

Deliverable three

Final Training results

In this deliverable, my goal was to improve the existing models that I have implemented in previous deliverables. Firstly, I have tackled some of the known issues from the previous implementation. By using "regex" I was able to remove the location tag before every mention of '(Reuters)'. While I ran into runtime issues in the implementation of regex, I was able to resolve this by using "regex" on the True.csv file instead of the prepopulated panda dictionary to improve speed significantly. I was able to make use of the article title by combining the title and the article body. I then dropped the irrelevant columns because there were inconsistencies in the input data when it comes to categorisation. I have also discovered that there are empty strings in the input data, I removed those empty ones to reduce noise. I have implemented a few more pre-processing techniques to improve the quality of the input data, by implementing stop-word removal, and lemmatization then word tokenization with some pre-existing libraries. Overall, I was able to improve the testing accuracy significantly to 93% from the previous 80%. I will attach the confusion matrix below.

```
array([[4306, 276],
       [ 334, 3939]])
```

The confusion matrix shows a fairly consistent result when it comes to predictions.

The **true positive** rate is $3939/4273 = 92\%$

The **true negative** rate is $4306/4582 = 94\%$

They are relatively consistent so I consider the Naive Bayes algorithm to be performing well

Final demonstration proposal

For the demonstration, I will construct a web app using React.js for the frontend visualisation, and Flask with python for the backend modelling. I have some previous experience with both of these technologies, although I lack experience with server deployment, technologies like firebase should not be too hard to learn just from reading the documentation. The web app should be a simplistic web page with a textbox taking inputs of an article and a textbox taking inputs of the article title. I will implement an external text escaper in the model.py part of the flask backend along with the classification algorithm so the input text will be stripped of redundant white spaces and quotation marks which could potentially break the model. The classification result can simply be a pop-up box suggesting "Fake" or "Real" .