273P Project Report

Jingchen Cao 70848969
Ruiyuan Zhu 40527807
https://github.com/cjc96/toxic.git

1 Introduction

In this project, we worked on Toxic Comment Classification Challenge. The tools on distinguishing negative online behaviors, like toxic comments (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion) can help us improve online conversation.

We used 4 models to solve the problem, including:

1.Logistic Regression

2.Decision Tree

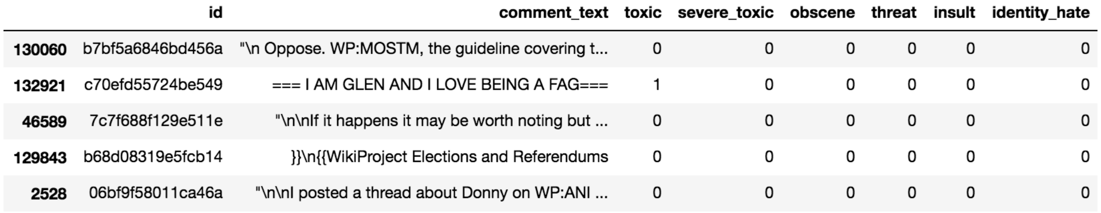
3.Random Forest

4.Long short-term memory(LSTM)

In our work, we first used some visualizing and analyzing methods to get familiar with our dataset. Meanwhile we analyzed our problem and goal. Then we chose some appropriate models to conduct experiments and finally analyzed the results.

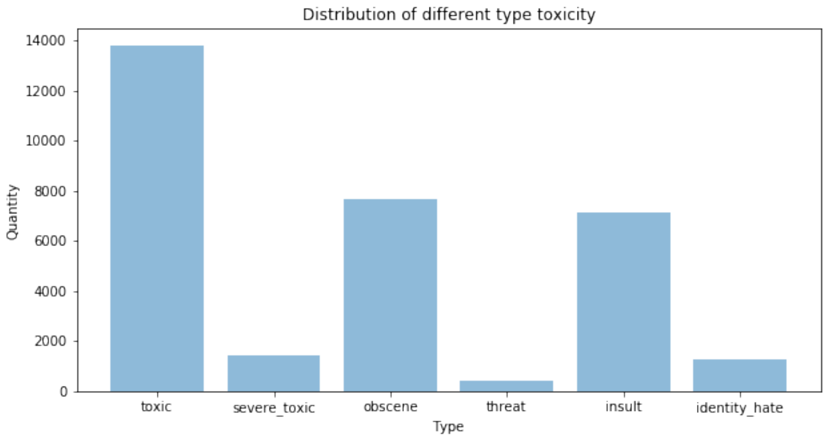
2 Data Analysis

First we got a general idea on the dataset.
In training set, there are totally 143645 rows and 8 columns. Each row refers to a comment record, including comment id, comment content and 6 labels: toxic, severe_toxic, obscene, threat, insult, identity_hate. Each record can be labelled any types of the toxicity and more than once.

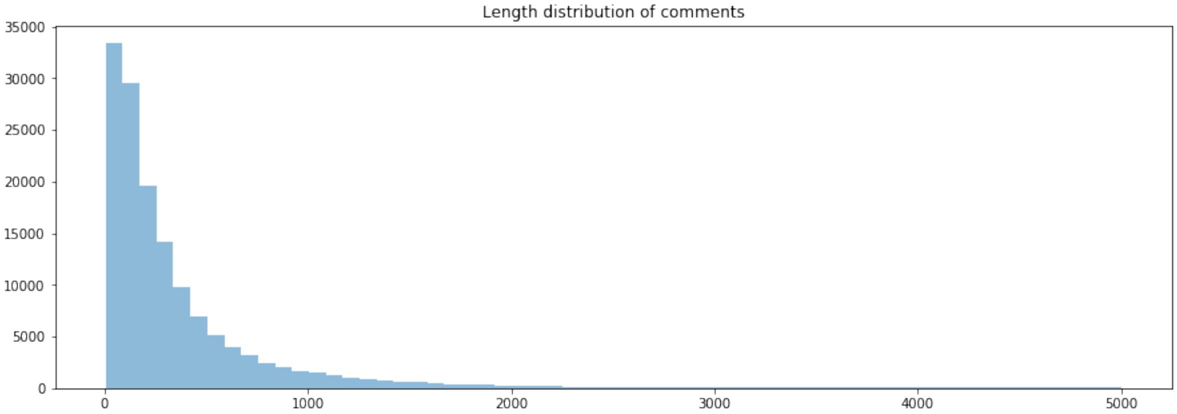


	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
130060	b7bf5a6846bd456a	"\n Oppose. WP:MOSTM, the guideline covering t...	0	0	0	0	0	0
132921	c70efd55724be549	=== I AM GLEN AND I LOVE BEING A FAG===	1	0	0	0	0	0
46589	7c7f688f129e511e	"\n\nIf it happens it may be worth noting but ...	0	0	0	0	0	0
129843	b68d08319e5fcb14	}}\n{{WikiProject Elections and Referendums	0	0	0	0	0	0
2528	06bf9f58011ca46a	"\n\nI posted a thread about Donny on WP:ANI ...	0	0	0	0	0	0

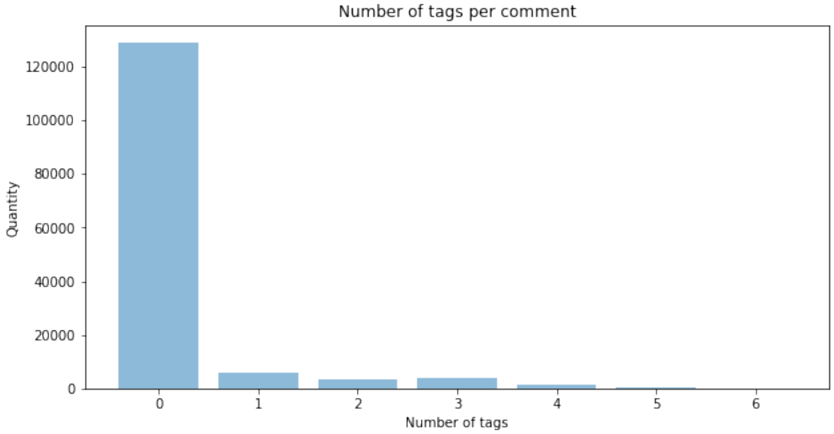
Similarly, validation set was divided from the original dataset and should be distributed similarly as training set. Test set were pure comment text and pre-labelled for evaluation.
Then we counted the exact amount of each type. We found that toxicity was not so frequent as expected compared to the large size of training data. In the different type, the ”toxic” appears most times.



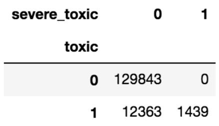
After that we found the length of comment text varied a lot. We used a histogram to illustrate.
As we can see, most comments are relatively short comments(no more than 1000 chars).



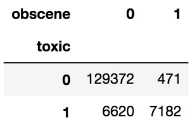
We counted the number of tags for each record here and show the distribution. We can find that most comments are clean with 0 tag as above. In remaining part, about a half has only 1 tag. Comments with 3 tags are even more than those with 2 tags, which refers to some extremely toxic ones.



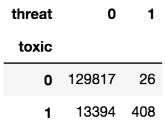
Then we explored the correlation of Toxic comments with the other types. We crosstab toxic with other columns and find:



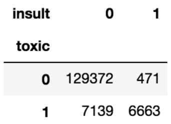
severe_toxic	0	1
toxic		
0	129843	0
1	12363	1439



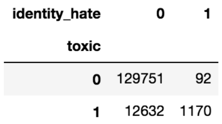
obscene	0	1
toxic		
0	129372	471
1	6620	7182



threat	0	1
toxic		
0	129817	26
1	13394	408



insult	0	1
toxic		
0	129372	471
1	7139	6663



identity_hate	0	1
toxic		
0	129751	92
1	12632	1170

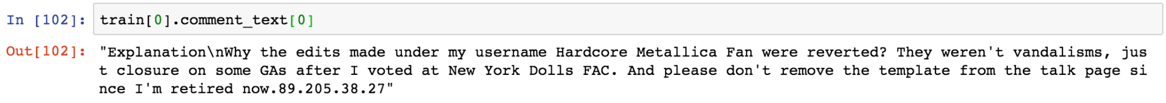
From the table we can conclude:
1.A severe toxic comment is always toxic.
2.Other classes seem to be often toxic except for a relatively few records.
Our conclude can contribute to Decision Tree in following parts.

3 Problem Definition

Given the test dataset, we should predict y for each input, which had 6 values, referring to the probability for each of the 6 possible types of comment toxicity.
To classify the results,we found that they were evaluated on the mean column-wise ROC AUC, According to the original website. In other words, the score is the average of the individual AUCs of each predicted column.

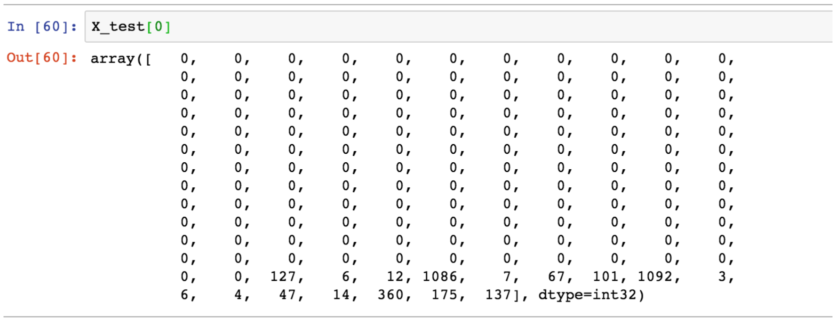
4 Feature extraction

As long as we are dealing with an NLP problem, we need to find out a method to transform comment strings into digital data, which can be recognized as training features by computers. After doing some research[1], we decide to use count vectorizer to extract features from comments. Count vectorizer will count word frequency from the given set and assign a frequency vector to each comment. We use features with different sizes during our experiments. For example, here is what a string looks like after we apply it to the vectorizer. It transformed into a 1 × 150 vector.



```
In [102]: train[0].comment_text[0]

Out[102]: "Explanation\nWhy the edits made under my username Hardcore Metallica Fan were reverted? They weren't vandalisms, jus
          t closure on some GAs after I voted at New York Dolls FAC. And please don't remove the template from the talk page si
          nce I'm retired now.89.205.38.27"
```



```
In [60]: X_test[0]

Out[60]: array([    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,  127,    6,   12, 1086,    7,   67,  101, 1092,    3,
                    6,    4,   47,   14,  360,  175,  137], dtype=int32)
```

5 Model

5.1 Logistic Regression

Logistic regression interprets $\sigma(\theta x)$ as a probability that y = 1 and uses a negative log-likelihood loss function.

[1] https://zhangzirui.github.io/posts/Document-14%20(sklearn-feature).md

It estimates the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick; these are represented by an indicator variable, where the two values are labeled "0" and "1".

5.2 Decision Tree

As we can see in the previous Analysis Part, comments with "Severe Toxic" label are subsets of comments with "Toxic" label. This indicates that there must exist a feature that its information gain on "Toxic" set is larger than "Severe Toxic" under any circumstances. Decision Tree can have a good performance to classify this data set. We train decision tree models with different Depth and maxLeaf in our experiments.

5.3 Random Forest

Since that Random Forest is an ensemble of decision tree, it should have a better performance than a single tree. Meanwhile, random forest shall not have overfitting problem[2]. We do experiments on toxic comments and justify this feature.

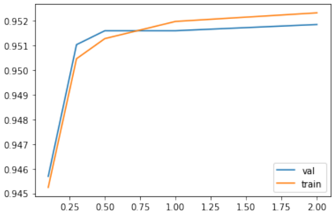
5.4 LSTM

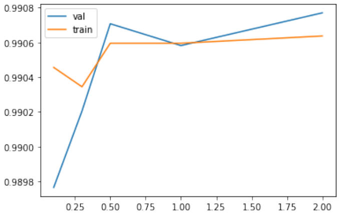
Long short-term memory (LSTM)[3] is an artificial recurrent neural network architecture used in the field of deep learning. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs.

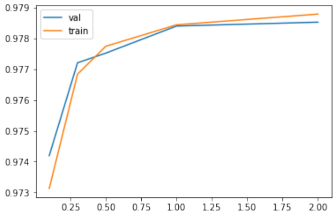
6 Experiment

6.1 Logistic Regression

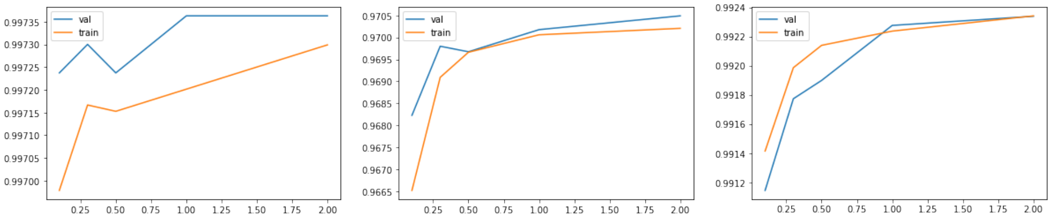
Logistic Regression can only predict one dimension of class at a time. So we train six independent model with different regularization parameter C. Later we pick different C for each model and combine them into one. The final AUC on test set is 0.896. Plots with different C for each LR model are shown below.







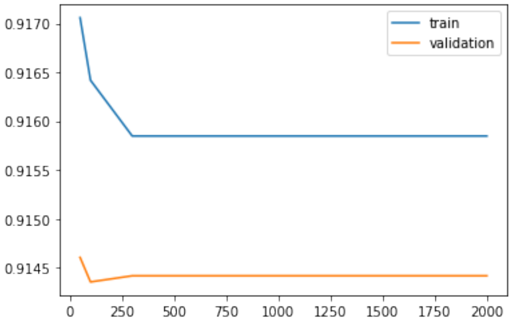
[2] https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#remarks
[3] https://en.wikipedia.org/wiki/Long_short-term_memory



6.2 Decision Tree

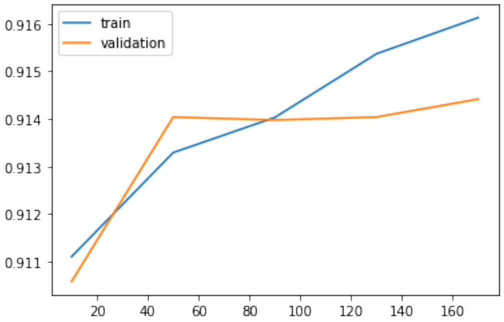
As we mentioned before, since that comments' label are overlapped to some extends. For example, all the severe toxic comments are part of the toxic comments. We believe that decision tree shall have a good performance to classify comments. We trained our decision tree model with various maxLeaf and maxDepth. Here are the plots.

Various MaxDepth:



Then we fixed the depth at 300. Fortunately, it didn't overfitted. We guess this is because our data are well organized and precisely labeled.

Various MaxLeaf:



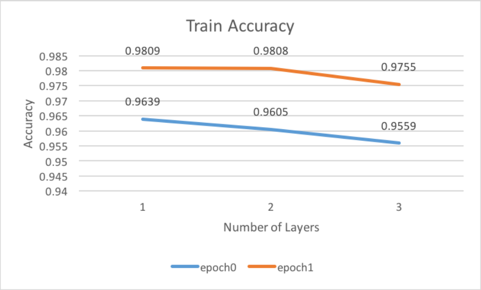
We found that maxLeaf at 160 will make our model robust. Later we did some research online. According to Users from quora and stackoverflow, maxLeaf near sqrt(data_size) shall have a good performance generally. In the end, we score an AUC of 0.873 on test set.

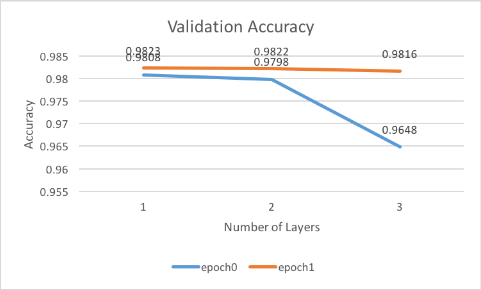
6.3 Random Forest

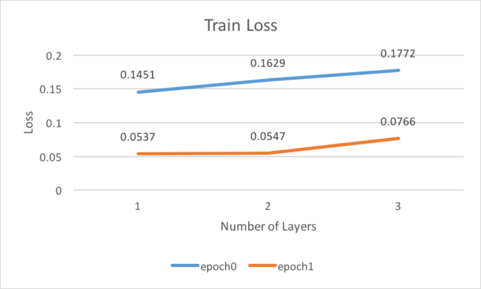
Since that random forest is the ensemble of decision tree, it shall have a better performance. We use the same maxDepth and maxLeaf as Decision Tree. This time, we score an AUC of 0.897 on test set. This justify that random forest has advantage over decision tree and ensemble is a good method for machine learning.

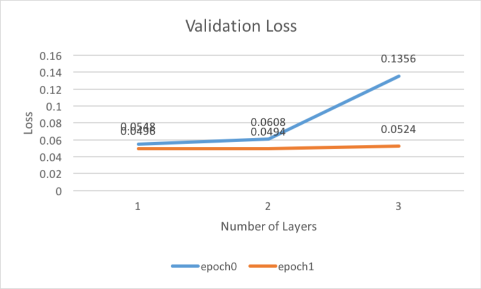
6.4 LSTM

Given that LSTM model is fit for our problem, we utilized LSTM from keras to build our own model here. It contains several layers, most of which are basic processing layers as well as one or multiple LSTM layers.
We evaluated the relationship between the number of layers and accuracy and loss with optimal arguments, and trained each of them for 2 epoches.

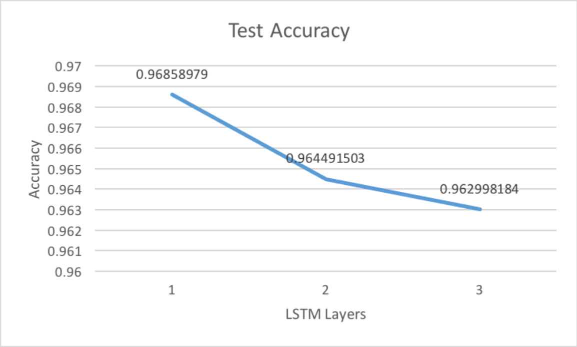








Finally we evaluated the accuracy of our model's prediction on our test dataset.



As we can see, our test accuracy is relatively high compared to other models. Meanwhile multiple layers can not benefit the accuracy and loss. We can conclude LSTM is really an effective and mature model for problems like this.

7 Conclusion

Here's the ultimate table showing the performance of four model we use in the project. We can see that Neural Network scores the best with no doubt. Since Neural Network could be the most popular method we use for Machine Learning today, its performance justifies its fame.

	Linear Regression	Decision Tree	Random Forest	LSTM
AUC	0.896	0.873	0.897	0.969

Zhu completed the data analysis and neural network part. Cao completed the linear regression, decision tree and random forest part. We wrote this report together.