MAIS 202 - PROJECT DELIVERABLE 3

Elinor Poole-Dayan

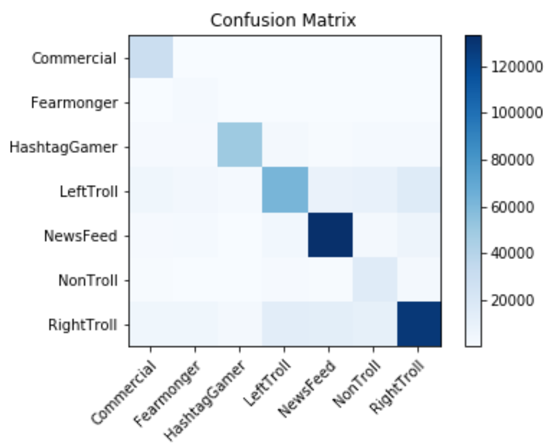
1. Final Training Results

I tried to change my model to a Recurrent Neural Network, but I was unable to get the model to run on enough of my data because it took too long. Therefore, I kept my initial SGD model and changed the loss function to 'modified_huber' in order to make the logistic loss metric available. It actually increased my train mean accuracy to 79.67% and test mean accuracy to 78.72% (from 76%).

The logistic loss is 1.108387. The precision-recall scores are:

```
[0.79465541, 0.16942645, 0.91655941, 0.69444683, 0.86670754, 0.46513871, 0.84396197],
[0.97779592, 0.75      , 0.84320592, 0.68063975, 0.89356762, 0.71907624, 0.71948273],
[0.87676407, 0.27641109, 0.87835385, 0.68747398, 0.87993265, 0.56488089, 0.77676692],
[ 30625,   2824,  59964, 106729, 150302,  22127, 180021].
```

Some sources of confusion for the model is that some of the troll categories are hard to distinguish. For example, it might be hard to tell apart a real Donald Trump tweet from a Russian troll because the word "Trump" or mentions to him "@realdonaldtrump" occur frequently in both. Also, the ambiguity of the Fearmonger category could be the reason the model found it hard to differentiate between it and the other categories, as evidenced by the relatively low numbers (light blue) for that category. Additionally, the model frequently confused left and right trolls, which could possibly be an effect of the political nature of both. Interestingly, using a bigram model did not increase the accuracy of the model.



I ran the model on some test tweets that I made up in order to see which words affect the classification the most. "#maga" and "#blacklivesmatter" were two words that largely affected the classification of the tweet. For example, the tweet "my dog is so cute #pets" was classified as a Fearmonger tweet with 19.41% certainty. When "#maga" was added to the end, it changed to a RightTroll tweet with 39.58% certainty and when "#blacklivesmatter" was added, it changed to a LeftTroll tweet with 42.99% certainty.

2. Final demonstration proposal

I am currently trying to use Flask to implement the webapp. My idea is to create a landing page where there is a short explanation of my model and an input text box where the user can input some text. There will be a button that will run the input text as a sample tweet and classify it as one of the 7 classes and output the percent certainty of the classification. I have already written code that does this in my Colab notebook, and the task is just creating the page and having that code run with the input text.

I have no experience with Flask or creating a web app, however I do have experience in web development. I am familiar with RapidWeaver, which is a sort of WYSIWYG software for iOS. Essentially, I have no experience with coding in the backend of a website. I plan to watch/read online tutorials and hopefully be able to learn it on my own.

The web address where I will be putting my webapp is http://192.241.151.161/.