FAKE NEWS DETECTOR

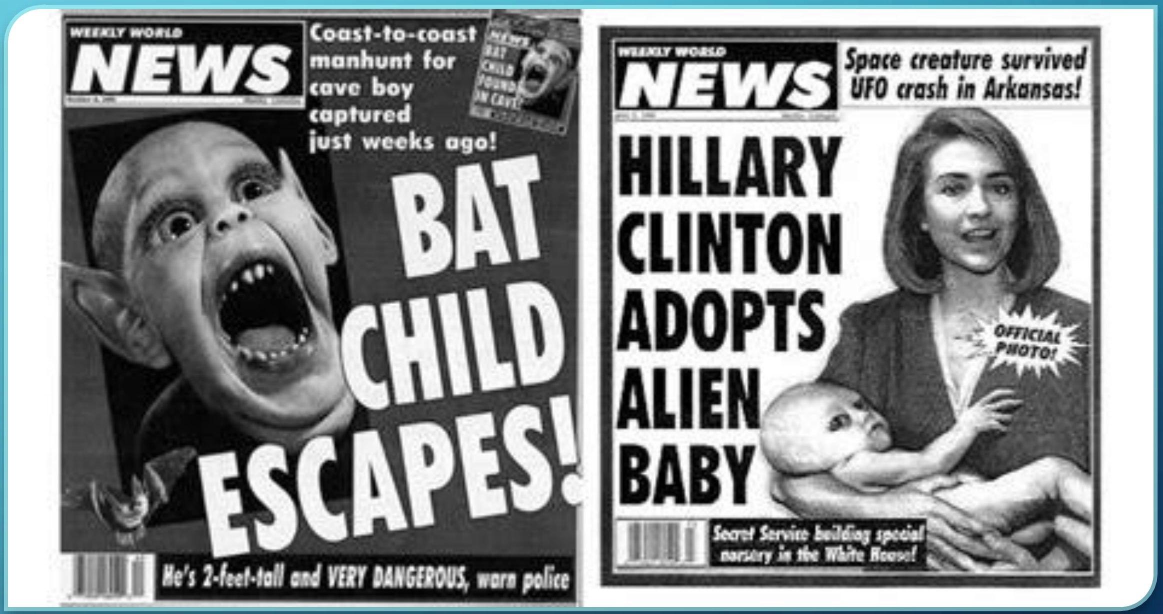
SPRINGBOARD CAPSTONE 3 PROJECT – JONNY PEARCE – JULY 2023

THE PROBLEM

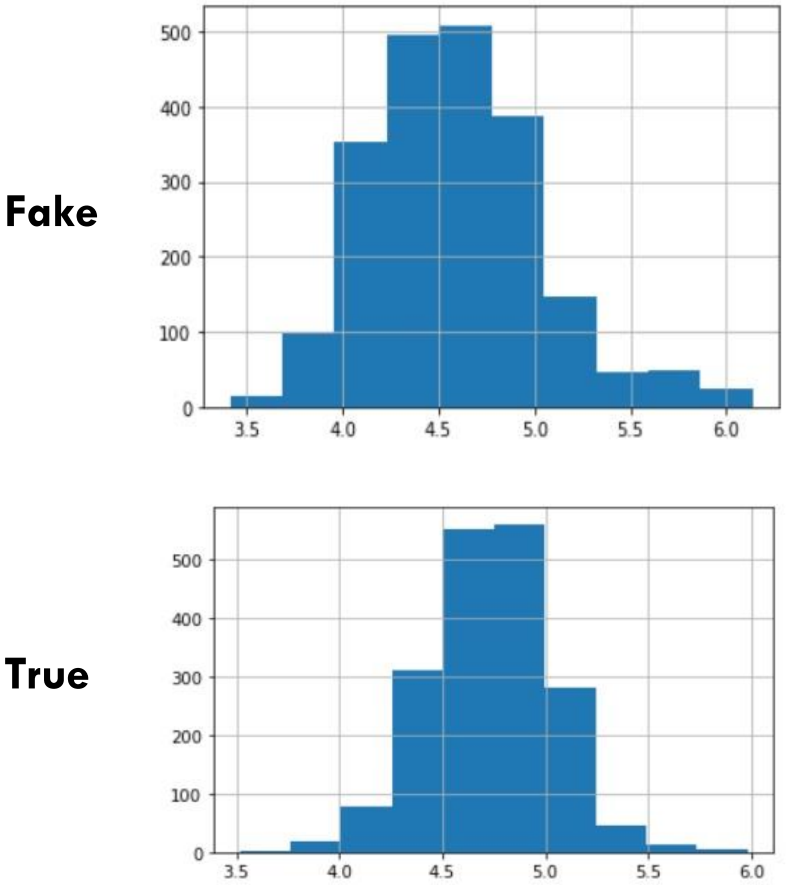
- Reliability of information key to businesses government and media around the world
- Rise of misinformation and fake news has massive impact on society as well as ability to sell products and services.
- Businesses have to cope with product/service reputation issues, along with legal/regulatory issues and legal liability risks.
- Thus knowing what is true and fake in terms of protecting your business and enhancing your corporate profile is crucial.
- Reliable and effective tools to manage fake news and false/misleading information more and more important

THE DATA

- Taken from https://www.kaggle.com/datasets/jruvika/fake-news-detection?select=data.csv
- Contains publication URLs, headlines, full content and binary ratings on over 4000 news articles
- Standardised and cleaned to:
  - Remove punctuation.
  - Remove non-alpha-numeric characters.
  - Change all the text to lower-case
  - Consider any other content issues, eg, removing website addresses, etc.
  - Drop null values

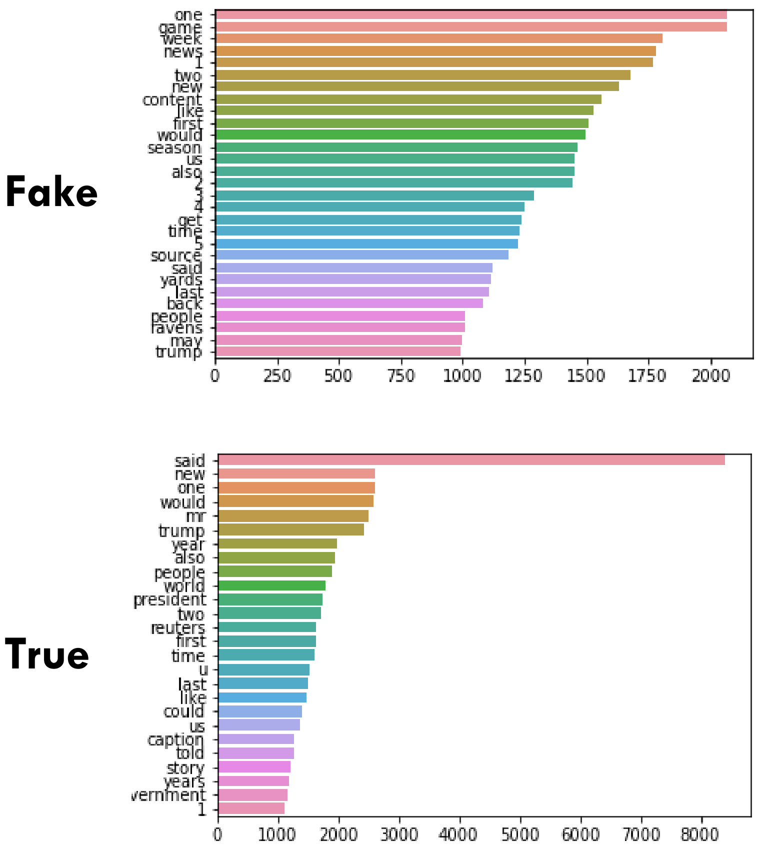


EXPLORATORY ANALYSIS (1)



Average word lengths of article for fake news and true stories subsets

- The average number of characters in fake news stories was 2395 compared to 3620 for the true stories.
- The average number of words in fake news stories was 417 compared to 618 for the true stories (and 511 for all the articles in the dataset).
- So, as a rule of thumb, the fake news stories were about one-third shorter than the true stories.
- Histograms on the average word length show that the fake news articles had a lower average word length than the true stories

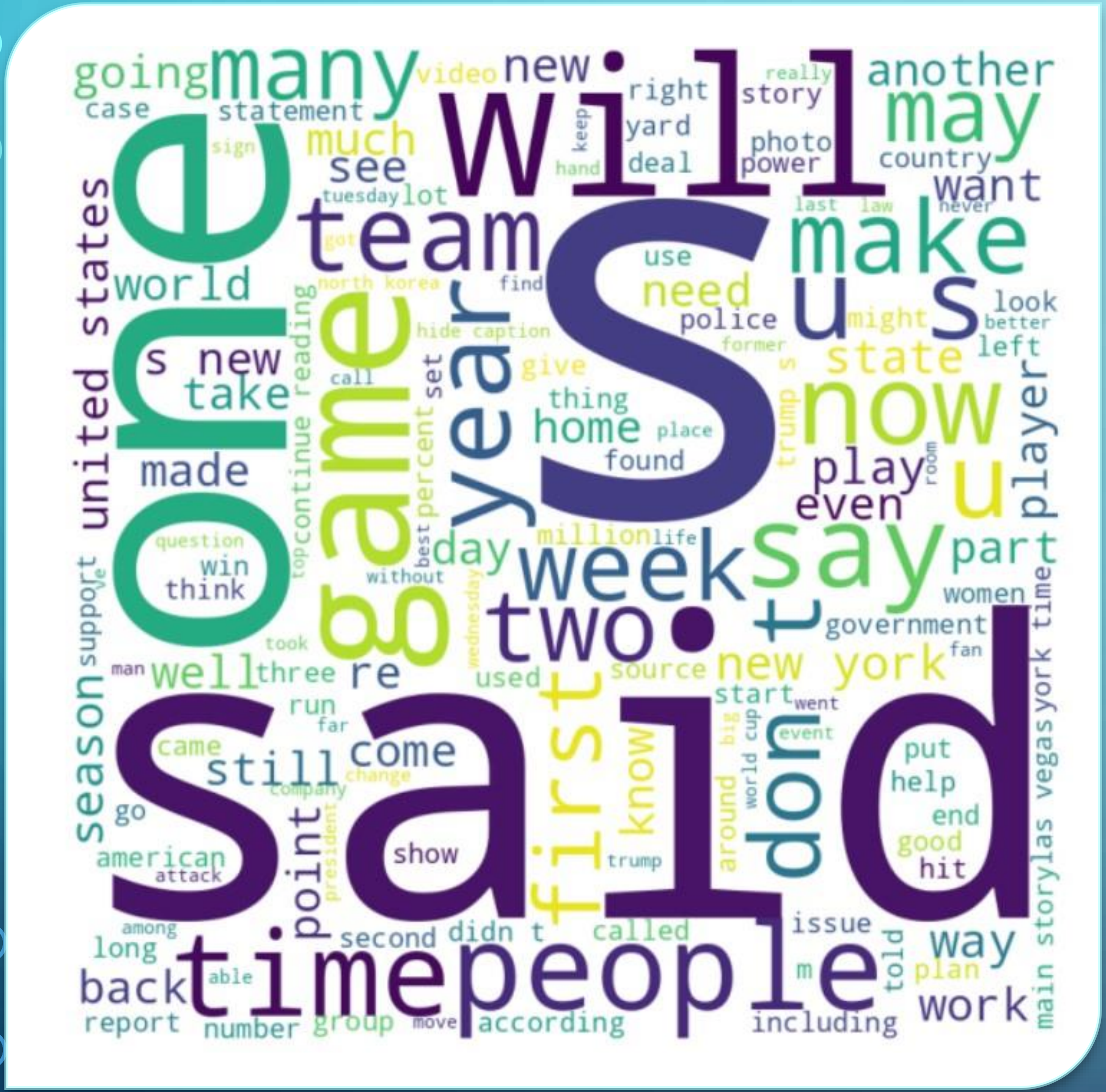


Top non-stopwords in fake news and true stories, respectively

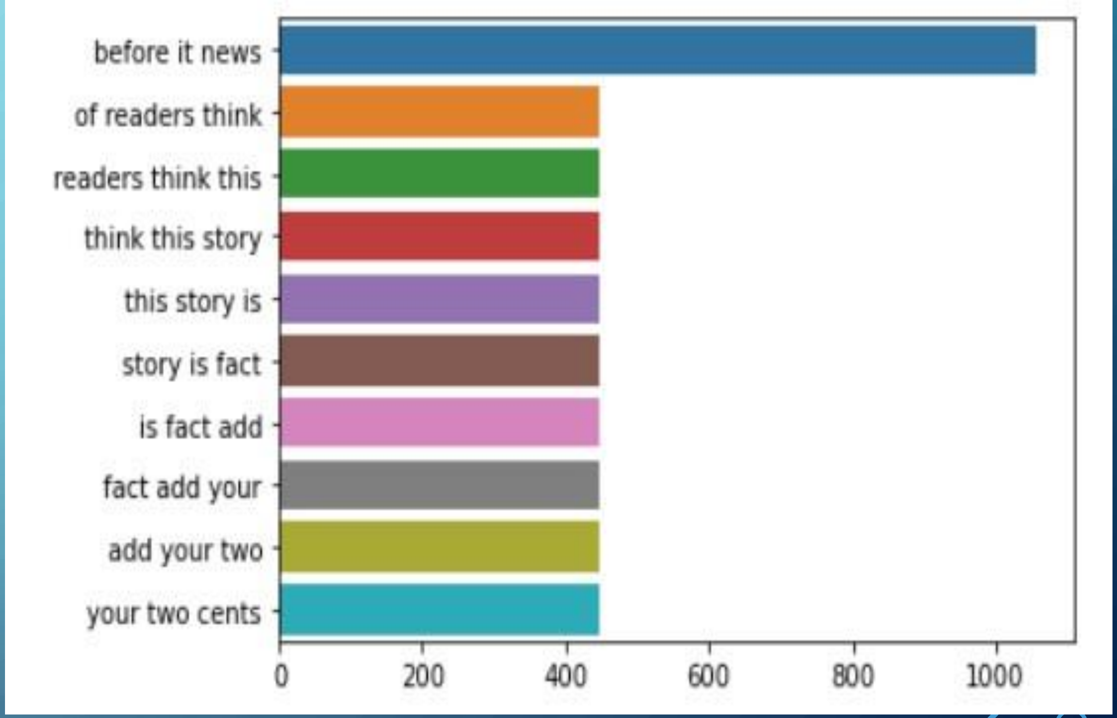
EXPLORATORY ANALYSIS (2)

- Removed stopwords
- 8,000+ uses of "said" in true stories compared to 1,000+ in fake news
- Suggests that validation, attribution and accurate citing of sources play a strong role in the true stories

EXPLORATORY ANALYSIS (3)



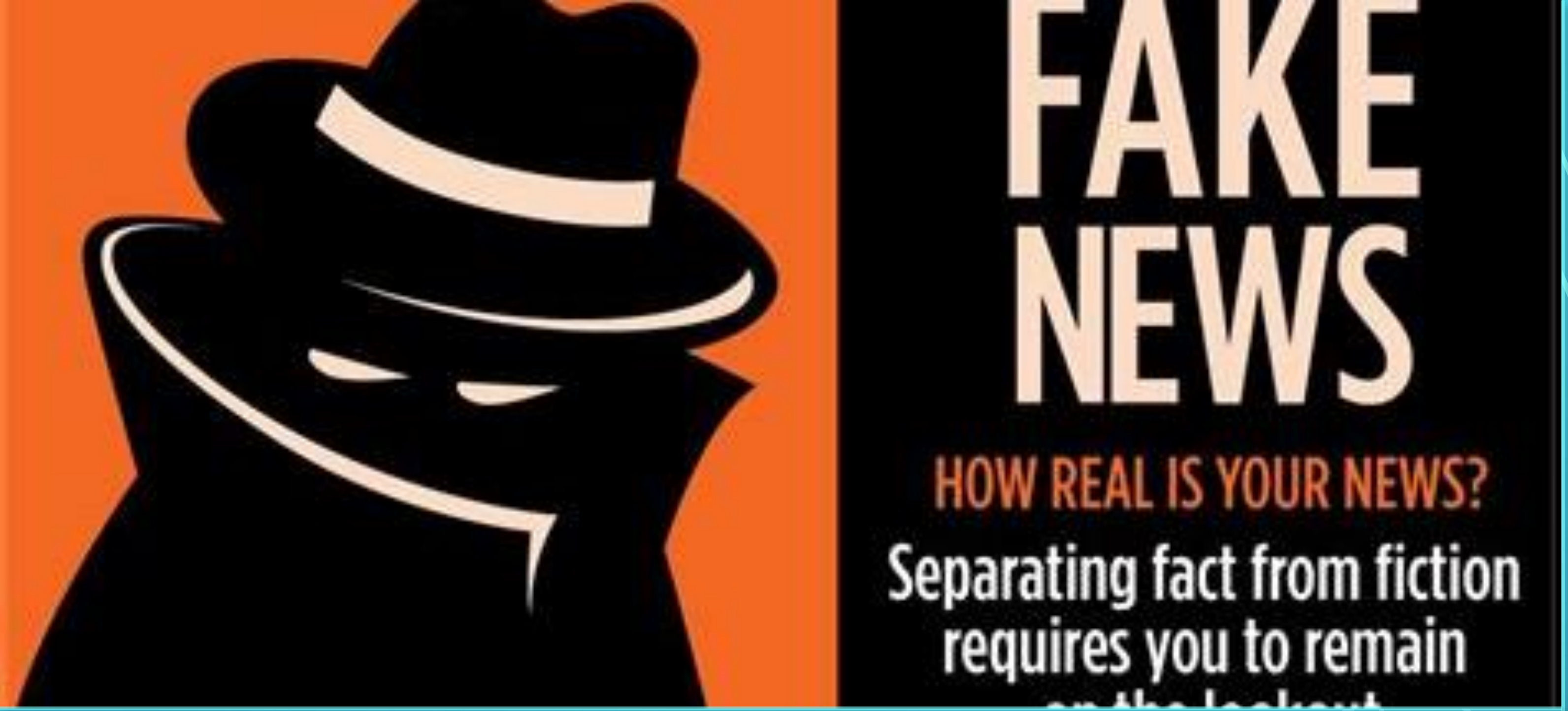
Word cloud of full dataset



Top trigrams in fake news

MODELLING AND SELECTION

- Vectorization
  - Term Frequency – Inverse Document Frequency
  - Word2Vec – didn't work very well
- Classification
  - Passive Aggressive Classifier
  - Random Forest Decision Tree Classifier
  - Logistic Regression Classifier
  - Linear Subject Vector Classifier
  - Naive Bayes Multinomial Classifier



MODEL SELECTION

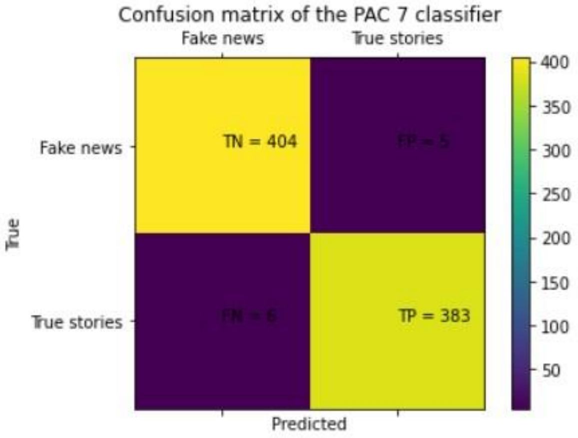
Initial results

All models had good accuracy results:

- Passive Aggressive - 98.37%
- Random Forest - 96.49%
- Logistic Regression - 97.37%
- Linear SVM - 98.5%
- Multinomial Naive Bayes - 93.86%

- However, two stood out:
- PassiveAggressive Classifier; and
- Linear SVM Classifier.

Not only were accuracy scores slightly higher, but the number of false positives (ie, the number of fakes news stories that they let through) were significantly lower than the other algorithms (6 and 8, respectively, compared to double figures for the others and sometimes into the high 20s).



Classification_report

	precision	recall	f1-score	support
0	0.99	0.99	0.99	410
1	0.98	0.99	0.99	388
accuracy			0.99	798
macro avg	0.99	0.99	0.99	798
weighted avg	0.99	0.99	0.99	798

FURTHER TESTING AND REFINEMENT

- Worked further with PAC and Linear SVM models:
    - Manual and GridSearch parameter tuning.
    - No improvement to Linear SVM
    - Some improvement to PAC model.
- Passive Aggressive Classifier final selected model

CONCLUSIONS AND NEXT STEPS

- Having tried a number of classifiers, it's clear that once you have a cleaned, standardised and vectorized dataset of articles/text, it's possible to build a passable model with many of them.
- However, only the PassiveAggressive Classifier and the Linear Subject Vector Machine Classifier were really up to the job.
- Parameter tuning offered only marginal improvements, but worth doing for the PassiveAggressive Classifier.
- In terms of word vectorization, interesting to see that TF-IFD performed well, in marked contrast to Word2Vec. May be that Word2Vec works better on much larger datasets.

- **Next steps:**
- Deploy model
- Refine with larger datasets
- Test with different categories of content

QUESTIONS?