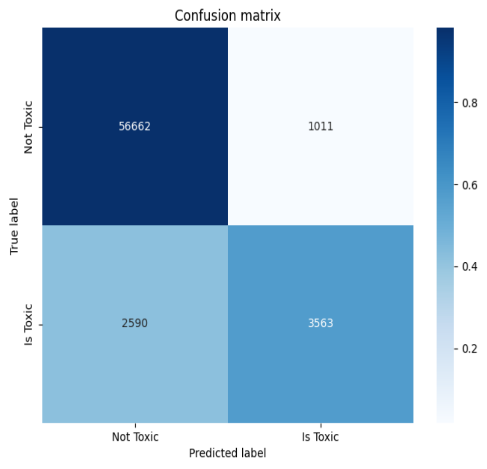
On further, reducing the class nonToxicComments, to 50000 we yield the following metrics:

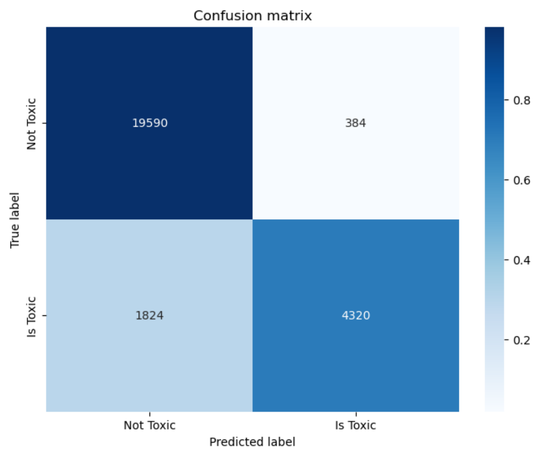
Training Iteration VI:

Model	Accuracy	Precision	Recall	F1-Score
LR	0.9154	0.9183	0.7031	0.7964

Table: Model metrics for trainset with toxic comments = ALL & Nontoxic comments = 50000

That is a decent score, but what about quantifiable results? Lets take a look at the confusion matrices:





Confusion Matrix: Naïve Bayes ***Confusion Matrix: LR***

If we look at the confusion matrices, the common issue among both is higher false negatives i.e. comments which are actually toxic are being classfied as not toxic.

Altough both these models have their differences of being a generative and deterministic models. The LR does a better job for current task of classification, there could be many possibilities as to why, few of them I think is the reason of choosing setting the word embeddings to be binary rather than compute frequencies for Naïve Bayes training vectors, produced by CountVectorizer(binary = True).

On the overview of the confusion matrix for both these models. Where the training data and test data are the same, Logistic Regression does a better job of classification. Reason being, the Naïve Bayes classifier has a lot more False Negatives than Logisitic Regression, I will try and improve the LR model to see how far I can push it. Usually Naïve Bayes should be better interms of binary classification because they tend to se distribution over x, however in this instance, our LR model is better.

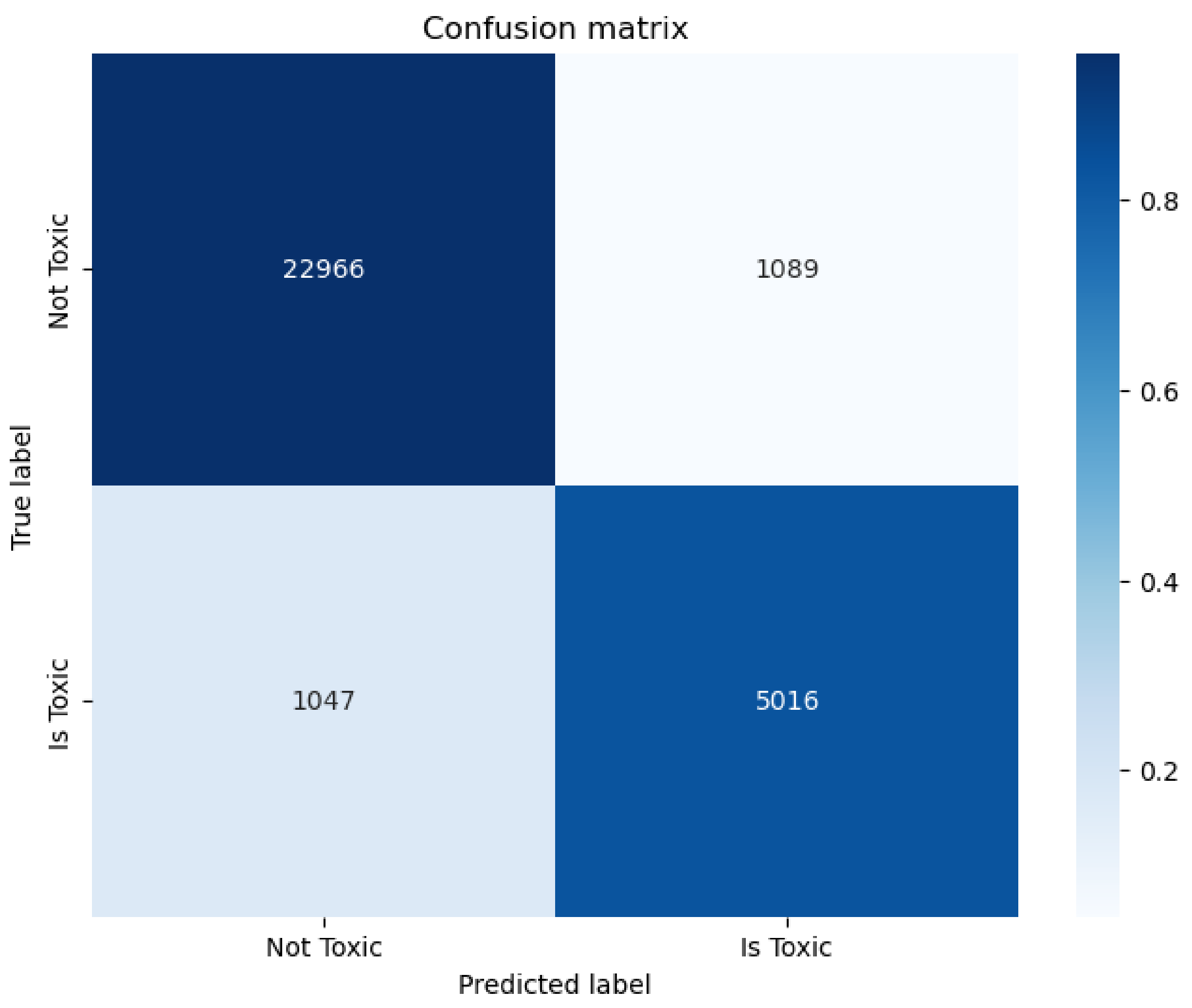
Now let's try and lower the threshold for comments to be classified as Is Toxic, and see if there is any improve in the confusion matrix for LR. By Lowering the threshold to 0.3 and Increasing the count of Non-Toxic comments, we are able to improve this by a huge margin!

Test:

Model	Accuracy	Precision	Recall	F1-Score
LR	0.9290	0.8216	0.8273	0.8244

Table: Model Metrics on Testset

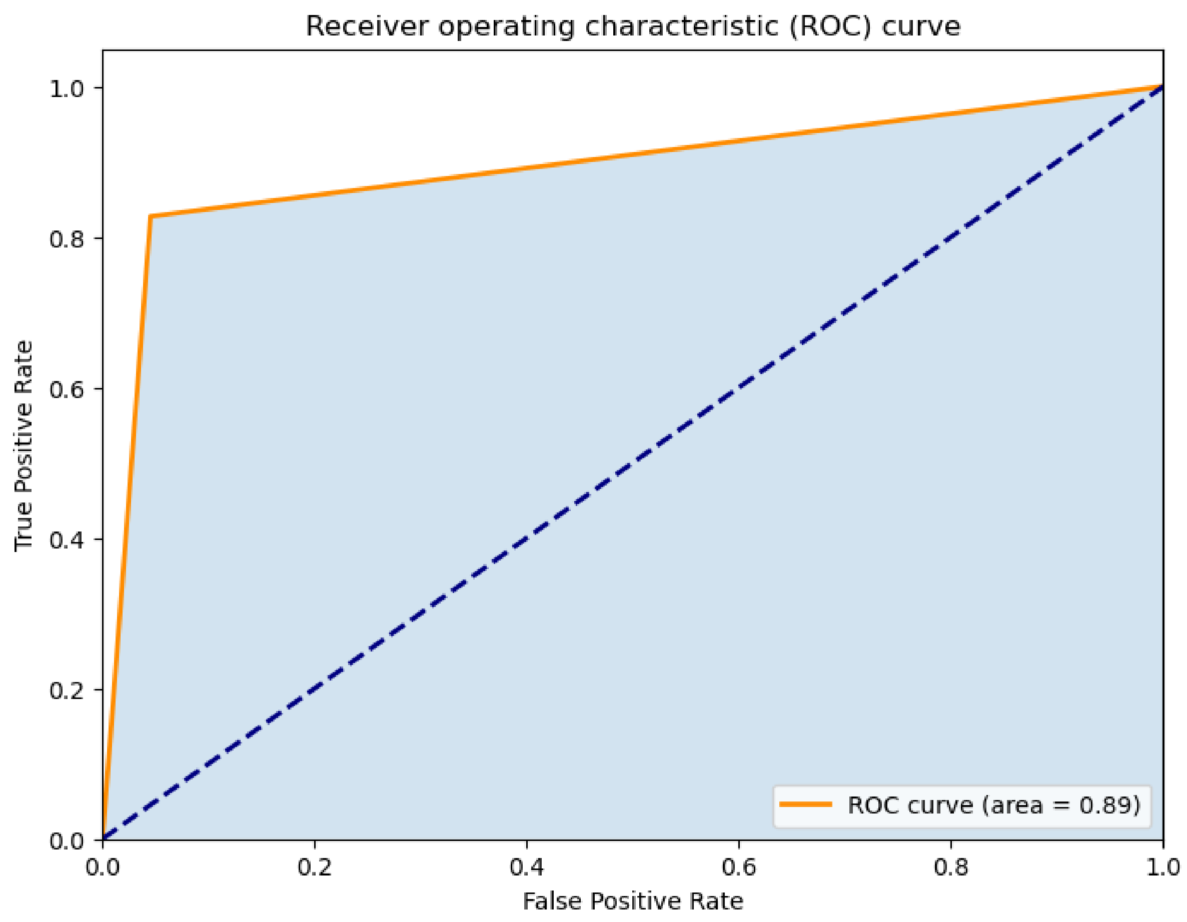
No model is perfect especially in cases like this where we are looking for user based input that can vary by use margins for comment come comment, So I believe the current LR fit is the best version possible. For the Data we are working with, we yield the following Confusion Martix which has balanced False Negatives and False Postives, we can further improve this by better data processing, or creating a new feature that can help in identifying the sentiment of the comment.



Confusion Matrix for LR with 0.3 Threshold and Increased NonToxicComments Data.

Receiver Operating Characteristic (ROC) Curve

In order to understand which threshold helps us to perform the best classification possible, and have a better Recall and F1-score, lets plot the ROC curve. For the threshold 0.3 we get the following ROC:



ROC curve

We can see that the Area Under Curve **AUC** (Area Covered) is 0.89, which is decent to good for a Binary Classifier, by considering the previous scores, and this graph we can say for now that the LR is a better classifier for the tests performed.

References:

Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing*. Stanford University Press. Draft of January 7, 2023. https://web.stanford.edu/~jurafsky/slp3/5.pdf

Khurdula, H. V. (2023, March 5). *Logistic-Regression-ToxicComment-Classification-and-Analysis*. GitHub. Retrieved March 5, 2023, from https://github.com/Khurdhula-Harshavardhan/Logistic-Regression-ToxicComment-Classification-and-Analysis.git

Khurdula, H. V. (2023, February 21). *Naive-Bayes-ToxicComment-Classification-and-Analysis*. GitHub. Retrieved March 5, 2023, from https://github.com/Khurdhula-Harshavardhan/Naive-Bayes-ToxicComment-Clasification.git

Khurdula, H. V. (2023, February 27). *Toxic Comment Classification using Naïve Bayes*. Unpublished manuscript.