Community Analysis through Toxic Comment Identification

By Nicholas Delucchi

Introduction

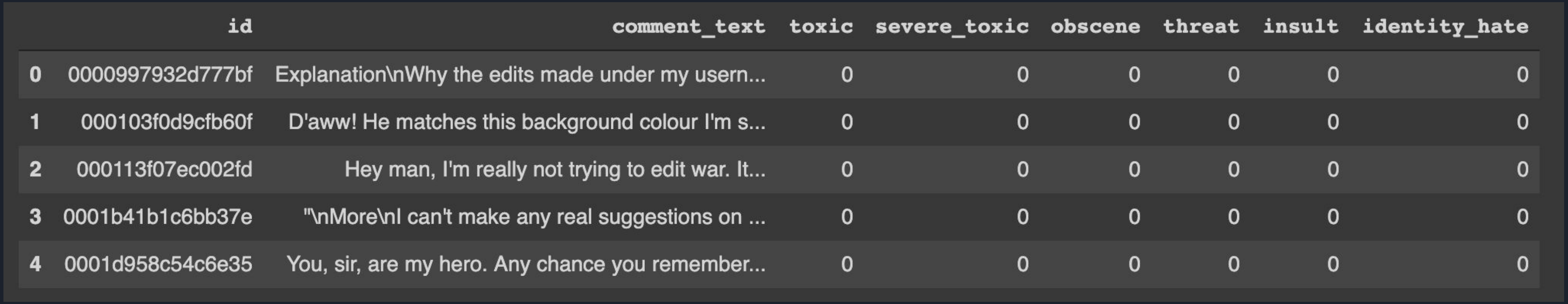
In this project we aim to:

- Identify toxic comments with high accuracy
- Explore ensemble and tensorflow models
- Prepare the final model for production use
- Have fun!

The data that we'll be training with was provided by YouTube from a Kaggle competition they ran in 2018.

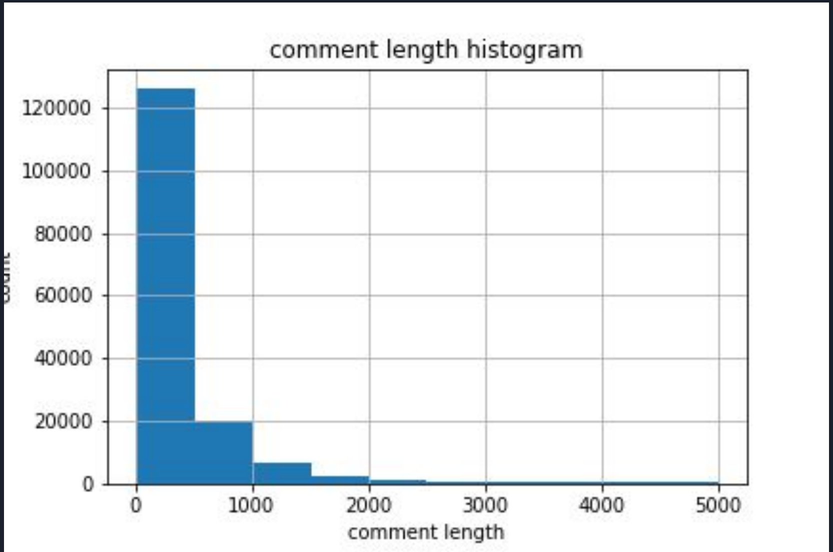
We'll be performing Term Frequency-Inverse Document Frequency analysis on the comments to train our models with in order to quantify the toxicity of a community which could then be meaningful to advertising firms.

Import Data



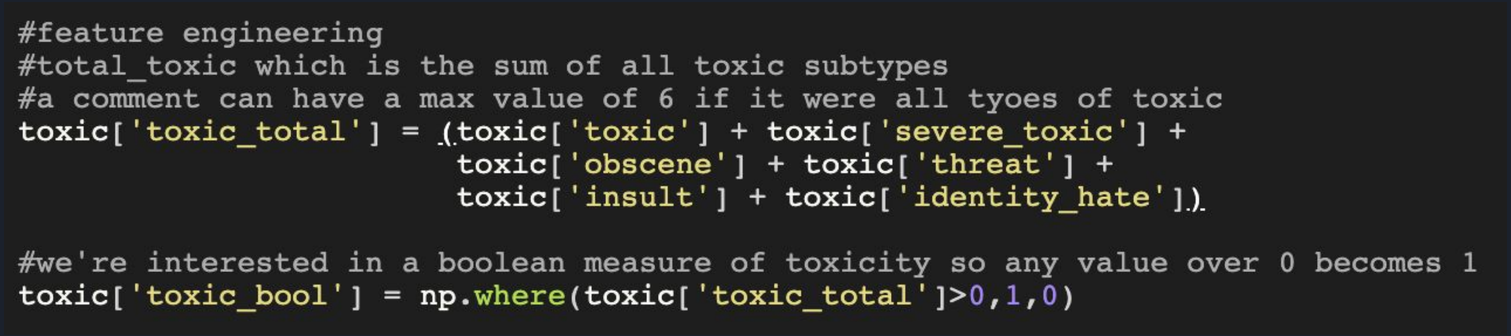
	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

Data Exploration



- Average Comment Length: 394 words
- Standard Deviation: 591 words
- Max Comment Length: 5000

Feature Engineering



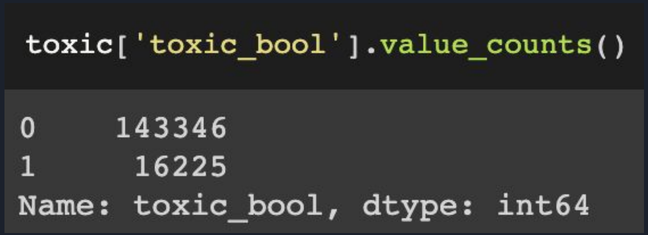
```
#feature engineering
#total_toxic which is the sum of all toxic subtypes
#a comment can have a max value of 6 if it were all tyoes of toxic
toxic['toxic_total'] = (toxic['toxic'] + toxic['severe_toxic'] +
                        toxic['obscene'] + toxic['threat'] +
                        toxic['insult'] + toxic['identity_hate'])

#we're interested in a boolean measure of toxicity so any value over 0 becomes 1
toxic['toxic_bool'] = np.where(toxic['toxic_total']>0,1,0)
```

Data Preprocessing

- Vectorize the data
- Reshape the vectorizer output using .tocsr()
- Perform TF-IDF on the comment body
- Normalize the data for easier learning

Class Balancing

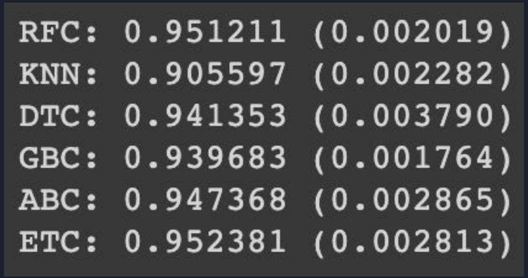


```
toxic['toxic_bool'].value_counts()
```

```
0    143346
1     16225
Name: toxic_bool, dtype: int64
```

- Our target output has a minority that only represents about 10% of the total
- Use SMOTETomek to rebalance to bring the minority up to the majority value

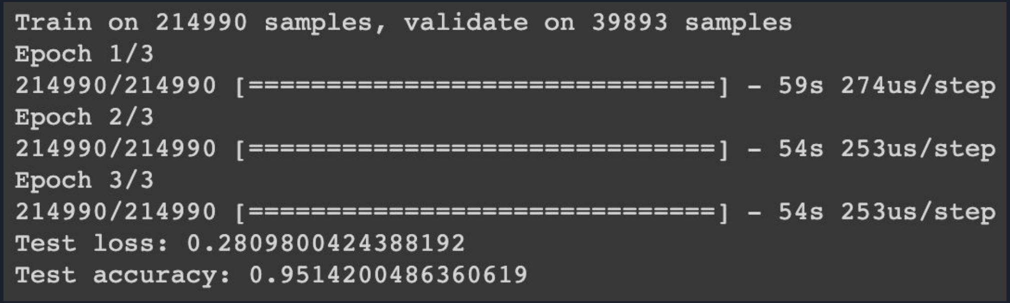
Iterate through models



```
RFC: 0.951211 (0.002019)
KNN: 0.905597 (0.002282)
DTC: 0.941353 (0.003790)
GBC: 0.939683 (0.001764)
ABC: 0.947368 (0.002865)
ETC: 0.952381 (0.002813)
```

- All models perform pretty well, but ExtraTreesCalssifier outperforms them all
- We’ll pickle the model to save it for later use

Tensorflow/Keras Model



```
Train on 214990 samples, validate on 39893 samples
Epoch 1/3
214990/214990 [==============================] - 59s 274us/step
Epoch 2/3
214990/214990 [==============================] - 54s 253us/step
Epoch 3/3
214990/214990 [==============================] - 54s 253us/step
Test loss: 0.2809800424388192
Test accuracy: 0.9514200486360619
```

- Using 'relu' activation yielded some pretty great results.
- I tested out numerous combinations of other activation functions but none yielded better results, measuring 0.93 and lower.

Conclusion

The ensemble method ExtraTreesClassifier and a Mean-Squared Error Tensorflow/Keras model performed quite well but ultimately the ensemble model ExtraTreesClassifier just barely performed the better so we'll go ahead and use a pickled version of that model going forward to analyze the youtube comments that we'll collect.

Obstacles

I thought that my biggest problem in this project would be the size of the data but it actually was determining the appropriate order of each step to get it to. I'm sure it seems obvious now in hindsight.

Another obstacle we encountered was significantly poorer performance of the model before class balancing was performed on the toxic boolean feature, where the minority represented only about 10% of the total inputs.

The Future

- Eventually the corpus will be a little out of date, missing relevant terms (i.e. slang, etc)
- Re-fit the model, could be taxing to label a brand new set of a few hundred thousand comments as various kinds of toxic.