Fishing the Fakes from the Reals

Abstract

"Fake news" are fabricated news stories in which no clear evidence is provided, making them highly inaccurate. These stories are intentionally created to misinform or deceive readers. With the growth in popularity of social media and in an era where most people get their news off the internet, fake news has become increasingly widespread, making manual fact-checking impractical. Hence, we recognised that a Machine Learning approach would be crucial in filtering out these fake news from the vast sea of information.

Data

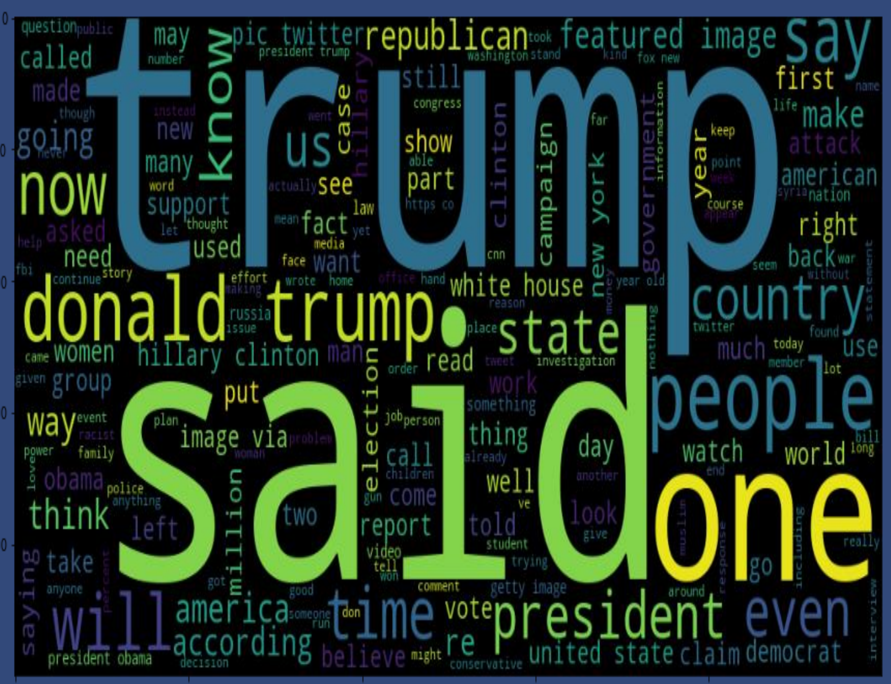
We chose to use the ISOT Fake News dataset, from the University of Victoria which contains pre-labelled true and fake news articles with title, text, subject and date. The dataset consists of 44,261 news articles altogether, with 21416 news labelled as true and 22845 others labelled as fake. Since it is from a reputable source, is a suitably large dataset and is relatively clean, we deem it reliable and feel confident to use it in our Machine Learning models.

- ### Text Preprocessing

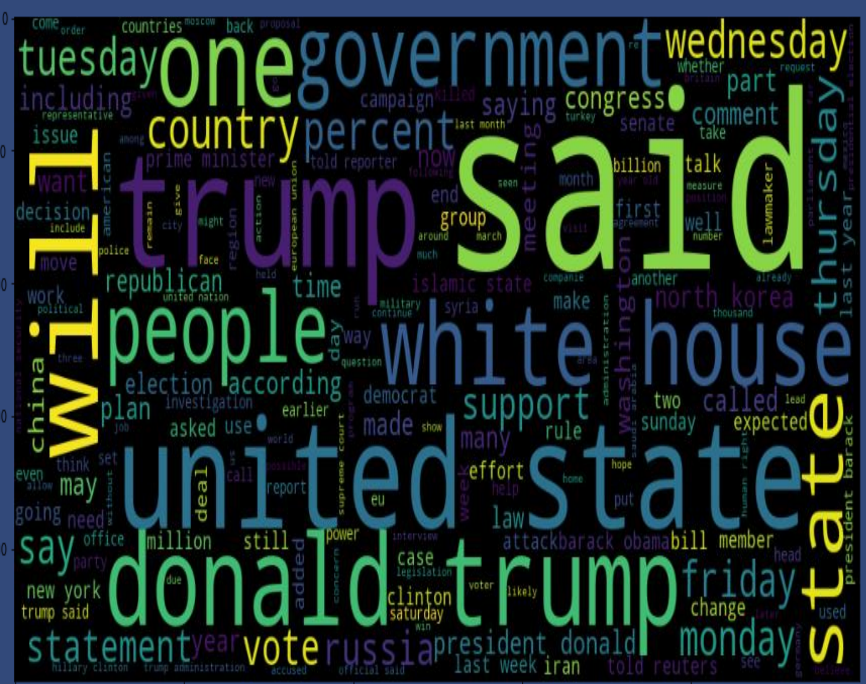
We cleaned up the data by removing rows with empty texts and used the NLTK package in Python to remove non-alphanumeric characters and stop words, thereby removing any additional noise. Additionally, all words were lemmatized and converted to the lowercase format for standardization.

- ### Text Insights

Using the cleaned dataset, we created Word Clouds to identify top common words found in fake and true news articles' texts. As observed below, majority of the dataset covers US Political News. A clear difference between the two is the vast use of the words 'trump' and 'said' in news classified as fake while the true news has quite a fair share of 'said', 'will', 'trump', 'donald trump', 'united states' and other words.



Top Common Words in 'Fake' news dataset



Top Common Words in 'True' news dataset

Text Vectorization

Process: Instantiate → Fit → Transform

TF-IDF, a frequency-based vectorizer, reflects the relevance of a word for a document in a collection of documents. The higher the metric, the more relevant the term is to a particular document (less common across documents). In this way, documents with similar words will have similar vectors, which is what we hope will help us differentiate true and fake news articles.

We also used **Word2vec**, a prediction-based vectorizer, which involves Neural Networks (NN). This means being Connectionists, and incorporating back-propagation in our model! It efficiently creates word embeddings by using surrounding words to represent the target words with a NN whose hidden layer encodes the word representations.

Metrics

Taking into account the context of our project..

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

1. <u>F1-Scores</u>
   → harmonic mean between precision & recall
   → how well the model has identified news that are true

1. <u>False Positive Rate (FPR)</u>
   → we want to avoid detecting fake news as true
   → we only want to prevent fake news from spreading, not true news!

Machine Learning Models

1. Linear Support Vector Classifier (SVC)

Linear SVC gave us a high F1-Score of **0.995**, and a low FPR close to 0.005 on article texts with TF-IDF after tuning. The performance of SVMs, commonly regarded as one of the best text classification algorithms, can be attributed to the fake news detection problem being largely linearly separable. The classification hyperplane separates the vector representations of fake news from those of true news. Here, Linear SVC supports a linear kernel only. It is faster and scales better. As these vectors are maximally separated, the model has great generalisation. It is also effective in high dimensional spaces, which is helpful with news texts because there is a large amount of words (features) in the text and title.

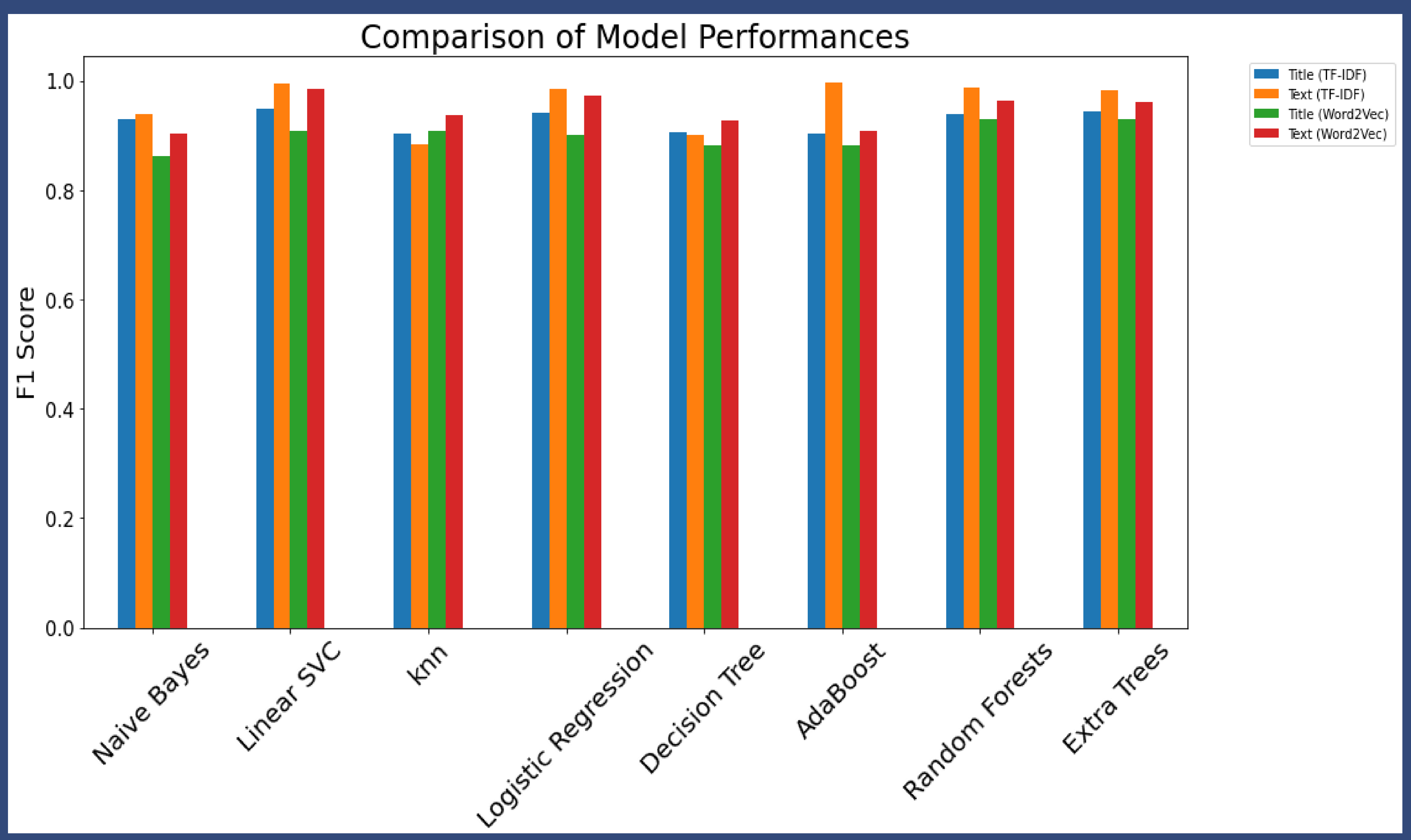
2. Ensemble Learning

Ensemble methods generally gave very high F1-Scores and FPRs. Let's look at the Decision Tree model which takes the values of the words (features) in the text and title, and splits them into groups by highest information gain, helping to discard irrelevant words. This is repeated until each group consists of a single classification - fake or true. The main issue is that it does not generalize well and has high variance with results varying from one training set to another. It is also relatively slower and is unable to capture interactions between words, leading to poorer performance due to the lack of contextualisation.

Random Forests uses a collection of Decision Trees. Rather than training tree on all data, each tree gets a random subset of the data to be trained on. It lessens overfitting in the trees and introduces randomness due to the variation across the trees thus it is more generalisable. With the most common classification as the final output, we attain a higher F1-Score for Random Forest compared to Decision Tree. So while Decision Tree is more interpretable, it isn't the best performing ensemble model.

3. k-Nearest Neighbours (kNN)

With the feature vector of a particular news, its distance from all the other feature vectors will be calculated. K nearest vectors are selected and the most common class shall be the predicted label for the news. The kNN model did not perform relatively well, compared to prior models, but still yielded a F1-Score of 0.885 and FPR of 0.094 on article texts with TF-IDF after tuning. The downside of kNN is that it doesn't work well for high-dimensional data like the case in this news data, where the number of features is large. With larger dimensions, it becomes hard for the model to calculate feature similarity and find the nearest neighbours. Calculating distances for every news we want to classify is thus very costly, rendering this model not so useful in real-time prediction.



Overall Insights

Our 8 models performed well on both titles and texts with all attaining F1-Scores above 85% after tuning. We observe that Linear SVC, Logistic Regression, Random Forests and AdaBoost perform best, with non-ensemble methods being more efficient (faster). It is worth noting that for almost all the models (with the exception of kNN and Decision Tree), running them on news text performed better than on title data, regardless of the vectorization technique used. This might be attributed to texts having significantly more words than the titles hence the models that work well with high-dimensional data learn and generalise better. Majority of our models, vectorized by TF-IDF also performed better than when vectorized with Word2vec. This could be because TF-IDF uses a scoring scheme to measure the local importance of a word occuring in the news text. Word2vec may not be able to capture this as the length of the texts may be too long. We observe that the true news dataset contains features involving the source of the news - which may be a significant factor as to why our models perform extraordinarily well. This may have caused our models to predict news with versus without source instead of true versus fake news although source can indeed be a feature of reliability.

Future Scope

Testing the models on fake and true news datasets from other sources may be a good idea to truly test the performance of the models.
We would also like to explore deep learning models such as Recurrent Neural Networks (RNN) in the future as its ability to capture sequential data makes it very appropriate for text classification due to the natural sequence in English sentences.

References

Shoemaker, E., 2019. *Using Data Science To Detect Fake News*. Undergraduate. James Madison University.

O'Brien, N. (2018). *Machine Learning for Detection of Fake News* (Master's). Massachusetts Institute of Technology.