Written Report

Project Title: Armitage (Group 9-09)

Or, the implementation of machine learning and natural language processing to detect fake news

Leader:
Oak Soe Khant (4i2 20)

Group Members:
Tan Yi Jun (4A1 26)
Tan Jun Hao Enzo (4P1 29)

Abstract

The progressively prevalent problem of the propagation of fake news in recent years has picked up pace, and the need for a tool to automatically detect fake news has only become more pronounced in the present-day. Armitage aims to partially rectify this by providing conscious consumers of the media with a tool to more easily conduct automated fact checking of news sources, utilising a simple yet effective machine learning architecture to detect if news is fake or true. Our model is based on knowledge-based detection of fake news and relies on referencing with other established sources in order to come up with a prediction of the truthfulness of a news article or statement.

Contents

Section 1: Introduction

1.1 Thesis Statement

Armitage is a fake news detection program aimed at providing consumers of media with a tool to automate the process of verifying the accuracy of a source or statement, based upon the principles of machine learning and natural language processing in order to create a simple yet effective and computationally inexpensive stance detection-based model. This tool is meant to allow users to more easily fact check statements and news articles by referencing from a database of established online sources.

1.2 Rationale

The recent trends of digitisation and the popularity of social media platforms as a channel for the propagation of news has made fake news an increasingly serious problem in

today's modern world. While various measures have been proposed to combat fake news, these measures have proven mostly insufficient and lackluster against the waves of false information circulating online. Fake news, unfortunately, is growing more prevalent by the day. According to a global study conducted in 2019 by Statista, 62% of respondents felt that there was a fair extent of or great deal of fake news on online websites and platforms. By comparison, 10% less said the same about TV, radio, newspapers, and magazines, marking out traditional media in general as being more trustworthy than online formats. We can see that in all forms of media, over half of the viewers agreed that fake news was present, which is certainly a worrying statistic.

One area commonly associated with fake news is politics. The potential impact of fake news on then President Trump's victory in 2016 was among the most contested controversies of his presidency. Looking at some statistics, we saw that of the known fake news stories that appeared in the three months before the election, those favoring Trump were shared a total of 30 million times on Facebook, while those favoring Clinton were shared 8 million times. More worryingly, just over half of average American adults who recall seeing fake news say they believed the stories. Now, this could have contributed to the number of people in favour of Trump being more than those for Clinton, especially when articles like "Pope Francis Shocks World, Endorses Donald Trump for President" are getting almost 1,000,000 engagements right before Election Day. It is hence not hard to see how election outcomes could be swayed by fake news.

1.3 Focus & Significance of Project

Our project is targeted at news articles that warrant a certain suspicion upon reading about whether they are genuine or fake. It serves as a tool to aid any readers with these suspicions to ascertain whether these news articles are likely to be fake news or genuine news. By allowing readers to find out if these news articles should be believed or not, we can hope to achieve less fake news being shared, allowing people to come across less

fake news and be duped less, thus making sure that those vulnerable to believing fake news don't fall for such information.

1.4 Scope of Project

This project is

Section 2: Literature Review

2.1 Existing use of Natural Language Processing (NLP)

The Natural Language Processing (NLP) rating of an algorithmic system enables the combination of speech understanding and speech generation and can be utilised to detect actions with various languages. It may be an ideal system for extraction actions from languages of English, Italian and Dutch speeches through utilizing various pipelines of various languages such as Emotion Analyzer and Detection, Named Entity Recognition (NER), Parts of Speech (POS) Taggers, Chunking, and Semantic Role. The Sentiment analysis uses dual languages Resources for analysis: Glossary of meaning and Sentiment models database. for constructive and Destructive words and attempts to give classifications on a level of -5 to 5. Parts of speech taggers tools for languages such as European languages are being explored to produce parts of language taggers tools in different languages such as Sanskrit, Hindi and Arabic.

Z Khanam (2021) *"Fake News Detection Using Machine Learning Approaches"* Retrieved from: https://iopscience.iop.org/article/10.1088/1757-899X/1099/1/012040/pdf

The World Wide Web contains data in diverse formats such as documents, videos, and audios. News published online in an unstructured format (such as news, articles, videos, and audios) is relatively difficult to detect and classify as this strictly requires human

expertise. However, computational techniques such as natural language processing (NLP) can be used to detect anomalies that separate a text article that is deceptive in nature from articles that are based on facts. Other techniques involve the analysis of propagation of fake news in contrast with real news. More specifically, the approach analyzes how a fake news article propagates differently on a network relative to a true article. The response that an article gets can be differentiated at a theoretical level to classify the article as real or fake. A more hybrid approach can also be used to analyze the social response of an article along with exploring the textual features to examine whether an article is deceptive in nature or not.

I. Ahmad , M.Yousaf, S. Yousaf , and M.O Ahmad (2020,Oct 17) *"Fake News Detection Using Machine Learning Ensemble Methods"*
Retrieved from: https://www.hindawi.com/journals/complexity/2020/8885861/

2.2 Accuracy of multilayer perceptron (MLP) and Term Frequency (TF)

Dataset was analysed using linguistic properties of text such as term frequency (TF) and term frequency-inverse document frequency (TF-IDF) as a feature set, and a multilayer perceptron (MLP) classifier is used with one hidden layer and a softmax function on the output of the final layer. The dataset contained articles with a headline, body, and label. The system's accuracy on the "disagree" label on test examples was poor, whereas it performs best with respect to the "agree" label. The authors used a simple MLP with some fine-tuned hyperparameters to achieve an overall accuracy of 88.46%.

I. Ahmad , M.Yousaf, S. Yousaf , and M.O Ahmad (2020,Oct 17) *"Fake News Detection Using Machine Learning Ensemble Methods"*
Retrieved from: https://www.hindawi.com/journals/complexity/2020/8885861/

Section 3: Methodology

3.0 Tools And Languages Used

The main language much of the programming was done in was Javascript; this was because it was the script the team was most familiar with. The runtime environment used was Node.js; and libraries like Tensorflow.js were used, mainly for the purposes of performing lower level mathematical operations needed for text vectorisation and feature extraction.

Fig 3.0: A brief overview of the tools used. Not included are HTML and CSS.

3.1 Overview of Proposed Method

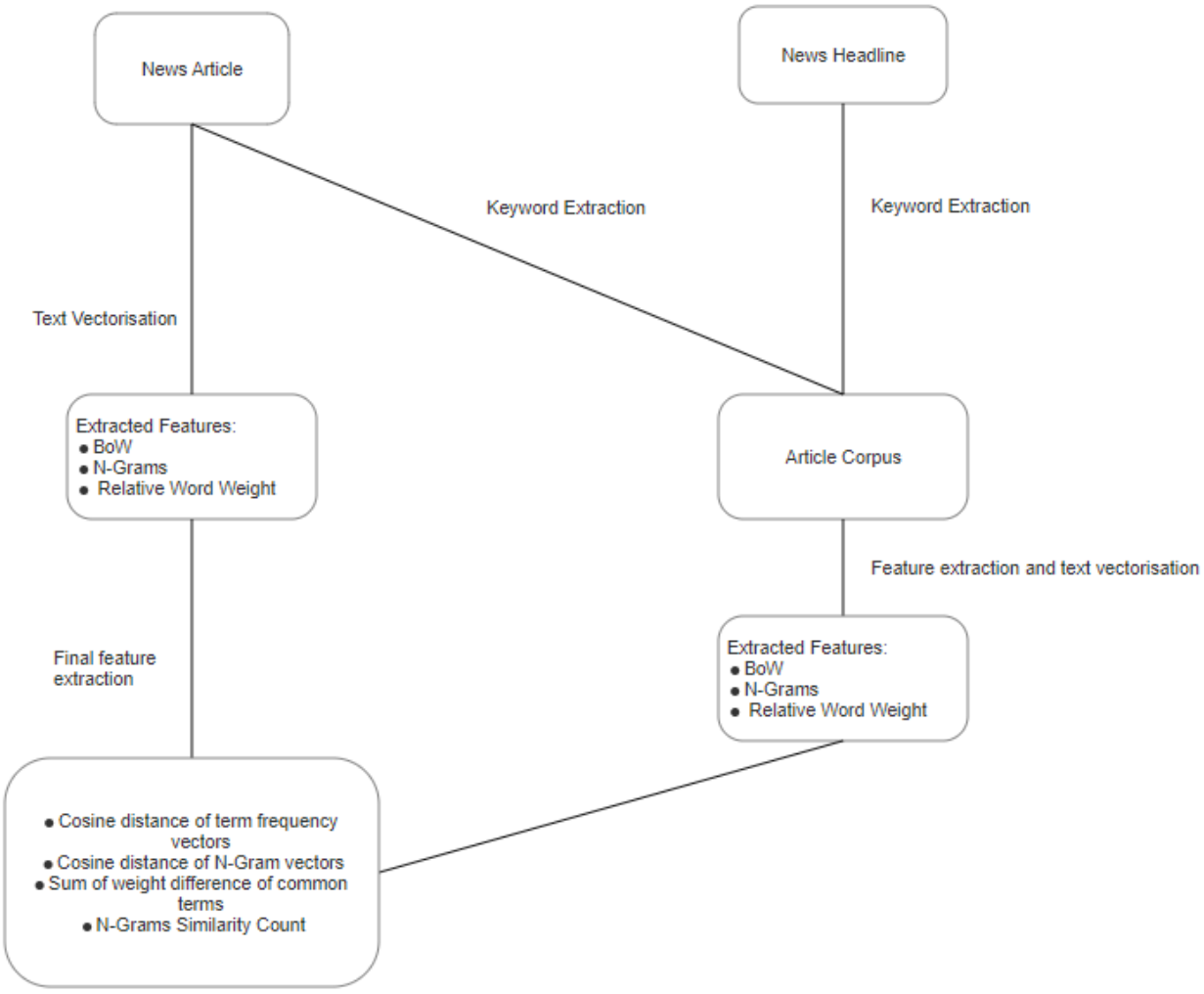

Fig 3.1: The feature extraction process.

Our model was based on identifying fake news through referencing an article, with or without a headline, with a variety of sources users could select to see if a source was reliable. For the purposes of training the model, the model would extract keywords from the headline and article based on term frequency (after part-of-speech tagging, lemmatization, and stopword removal), and use these keywords to scrape for related

articles from the web. This would allow a reference corpus of articles from reliable sources to be collected, so that further feature extraction could occur.

The following news sites and sources were, after extensive research into media bias, chosen to be the reference sources, due to their proven and documented low bias, extensive coverage of news issues, and lack of a paywall:

Table of Reference Sources

News Site	**Description & Reason for Selection**
National Public Radio (NPR)	An American privately and publicly funded non-profit media organization based in Washington, D.C Reason for selection: Credible source with a wide variety of news sources.
CNN	A multinational news-based pay television channel headquartered in Atlanta, United States. Reason for selection: A wide selection of news articles with many differing authors with differing opinions and views.
BBC	The national broadcaster of the United Kingdom. Reason for selection: Mainly neutral writing of news sources, also a large variety of articles to choose from.
CNBC	An American pay television business news channel. Reason for selection: Little bias in reporting and big range of news sources.

CNA	An English-language news channel based in Singapore. Reason for selection: Local opinions on local and foreign matters, many types of news articles.
The Straits Times	An English-language daily broadsheet newspaper based in Singapore. Reason for selection: Main news of Singapore, wide selection of news articles, unbiased and factual
CBS	The news division of the American television and radio service CBS. Reason for selection: Big news service of America and covers many topics.

Table 3.1

After that, several processes were carried out on both the reference corpus and article in order to extract the necessary features, after text preprocessing was carried out.

3.2 Data Collection and Text Preprocessing

Data Collection

Initially, we collected our data from the Fake News Challenge Stage 1 (FNC-1) data set. However, since the FNC-1 data set was meant for a stance detection model, we decided to instead opt for the ISOT Fake News Dataset. The ISOT Fake News Dataset is a compilation of several thousands fake news and truthful articles, obtained from different legitimate news sites and sites flagged as unreliable by politifact.com. In total, there are 21417 real articles and 232481 fake articles in the ISOT dataset. During actual training and testing, a total of 7000 samples from the original dataset were used; this was

due to limits in computing power of our devices and the fact that the training process, due to the way our model was designed (to include automatic web scraping), required long amounts of time to carry out processing of features. Around 658 samples were taken out for the purposes of testing; the remaining remained for training and validation purposes.

Extracted from the data were the headlines, bodies, and truthfulness (nature) of the article in question. One hot encoding was used to label fake and true news samples; fake news samples were labeled as “[1, 0]” and truthful news samples were labeled “[0, 1]”.

Web Scraping

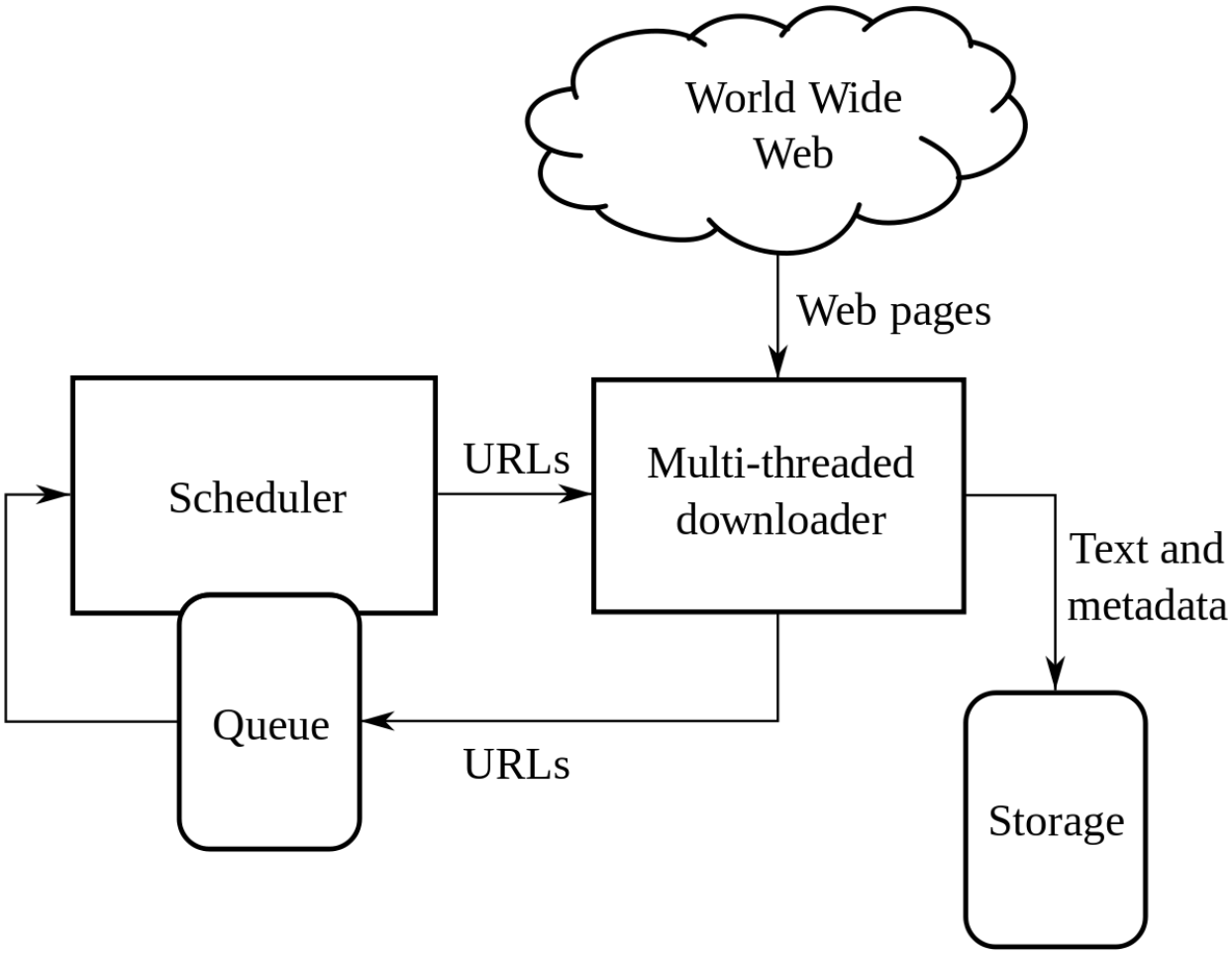

Fig 3.2, showing how a typical web crawler might work. Ours was more limited in scope and scale but functioned under similar principles.

Web crawling is done using a simple yet effective program built from scratch. First, keywords are extracted from the headline and article (see below); then, a request is sent to the various news sources listed in Table 1 with attached keywords. The most relevant 10 articles are thus scraped from each source, and all the content is combined into one master corpus containing an array of article bodies.

The web crawler was created from scratch, and due to the team's lack of experience with creating such programs, the webcrawler was rudimentary. It functions by searching up keywords with the name of a news source appended at the ends of the query via Google; after which it scrapes all the links and puts them through a filter to select the most relevant articles and links based on the words in the link and the punctuation and unique features of the news site (that had to be found through observations and testing). After which, the DOM element is extracted and run through a filter to isolate the relate to tags with classes.

Text Preprocessing

Both the headline and article were tokenised to give individual sentences, before each of the sentences were further tokenised. Each of the sentences were then put through a function to carry out part of speech tagging, so that lemmatisation of the words could occur; this was necessary to prevent words of the same meaning but different grammatical states (such as plural and singular nouns and verbs, or verbs in the different tenses) being marked as different terms and thus falsely affecting the term frequency vectors. A similar process is applied to the articles in the reference corpus.

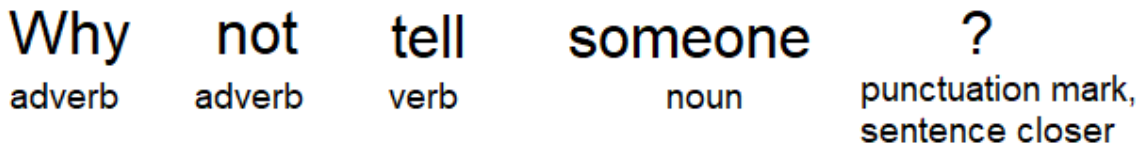

Fig 3.2, demonstrating part of speech tagging. The POS-tagging was carried out by function utilising a Hidden Markov Model.

After this, stopword removal is carried out and the text has been processed and is ready for vectorisation using term frequency analysis. On top of this, keywords are extracted from the headline and top weighted words from the input article are plugged into our web crawler for the purposes of finding articles to add to the reference list (see above section).

Term Frequency Analysis

Term frequency analysis is done via several methods, so as to represent frequency and thereby importance of a word in 3 ways.. First, the total count of a term appearing in a body of text (both in reference corpus and input article) are kept track of; second the percentage of the overall non-stopword terms which comprises of said term, and third, the relative frequency, given by the equation:

$$RF_i = \frac{f_i}{f_{max}},$$

Equation 3.1, showing formula for calculating relative frequency

Where RF is the relative frequency of a term, f_i is the term frequency of a given term, and f_{max} is the maximum term frequency.

3.3 Feature Extraction

Table of Features

Feature	Description
Cosine similarity of reference corpus vector and article vector	Text from the reference corpus and article had both initially been vectorised using term-frequency analysis. The cosine similarity of these vectors were then found by the following equation: $$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$ **Equation 1.1: Finding the cosine similarity of 2 vectors.** In this case, A and B are the article and reference corpus relative term frequency vectors. The larger the value of the cosine similarity, the more similar the 2 texts are. This is a method frequently used in document similarity analysis, as term-frequency vectors are typically very long and sparse (i.e., they have many 0 values), and traditional distance measures do not work well for such sparse numeric data.
Average of common N-Gram Weights between reference corpus and input article (w.r.t input article)	N-Grams were extracted from both the reference corpus and input article after text preprocessing. After which, similar N-Grams found in both the reference corpus and the input article had their overall weights summed together. This value was obtained by summation of the weights (relative frequencies) of the terms inside the N-Grams. Then, the average weight of common N-Grams were found.

	$W_{avg} = \frac{1}{M} \sum_{i=1}^{M} \left(\sum_{j=1}^{L_{N_i}} x_j \right)$ Where x_j is the weight of a term occurring in one N-Gram w.r.t input article, *M* is the number of similar N Grams, and *L* is the length of one N-Gram.
Average difference in common N-Gram weights between reference corpus and input article	This was found by taking the average of the difference in overall weight of similar N-Grams between the reference corpus and input article. $\Delta W_{avg} = \frac{1}{M} \sum_{i=1}^{M} \left(\sum_{j=1}^{L_{N_i}} \left\| x_{ij}^{(ref)} - x_{ij}^{(in)} \right\| \right)$ Where $x_{ij}^{(ref)}$ is the weight of a term occurring in one similar N-Gram w.r.t. the reference corpus, $x_{ij}^{(in)}$ is the weight of a term occurring in one similar N-Gram w.r.t. the input article, *M* is the number of similar N Grams, and *L* is the length of one N-Gram.

Table 3.2, showcasing all the features used for training

3.4 Model Architectures

Training of the data set was carried out with 3 models: a random forest classifier, a simple logistic regression model, and a Multilayer Perceptron (MLP) model. This was done to see which model would yield the most accurate results, from which the best would be picked.

<u>Random Forest Classifier</u>

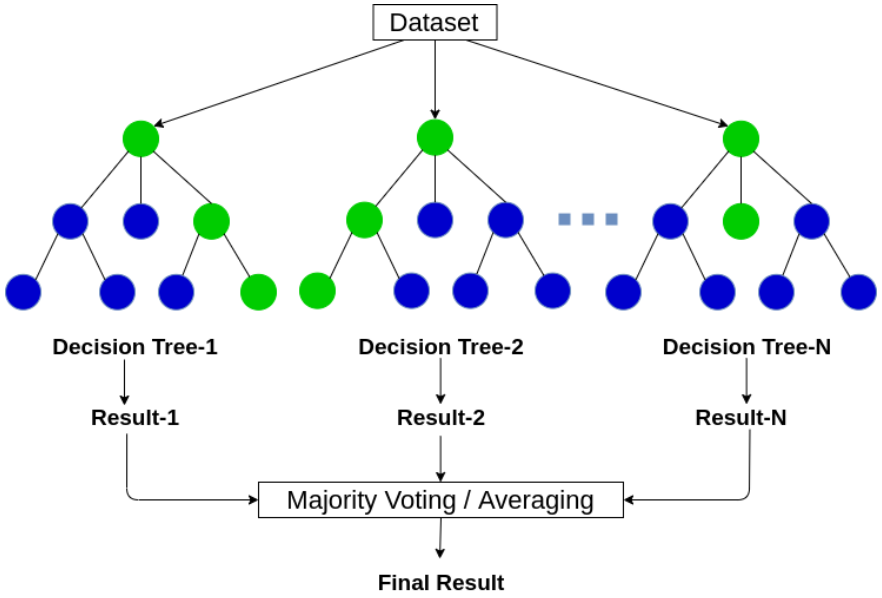

Fig 3.3, showing the basic architecture of a random forest classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent model.

Technical Details (Hyperparameters):

Hyperparameter	Description	Value
N-Estimators	The number of trees to be built before taking the maximum voting . A higher number of trees generally gives a better classification result at the expense of the training speed.	200
Max Features	The maximum number of features a classifier is allowed to try in an individual tree. A higher number of trees generally increases model	0.8

	accuracy, although this may not necessarily be the case.	
Max Depth	Represents the depth of each tree in the forest. The deeper the tree, the more splits it has and it captures more information about the data.	32

Table 3.3: Hyperparameters which were toggled in training the random forest classifier

Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. For our model, a simple binary logistic regression classifier was used.

A standard logistic function, a sigmoid function, is defined as:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Equation 3.2: Logistic regression function

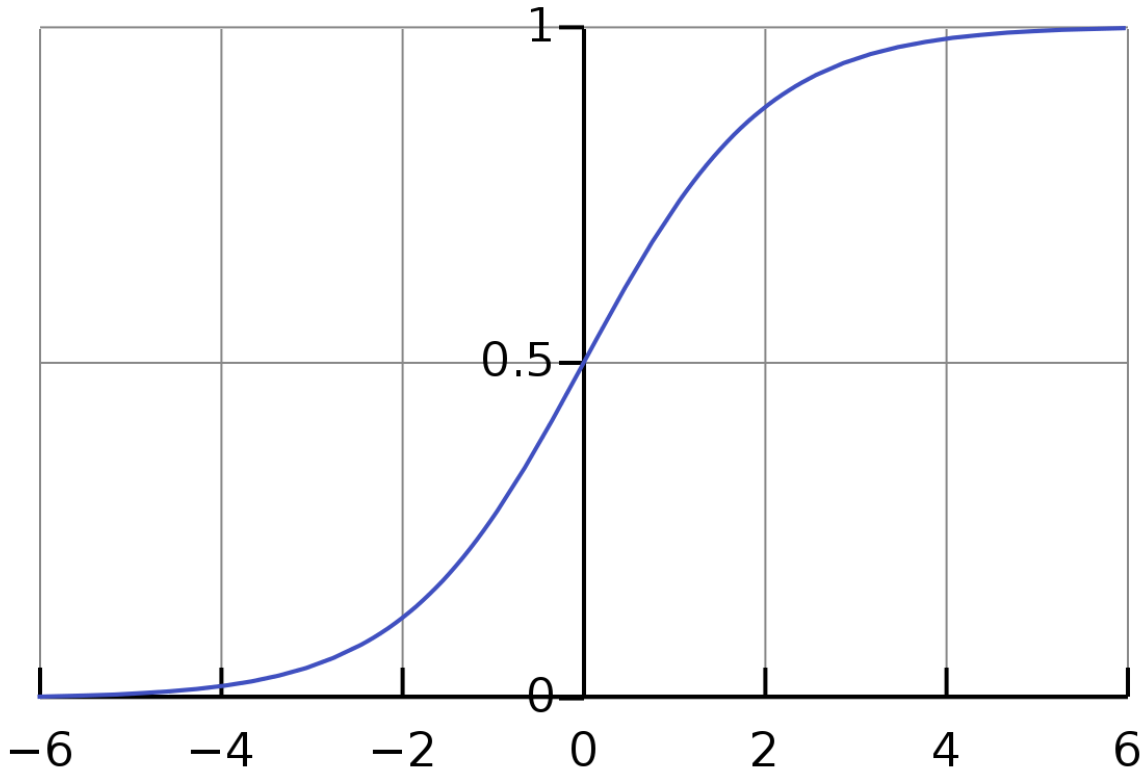

Fig 3.4 A graph of a sigmoid function

The cost function used was Binary Cross Entropy:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Equation 3.3: Binary Cross Entropy

Where *P(yi) is the predicted probability of the event being true (in this case, the event is the news being true), and yi is the actual probability (either a 1 or 0 value).*

Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is an artificial neural network, with an input layer, one or more hidden layers, and an output layer and can be fine-tuned with various parameters and number of layers to generate an optimum predicting model. A basic multilayered perceptron model with one hidden layer can be represented as a function as shown below:

$$f(x) = g\left(b^{(2)} + W^{(2)}\left(s\left(b^{(1)} + W^{(1)}x\right)\right)\right).$$

Equation 3.4

For our MLP architecture, through extensive testing and analysis, we found that an MLP with a single dense layer containing 10 nodes, with ReLU activation, provided the highest accuracy. ReLU activation was preferred over other activation functions, such as tanh or sigmoid, as during testing we found this to give the best results.

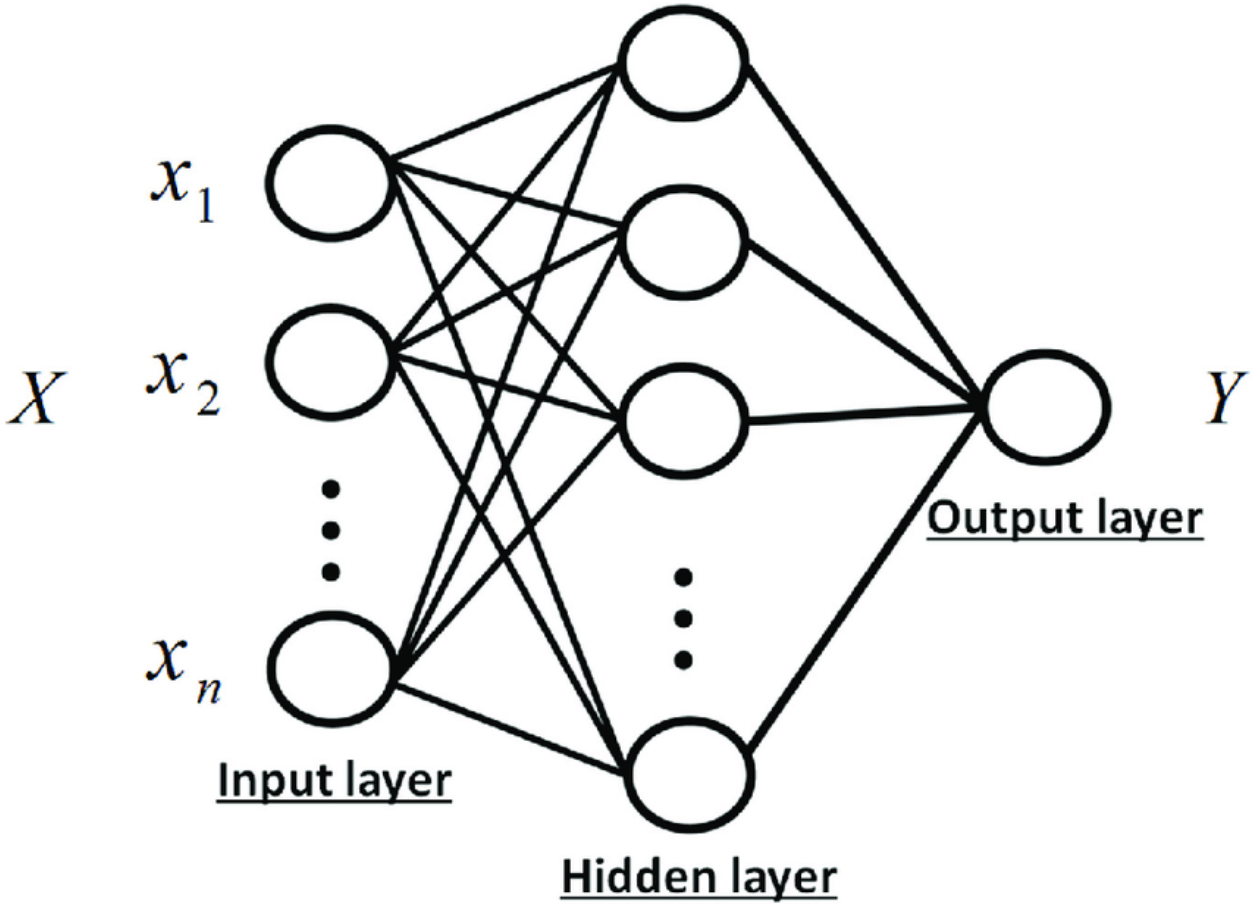

Fig 3.5, showing the architecture of a typical Multilayer Perceptron

Section 4: Results and Analysis

4.1 Confusion Matrices

Logistic Regression

	Predicted: Positive	**Predicted: Negative**
Actual: Positive	127	202
Actual: Negative	105	224

Table 4.1

Random Forest

	Predicted: Positive	**Predicted: Negative**
Actual: Positive	208	121
Actual: Negative	68	261

Table 4.2

Multilayer Perceptron

	Predicted: Positive	**Predicted: Negative**
Actual: Positive	227	102
Actual: Negative	67	262

Table 4.3

4.2 Classification Accuracy Metrics and Analysis

Table 4.2: Showing the accuracies of our models,

Logistic Regression Classifier

Measure	**Value**	**Derivations**
Accuracy	0.4358	(TP + TN) / (P + N)
F1 Score	0.3735	2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	-0.1390	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN +FP)*(TN+FN))

Table 4.4___

Random Forest Classifier

Measure	Value	Derivations
Accuracy	0.7128	(TP + TN) / (P + N)
F1 Score	0.6876	2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.4312	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Table 4.5

Multilayer Perceptron

Measure	Value	Derivations
Accuracy	0.7432	(TP + TN) / (P + N)
F1 Score	0.7287	2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.4891	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Table 4.6

Matthews Correlation Coefficient metrics (MCC) were included as part of our analysis as it was found to have advantages over accuracy and F1 score in determining actual accuracy; this is because its score is high only if the classifier does well for both positive and negative classes.

From the metrics above, we can see that the Multilayer Perceptron performed marginally better than the Random Forest classifier; and the logistic regression classifier performed the worst out of all the classifiers. Hence, it was decided that the end product should use the MLP model for classification purposes.

4.3 Comparison With Other Models

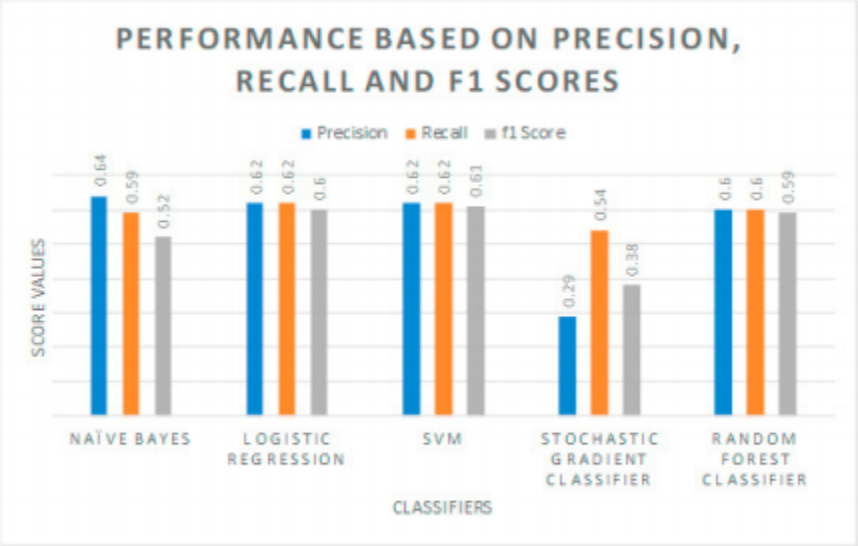

Fig 4.1: A graph showing the average precision, recall, and F1 scores of different fake news detection models (Sultana et. al)

As can be seen from comparing F1 scores, our MLP and random forest classifiers outperform other similar classifiers; our 2 best models had higher F1 scores, indicating greater model accuracy. However, our model is still unable to compete with state of the art classifiers, such as Perez-LSVM, which have F1 scores close to value of 1.0. Therefore, it can be concluded that our models were definitely above average in terms of accuracy, but were unable to achieve state-of-the art performance.

We identify a few key reasons for this being the case:

1. The considerably small training sample and test size. Due to limitations on hardware as well as the fact that our model relied on scraping links for each sample to obtain related articles, without the use of any API, it made it such that feature extraction became a very time-consuming and computationally expensive task that made us limit the training and test sample size.

2. The use of reference sources for fake news detection meant that relevance of the scraped links would play a key role in the process of feature extraction; due to the rudimentary nature of our web scraper, which was limited not by its ability to filter out relevant articles but the fact that the scope of articles and links which would be scraped were small (restricted to the sources in Table 3.1), it meant that certain news articles and headlines in our training set may not have had any related articles found at all. We tried to get around this by relying on news sources with the broadest coverage on all categories of news, and using multiple sources; however, this model is still limited.
3. Certain important features may not have been extracted too. Things like pre trained word vectors (such as word2vec), which we initially planned to use as part of the feature extraction process, could not be implemented due to time constraints.

5.1 Members' Role and Job Distributions

Oak Soe Khant → Lead Programmer (Backend Programmer)
Tan Yi Jun → Programmer (Frontend Programmer)
Tan Jun Hao Enzo → Main Administrator

5.2 Project Timeline

Implications and Recommendations

6.1 Implications of our project

Our model does a relatively good job of identifying fake news by cross referencing the suspected news with various other trusted news sources. It allows users to be able to verify the truthfulness of a news source, given a set of reference data; this will be handy especially for those who may want to fact check certain parts of articles or entire articles but not have the time or willpower to do so.

6.2 Area for Improvements and Possible Further Extensions

One key area for improvement would be the accuracy of the algorithm when it comes to detecting the fake news and how we could improve this depends on various factors, such as how many news sources we train the model with, improvements to our web scraper, improvements to the feature extraction process, and use of more powerful machine learning models such as recurrent neural networks.

If we can improve the accuracy, the next step would be to train the model to detect fake news from URLs and allow the model to scan pictures and videos to spot even more details of the fake news.

Conclusion

7.1 Reflection and Learning Points from the Project Development Process

Oak Soe Khant: From this project work, I learnt a lot about machine learning and programming which I previously didn't. I was quite unfamiliar with much of the math and technical knowledge behind this broad field of study, but having gone through it and seeing math concepts I learnt in school being applied in this field, has given me a newfound appreciation for computer science, math, programming, and learning as a whole. The experience has also taught me to be a better leader in how I manage and lead others in my team.

Yi Jun: Through this project, I have not only refined my technical knowledge but also learned more about the importance of effective teamwork, as well as the benefits of planning ahead of time. As the data collection, training and fine-tuning of machine learning models are all highly time consuming, it was critical for us to effectively carry out these tasks to meet deadlines. This project has taught me that despite having close to 4 years of project work experience, project work is very enriching and provides us with new challenges and opportunities which is evident in me still having to learn new concepts and gain new skills in order to do my part in this project.

Enzo: I think that it is a skill to be able to work together in a group to accomplish a project, working to each other's strengths and weaknesses to bring out the best in us. Even after 4 years of PW, there is still much to be learnt and refined from our journey this year. For instance, the Covid pandemic made it tough for us to meet up for discussions so we had to improvise and use online substitutes like zoom. I also think it was very essential that we not be disheartened by failure, but instead use it as motivation to do better, since it was no surprise that we encountered some issues throughout the project. All in all, I think that it was a very interesting year for PW and has definitely allowed me to learn more about my teammates and myself.

Bibliography

1. A. Watson (2020, April 22). *"Share of adults worldwide who believe fake news is prevalent in selected media sources as of February 2019"*
   Retrieved from:
https://www.statista.com/statistics/1112026/fake-news-prevalence-attitudes-worldwide/

2. C. Georgacopoulos and M. Grayce (2020, July). *"How Fake News Affected the 2016 Presidential Election"*

   Retrieved from: https://faculty.lsu.edu/fakenews/elections/sixteen.php

3. P. Paialunga (2020, May 6) *"Fake News Detection with Machine Learning, using Python"*

   Retrieved from:

https://towardsdatascience.com/fake-news-detection-with-machine-learning-using-python-3347d9899ad1

4. Z Khanam (2021) *"Fake News Detection Using Machine Learning Approaches"*
   Retrieved from:

https://iopscience.iop.org/article/10.1088/1757-899X/1099/1/012040/pdf

5. I. Ahmad , M.Yousaf, S. Yousaf , and M.O Ahmad (2020,Oct 17) *"Fake News Detection Using Machine Learning Ensemble Methods"*
   Retrieved from: https://www.hindawi.com/journals/complexity/2020/8885861/

6. J.W Zhang, D. Bowen, Philip S. Yu (2019, Aug 10) *"FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network"*
   Retrieved from:
   https://arxiv.org/pdf/1805.08751.pdf?fbclid=IwAR0tTTStVjPbDhGio2KMdvoCodCFPf2vSxRzTvH2jL3-ygA0AVD5oybR8X8