Fake or Real: Identifying Fake and Real News

**Ong Zheng Kai**

P2004048

DAAA/2A04

Abstract

Fake news has recently been a pressing issue. Most people consume their news through the internet and knowing that anyone can post anything they want, it further promotes the spread of fake news. As a result, trying to identify if it the news is real, or fake can be almost impossible. In this project, we will be attempting to experiment with different Artificial Intelligence models and see which will best suit the job of correctly identifying if a particular news is real or fake. As this project pure purpose is to check the feasibility of applying Artificial Intelligence into identifying fake news from real news, the model context is built in the United States of America context ranging from 2015 and 2018 where there is large amount of real and fake news.

1. Introduction

In this era of technological advancements, news can easily be accessed through one’s smart phone. This convenience however is a double-aged sword. Students from Singapore surveyed 802 people aged 15 to 35 on how they consume their news and the study found that 87.3% favoured accessing news on the internet which includes news sites and social media. About 47.6% respondents preferred news source coming from the social media (Sherlyn, 2019). As news is accessed online, anyone who wishes to write an article is able to do so freely and this promotes the spread of misinformation or fake news (Pérez-Rosas, 2018).

Knowing that fake news is all around us, can we identify fake from truth? Unsurprisingly not. In Singapore, 4 in 5 people are confident in identifying fake news, however, only 10 percent of those surveyed managed to correctly separate fake news from real news (Huiwen, 2018). With this knowledge, our aim of this Artificial Intelligence (AI) project is to differentiate real news from fake news. Moreover, we will also evaluate the speed of the AI models to ensure that they not only yield the best results, but also performs the best.

Admittedly, the model we will be training will be limited as will be explained later in Section 3 – Dataset. However, this project is aimed to test the feasibility of using AI to differentiate fact from fiction. For this reason, we will be using a large dataset of fake and real news from Kaggle which so happens to originate from United States of America (USA).

1. Related Works

As fake news becomes more and more prevalent, Social Media companies are picking up the responsibility of removing these contents. As a result, companies like Facebook uses machine learning algorithms to spot fake news (Sparks and Frishberg, 2020). Twitter ensures search results about COVID19 are credible and Instagram redirects anyone searching about COVID19 to credible sources (Marr, 2020).

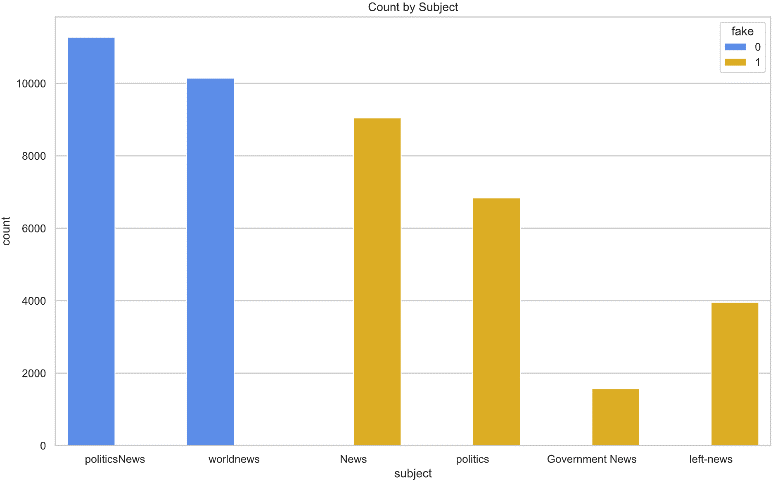
Some ongoing AI project includes the “Grover”. It is trained using 120 gigabyte worth of articles from end of 2016 to march 2018 from the top 5000 publications tracked by Google News. The “Grover” studies the style and content of the articles and builds a model of how certain phrases or styles are used in real and fake news (Devin, 2019).

Locally, there was an award-winning AI model built to detect fake news. The model was trained using 5000 Singapore news articles, 1000 international articles and 6000 fake international articles. Although 6 Fake news stories were released in Singapore, those were removed as it would affect the efficiency of the model. The aim of the model was to build an AI model with a local context (Kevin, 2019).

1. Dataset

The Dataset comes Kaggle, and it is titled “Fake and Real news datasets”. It contains two files, “Fake.csv” and “True.csv”. Most of the news in the dataset originates from the USA and is ranged between April 2015 to Mid-February 2018. This means that the model would only be able to correctly differentiate USA fake and real news dated between 2015 and 2018. Each dataset has 21417 rows amounting to a total of 42834. Each dataset contains 4 columns, news title, content, subject, and date of article.

Firstly, we will add a new column to both datasets called “fake”. If a particular news is fake, the value will be 1, and if the news is real, the value will be 0. Next, we will be combining the datasets into one single data set called “news\_data.csv”.



We first analyse the Subjects counts by the news type.

From the graph, we can tell that there is a distinct subject for the type of news. Hence, we can drop the Subject feature later.

1. Methodology

A pipeline will be made consisting of 3 main components: Count Vectorizer, Term Frequency times inverse document-frequency (Tf-Idf) and finally a model (which will be chosen later). Many steps must be put in placed as the data is in a text form and the classifiers simply cannot directly process text (Susan Li, 2018). As a result, basic pre-processing must be done before we train the classifier. Let us look at the purpose of each step.

In text analysis, there are 3 common ways to extract features from text content: tokenization, counting and normalizing. Let us also understand features and samples. A feature is each individual token occurrence frequency, whilst a sample is the vector of all token frequencies for a given document. Documents are described by word occurrences and completely ignores the position information of the words in the document (scikit-learn.org, n.d.).

Count Vectorizer readily handles 2 of the 3-extraction methods, Tokenization and Counting. There is another option which serves a similar purpose called the Hashing Vectorizer. Although the Hashing Vectorizer would take up lesser memory and faster to pickle and un-pickle, the Count Vectorizer is able to compute the inverse transform which allows us to view which features are the most important to the model.

A particularly notable parameter of Count Vectorizer we will be using is stop\_words. Stop words are words like “him”, “and”, “the” and so on. These words occur often and yet are uninformative in depicting the content of a text. Hence, they are usually removed to prevent the model from identifying these words as an important feature during classification. This would greatly improve the model accuracy (Sai, 2020) Instead of using sklearn default stop\_words list, we will be using one of the stop words list found on a GitHub repository.

The final element is normalization and to do just that, we will be using Tf-Idf. Tf-Idf is especially important to re-weight the count features into floating point values for usage by classifiers (scikit-learn.org, n.d.).

We can finally get to the modelling part. We will be testing the following models: Passive Aggressive Classifier, Support Vector Classifier, Decision Tree, Random Forest, Multinomial Naïve Bayes, Logistic Regression and Dummy Classifier which will be used a stupid baseline. To select our model, our primary mode of identifying the performance of the model shall be the accuracy.

1. Model Selection

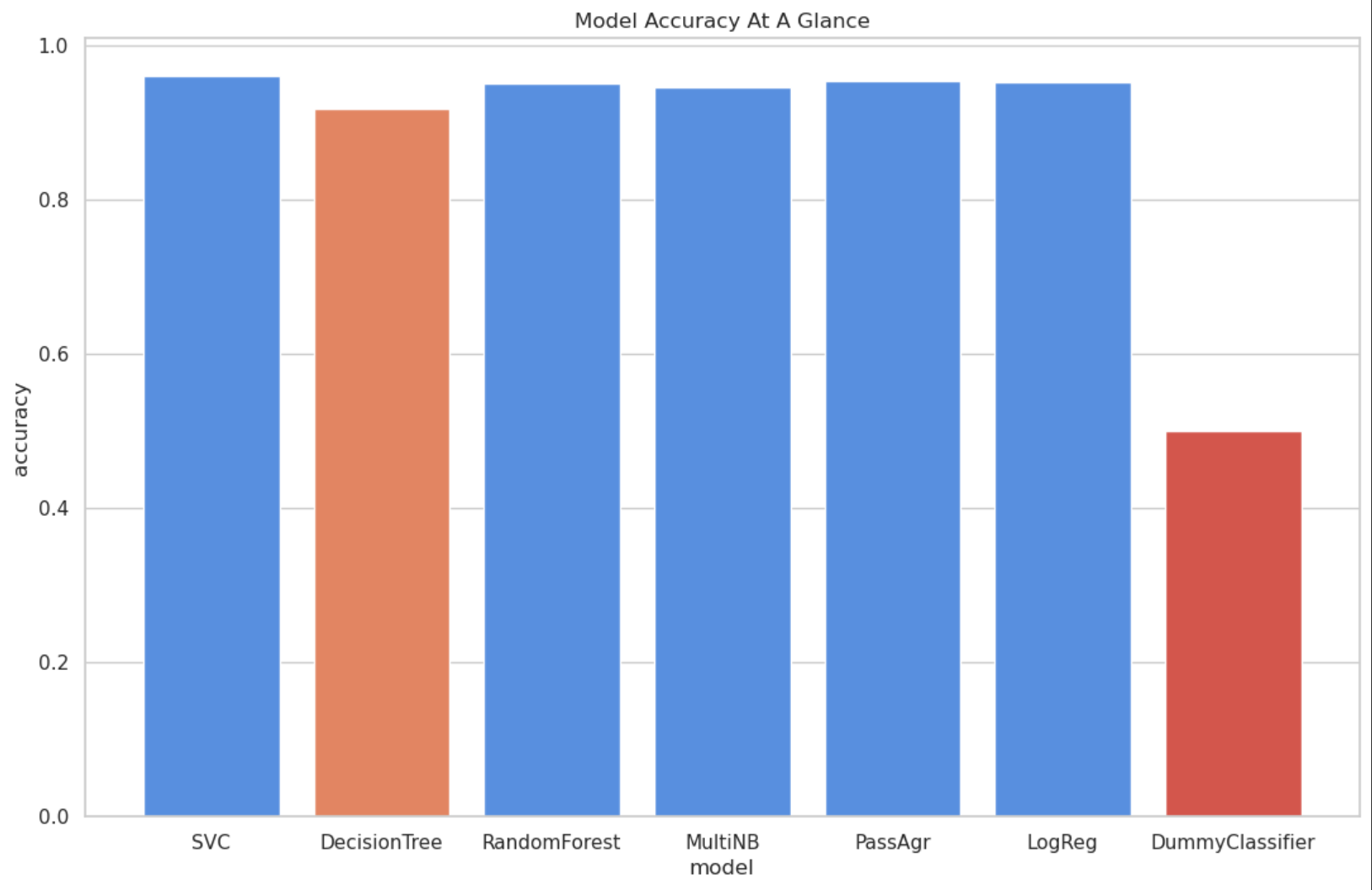
Most of the models are quite familiar with one exception – Passive Aggressive Classifier. Why should we include Passive Aggressive Classifier? Our datasets come from news networks. These news networks constantly push out articles daily or hourly and as a result, we need a model that is able predict and process an infinite amount of data (Louis, 2019). Online Learning models can serve this purpose and the Passive Aggressive Classifier is an Online Learning model. As a result, it this model is a good candidate to be tested. Although our current dataset contains a finite amount of news articles, it would be interested to give this new model a go.

As time has a part to play in this test, hardware is to be said to get a more realistic understanding of the timings. We will be using Google Cloud compute hardware configured with 32 N2 cores and 128 GB of RAM.

After testing, the results are as follow:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-score | Time (s) |
| SVC | 0.961917 | 0.962132 | 0.961917 | 0.961913 | 154.469989 |
| Decision Tree | 0.918376 | 0.918496 | 0.918376 | 0.918371 | 4.729711 |
| Random Forest | 0.950856 | 0.951407 | 0.950856 | 0.950842 | 17.201389 |
| Multinomial NB | 0.945925 | 0.946461 | 0.945925 | 0.945908 | 0.392799 |
| Passive Aggressive | 0.954329 | 0.954335 | 0.954329 | 0.954329 | 0.436747 |
| Logistic Regression | 0.953425 | 0.953747 | 0.953425 | 0.953417 | 0.696900 |
| Dummy Classifier | 0.500394 | 0.250394 | 0.500394 | 0.333771 | 0.400772 |

Let us plot the table above out:



After testing all the models, most models yield satisfactory results of accuracy scores more than 0.9. The worst performing model being Decision Tree, and the best performing model being Support Vector Classifier. As all models do not have an accuracy of 1.0, any model we will be selecting will be hyper parameter tuned for better results.

When comparing to the Dummy Classifier, we can conclude that all the models outperform the Dummy Classifier, whereby the Dummy Classifier only achieved an accuracy score of 0.501, equivalent to a random guess.

After looking at the results, we will be going with the Passive Aggressive Classifier. Why the Passive Aggressive Classifier? As all the models have a relatively close accuracy score, we have to start evaluating the models by the speed of the models. We will not be choosing Decision Tree as it has the lowest accuracy score. Although SVC has the highest accuracy, it simply takes far too long to train the model, and hence we are left with Random Forest, Multinomial NB, Passive Aggressive Classifier and Logistic Regression. Passive Aggressive classifier has the second highest accuracy score and moreover, it is faster compared to Random Forest. Hence, Passive Aggressive will be chosen.

1. Model Tuning

There are two main parameters we will be tuning, C value and max iterations. The C value will range from 100 to 0.0001, each time it will divide by 10. For max iterations, the values are ranging from 1000 to 10 000, increment of 1000. As mentioned in Section 4, we will be experimenting with the Stop Words parameter in the Count Vectorizer. There will be three different stop lists options, None, ‘English’ and stop\_list, an array of stop words found on GitHub.

After hyper parameter tuning, the best accuracy score was 0.96. The best parameter achieving this accuracy score were the following.

* C – Value: 0.1,
* Max Iterations: 3000
* Stop Words: None

An interesting parameter that was slightly shocking was the stop words parameter. This was surprising as I expected the presence of a stop list will give the model a better accuracy score.

Overall, the accuracy has indeed increased by about 1%. Let us test the model using these new parameters.

1. Understanding the Model

Now that we have a trained model, let us look at what are the key features which contributes to the model decisions in identifying if a news article is real or fake?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Top 10 tokens likely to appear in real news: | | |  | | Top 10 tokens likely to appear in fake news: | | |  | |
|  | | |  | |  | | |  | |
| Token | Weight |  | | Token | | Weight |  | |
| Factbox | -7.006539 | busted | | 4.383711 |
| says | -6.557875 | dem | | 4.515762 |
| urges | -4.468491 | whoa | | 4.539831 |
| rohingya | -4.432834 | racist | | 5.003172 |
| exclusive | -3.886221 | hillary | | 6.243774 |
| myanmar | -3.766333 | just | | 6.514467 |
| seek | -3.607736 | watch | | 7.747325 |
| kremlin | -3.582149 | gop | | 8.434112 |
| britain | -3.561239 | breaking | | 8.812951 |
| tillerson | -3.510540 | video | | 15.801865 |

On examination, the top tokens in their respective news types do make sense and seem to be in the right place.

1. Discussions

As seen from the above experiment, we are to achieve satisfactory results even from a relatively small data set of about 42 000 rows. However, this model is just the tip of the iceberg of what AI can do in the fight against misinformation and fake news. If such a model were to be deployed, the model will be trained every single day with credible news sources and keep itself up to date with the latest news.

However, the use of AI is limited by the data it is trained on. If we were to attempt to pass a Singapore news article for example through the model, it might not get as accurate results as we hoped for. As a result, we might require more advanced methods such as contextual understanding. Another approach is to create another AI model that is trained to produce fake news in local context. The model used to detect if a news is fake or not will then be tasked to identify the articles produced by the model which produced fake news articles as fake.

Such an approach however should be done with caution as if the fake news articles produced by the AI model were to be leaked, it might cause some issues. As a result, this approach needs to be done in a closed-door system which might possibly decrease the expandability of the model.

1. Conclusion

In conclusion, an AI model used to predict if a news article is very feasible to be carried out. To improve the model, it must be trained to understand context, and constant data needs to be fed into the model. We also found that Passive Aggressive classifier, an Online model is indeed a good candidate for this task of deciphering real news from fake.

We also learned that models need to be trained with news from all around the world to allow the model to understand the local context.

1. References

Sherlyn Seah, 2019. ‘Youths who rely on social media for news less knowledgeable about current affairs than those who read news sites, apps’, *Today*, January 15, accessed 4/6/2021

<<https://www.todayonline.com/singapore/youths-who-rely-social-media-news-less-knowledgeable-about-current-affairs-those-who-read>>

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea, 2018. ‘Automatic detection of fake news’, *Association for Computational Linguistics*, August, accessed 4/6/2021

<<https://www.aclweb.org/anthology/C18-1287/>>

Huiwen Ng, 2018, ‘4 in 5 Singaporeans confident in spotting fake news but 90 per cent wrong when put to the test: Survey’, *The Straits Times*, September 27, accessed 4/6/2021

<<https://www.straitstimes.com/singapore/4-in-5-singaporeans-confident-in-spotting-fake-news-but-90-per-cent-wrong-when-put-to-the>>

Hannah Sparks, Hannah Frishberg, 2020 ‘Facebook gives step-by-step instructions on how to spot fake news’, *New York Post*, March 26, accessed 4/6/2021

<<https://nypost.com/2020/03/26/facebook-gives-step-by-step-instructions-on-how-to-spot-fake-news/>>

Bernard Marr, 2020, ‘Coronavirus Fake News: How Facebook, Twitter, And Instagram Are Tackling The Proble’, *Forbes*, March 27, accessed 4/6/2021

<<https://www.forbes.com/sites/bernardmarr/2020/03/27/finding-the-truth-about-covid-19-how-facebook-twitter-and-instagram-are-tackling-fake-news/?sh=74ad9a861977>>

Devin Coldewey, 2019, ‘To detect fake news, this AI first learned to write it’, *Tech Crunch*, June 11, accessed 4/6/2021

<<https://techcrunch.com/2019/06/10/to-detect-fake-news-this-ai-first-learned-to-write-it/>>

Kevin Kwang, 2019, ‘RGS student develops award-winning AI-based fake news detector’, *Channel News Asia*, April 18, accessed 4/6/2021

<<https://www.channelnewsasia.com/news/technology/rgs-student-develops-award-winning-ai-fake-news-detector-11458402>>

Susan Li, 2018, ‘Multi-Class Text Classification with Scikit-Learn’, *towards data science*, February 19, accessed 4/6/2021

<<https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f>>

Scikit-Learn.org, Section 6.2 Feature extraction, accessed 4/6/2021

<[https://scikit-learn.org/stable/modules/feature\_extraction.html#text-feature-extraction](https://scikit-learn.org/stable/modules/feature_extraction.html%23text-feature-extraction)>

Sai Teja, 2020, ‘Stop Words in NLP’, *Sai Teja*, June 10, accessed 4/6/2021,

<<https://medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47>>

Louis de Benoist, ‘Incremental (Online) Learning with Scikit-Multiflow’, *towards data science*, December 26, accessed 4/6/2021

<<https://towardsdatascience.com/incremental-online-learning-with-scikit-multiflow-6b846913a50b>>