

Research Methods in Social Sciences and Design

Assignment 6

Preprocessing Steps used:

For the purpose of this assignment, I processed the available data in the following ways:

- **Removed stop words:**
Stop words are words like 'the', 'in', 'an' etc. that don't really add any meaning to the data. Moreover, they add to the size of the dataset and increase the time taken to train the model. Moreover, since only meaningful tokens are left after removing the stop words, the accuracy tends to increase.
- **Lemmatization:**
Lemmatization helps bring context to the words, whereby different inflected forms of a word are grouped together. This helps make the analysis easy, and in turn increases the accuracy of the model.
- **Stemming:**
For the purposes of classifying the news headlines in this assignment, I skipped stemming. Stemming converts words to its root/base form, whereby words like runner, runs, running, etc. will be converted to run. In most cases, stemming helps increase accuracy, however given the small dataset we were using, stemming was decreasing the accuracy by a small percentage. So, I decided to skip stemming.

Assumptions

The fundamental Naïve Bayes algorithm assumes that each feature (or word) makes an independent and equal contribution to the data. This assumption is quite wrong in the real word scenario, however seems to work fine for most classification purposes.

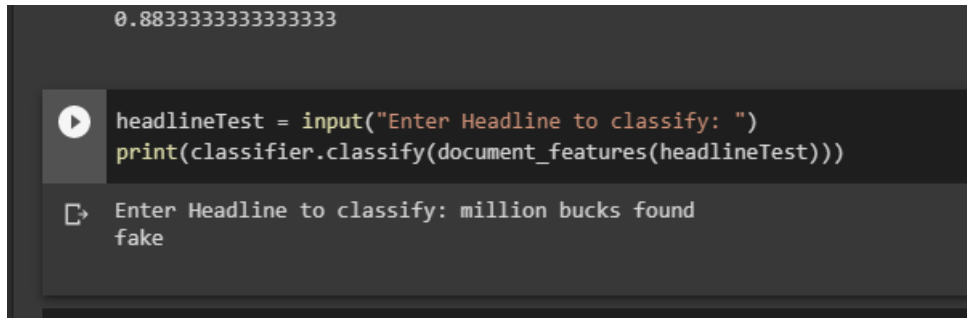
Observation

It was observed that following the correct pre-processing steps made a significant impact on the accuracy of the model. The size of the training dataset also impacts the accuracy to a huge extent.

Words like 'transcript' were usually found in real news headlines, while 'million' was usually found in fake headlines.

Accuracy:

My model returned an accuracy of 0.8833333, which implies that it will classify the given headline correctly almost 90% of the time.

A screenshot of a Jupyter Notebook interface. At the top, the output of a cell is displayed as '0.8833333333333333'. Below this, a code cell is shown with a play button icon on the left. The code contains two lines: 'headlineTest = input("Enter Headline to classify: ")' and 'print(classifier.classify(document_features(headlineTest)))'. Below the code, the input and output of the code cell are shown. The input is 'Enter Headline to classify: million bucks found' and the output is 'fake'.

```
0.8833333333333333
```

```
headlineTest = input("Enter Headline to classify: ")
print(classifier.classify(document_features(headlineTest)))
```

```
Enter Headline to classify: million bucks found
fake
```

Link to the colab code:

<https://colab.research.google.com/drive/1BP6Ghm4wmz3FpwmAjgU45n5xVD02cEWx>