Machine Learning Report

By Alex Westgarth

Executive Summary

Today's Network is a company seeking to use a machine learning model to determine if a document is fake or real. The model was written in the R programming language (R Core Team 2022) using the RStudio IDE (RStudio Team 2022). This report used a dataset containing 11500 articles to create a model employing the Logistic Regression method and upsampling to produce a model with a 64% accuracy rating.

Table of Contents

1. Introduction

Today's Network is a company seeking to utilise a Natural Language Processing (NLP) model to identify a news article as fictitious (fake) or factual (real). This report utilises a dataset containing 11500 articles and is completed using the R programming language (R Core Team 2022) in the RStudio IDE (RStudio Team 2022). The R script completes the full processing of the dataset and tests against several models to determine the model best suited for identification, scored using Receiver Operating Characteristic (ROC) curve, Area Under Curve (AUC), confusion matrix accuracy (F1) score.

2. Methodology

The general methodology for completing the NLP modelling is as follows:

- Load the data
- Perform preliminary investigations
- Determine valuable information
- Factorise target
- Load text into corpus
- Format corpus text into machine-viable items
- Load into a DocumentTermMatrix (DTM)
- Split the data into test/train sets
- Balance the training data
- Train the models

- Evaluate and score the models

2.1. Loading Data

The dataset is a comma separated values (csv) file with 11500 entries of articles, each containing the following items:

- Text: The body of the article
- Text_Tag: The associated labels of the article
- Author: The author of the article
- Date: The date of the article
- Label: The identification of the article as real or fake

The data is loaded into a DataFrame Object. This data is examined for any missing values. All items are expressed as character values, and none have NA or NULL values.

2.2. Investigation of text

The first point of investigation is the target item, “Label”. This is done by plotting the values of “Label” in figure 1 below.

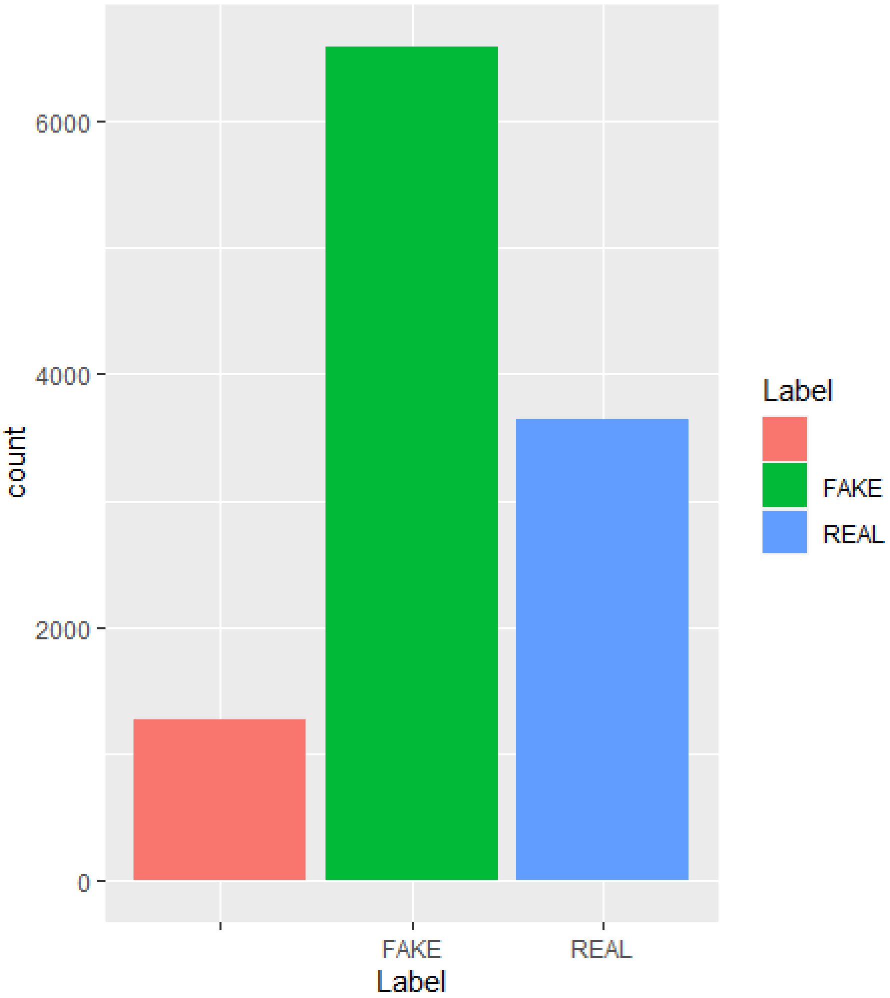

Figure 1: A plot of the counts of each value for the data item of “Label” in the loaded unedited dataset.

Two key observations are clear in Figure 1, the first is the existence of a blank value and the second is the imbalance between the fake and real counts.

The blank value indicates an article with an unknown identifier. Since the model cannot act without this information, any datapoints with this blank label are of no use. By stripping the blank values and replotting we get the view of our data in Figure 2 below.

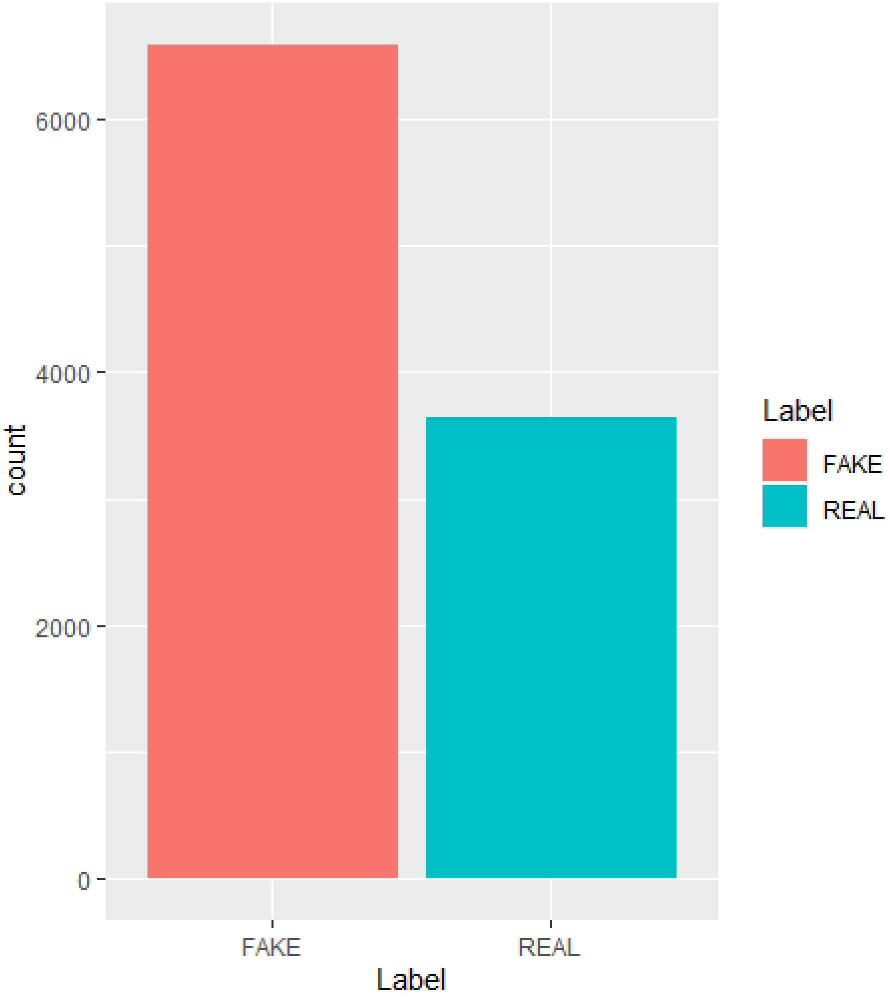

Figure 2: A plot of the counts of each value for the data item of "Label" with the blank values removed.

The imbalance between the two labels is approximately 2:1. This imbalance will cause the model to become less accurate when determining the minority label, which in this case, are the real articles. There are multiple methods for correcting the imbalance but should only be performed on the training set of the data and not the testing set. As such, the rebalancing of this data will occur after the data has been split.

The NLP model is going to be utilising the "Text" item in the data, but the other items should also be assessed to ensure no loss of valuable information is occurring.

"Text_Tag" contains a character string with each tag for the article's topics contained within, separated with commas. To ensure the total count of each tag is correctly measured, each string was split according to the commas and then summarised into the top ten by fake and real and plotted in Figures 3 and 4 below.

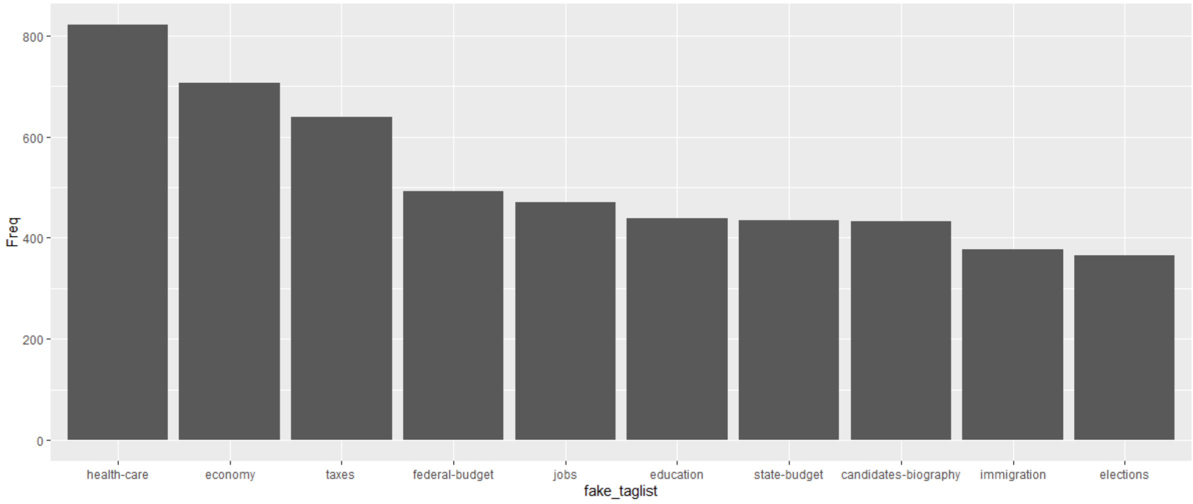

Figure 3: A graph of the top ten topics in articles marked as fake

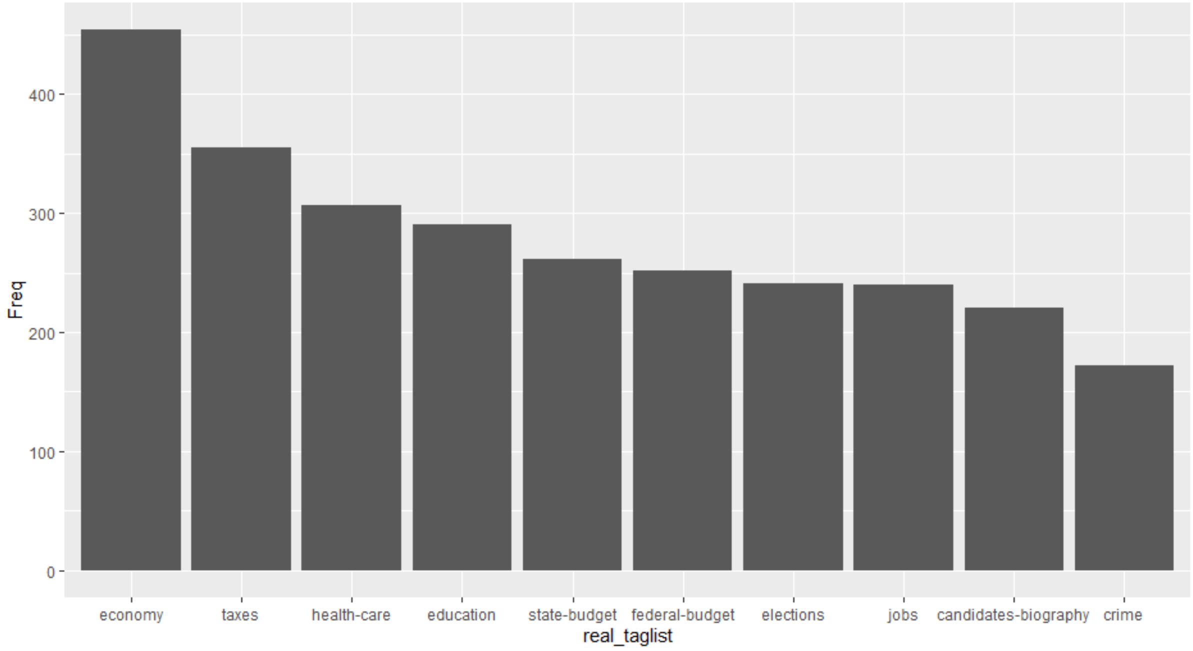

Figure 4: A graph of the top ten topics in articles marked as real

Only minor differences exist between the two lists, implying that there is no meaningful information within the “Text_Tag” item. The information presented by “Text_Tag” will be ignored in the model.

Authors shouldn’t require a detailed investigation. Even if one was to assume there are some consistencies within the dataset that imply certain authors publish more fake articles than real, this information would not assist the model in determining the inclination of any new authors. The model would have to make assumptions of the author based on their name alone, which is introducing unwanted ethical bias into the model’s functionality. For the interest of being thorough, a list of each unique author was listed for the entire dataset as well as for articles marked real and fake and produced the results seen in Table 1 below.

Dataset	Unique Author Count
Full	1000
Real	996

Fake	983

Table 1: The count of unique authors represented in the full dataset as well as in all fake and all real articles.

The dataset contains a clean 1000 authors total, and less than 5% of authors are unique to any specific label.

The last item is date. Similarly, to authors, there is no reason to assume that there is going to be an identifiable trend associated with date for the label of the dataset. Again, in the interest of being thorough, a list of dates seen in only fake or real articles is noted in Table 2 below.

Dataset	Unique Date Count
Real	0
Fake	7

Table 2: The count of unique dates present in the articles marked as real and as fake.

With not even 10 unique dates across both labels, it is safe to assume date contains no useful information.

It is now established that the only useful information is the Text and the Label.

2.3. Preparation of text

The text of the dataset is then loaded into a Corpus object. A corpus object allows quick and simple cleaning and formatting of large bodies of text. By invoking various functions, the text in the data can be set to lower case, striped of any numbers, punctuations, and whitespace. Additionally, and most importantly, corpus can strip stop words and perform stemming/lemmatization. Stop words are words present in most speech that do not provide strong context to the text value, such as “in”, “do” and “the”. Stemming involves breaking down words to their “stem” format, reducing “cold”, “colder” and “coldest” to just “cold”. Lemmatization is stemming with additional grouping based on different forms of the same word.

After performing all the formatting within the corpus, it was then loaded into a Document Term Matrix (DTM). A DTM provides a one-hot style encoded matrix that scores how many times a term appears in the text, for every term and every text. The DTM was then further filtered and cleaned to reduce the number of statistically irrelevant words. From this a wordcloud of the data was produced as seen in Figure 5 below.

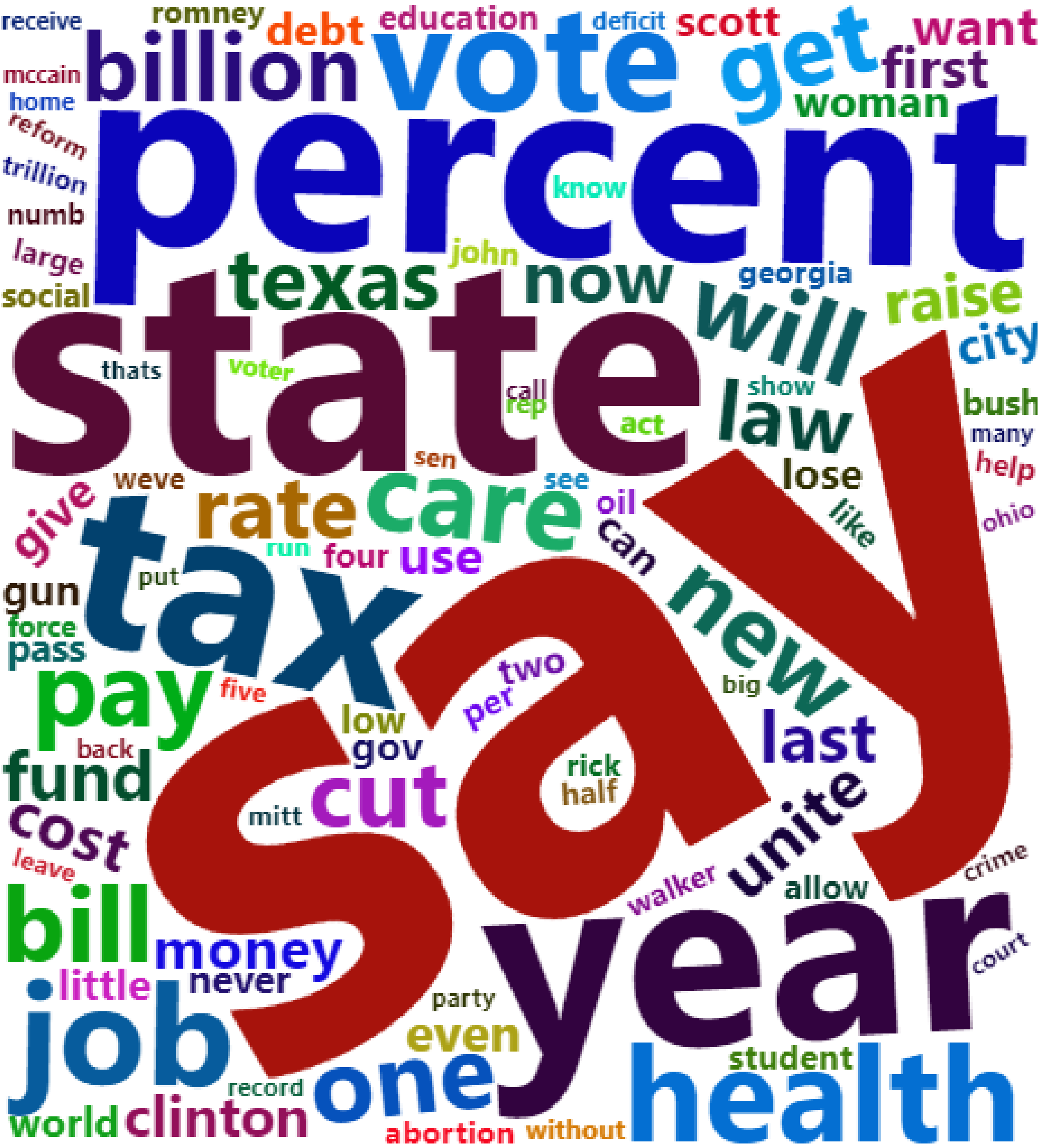

Figure 5: A wordcloud of the DTM.

This wordcloud gives a quick and simple visual representation of the most recurrent words presented within the datasets.

The data was now split into testing and training sets, with the test set representing 25% of the total original set. The training set retains the imbalance between labels as noted initially and requires balancing. To accomplish this several options were available. The results of these rebalances are seen in table 3 below.

Dataset	Count of Fake Articles	Count of Real Articles
Unbalanced	4954	2719
Upsampled	4954	4954

Downsampled	2719	2719

Table 3: A count of the fake and real articles in the unbalanced and rebalanced datasets.

Upsampling involves generating datapoints to push the minority value up to the majority value. Downsampling works in the opposite manner, stripping datapoints from the majority value until the counts are equal. Two additional common balancing methods exist, SMOTE and ROSE, however both operate primarily on number-based data, not text-based data and so these methods were ignored.

2.4. Modelling

Now that the test set and the three train sets are ready, they were loaded into three separate classification models, Naïve Bayes, Logistic Regression and Random Forest. To begin, one model was trained using the three forms of balanced to determine the best balance. Then each model was trained against the same data to determine the strongest model. For each set of tests, a ROC graph and a score was produced.

3. Results and Discussion

The results of the balancing tests can be seen in figure 6 and table 4 below.

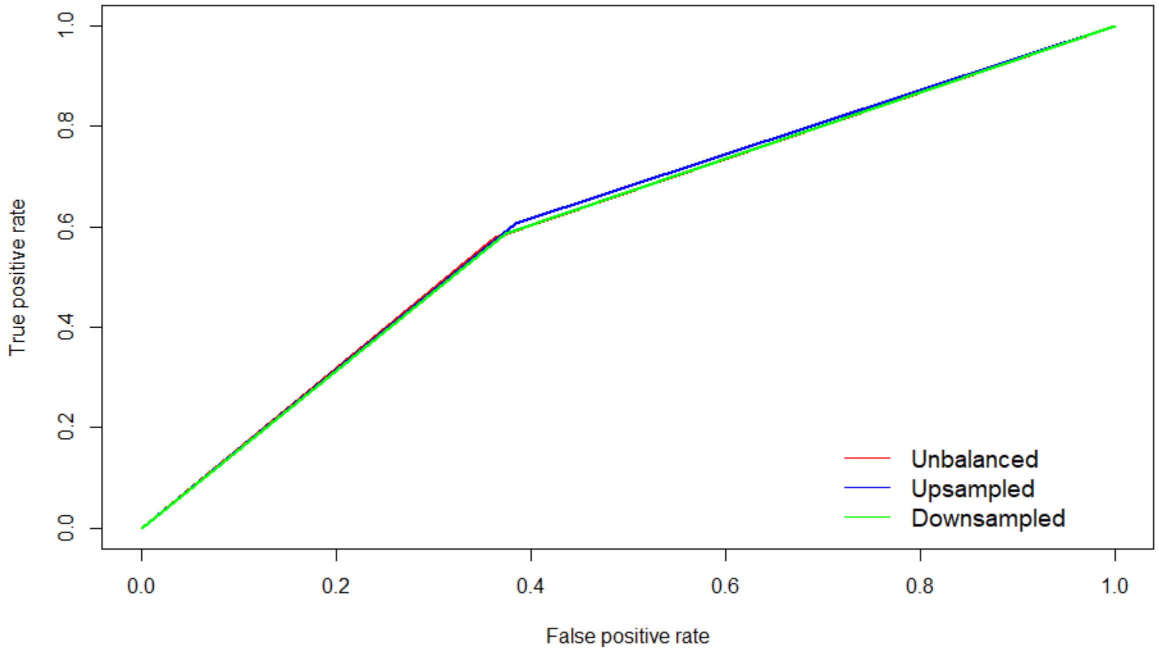

Figure 6: The ROC of the datasets using the Naïve Bayes model.

Balance	Accuracy	F1 Score
Unbalanced	0.568	0.633
Upsampled	0.578	0.639
Downsampled	0.566	0.628

Table 4: The Accuracy and F1 Score of the balancing datasets using the Naïve Bayes model.

The difference between the three sets is very minor however the general accuracy and score is greater for the upsampled dataset than the others.

The results of the model comparison tests can be seen in figure 7 and table 5 below.

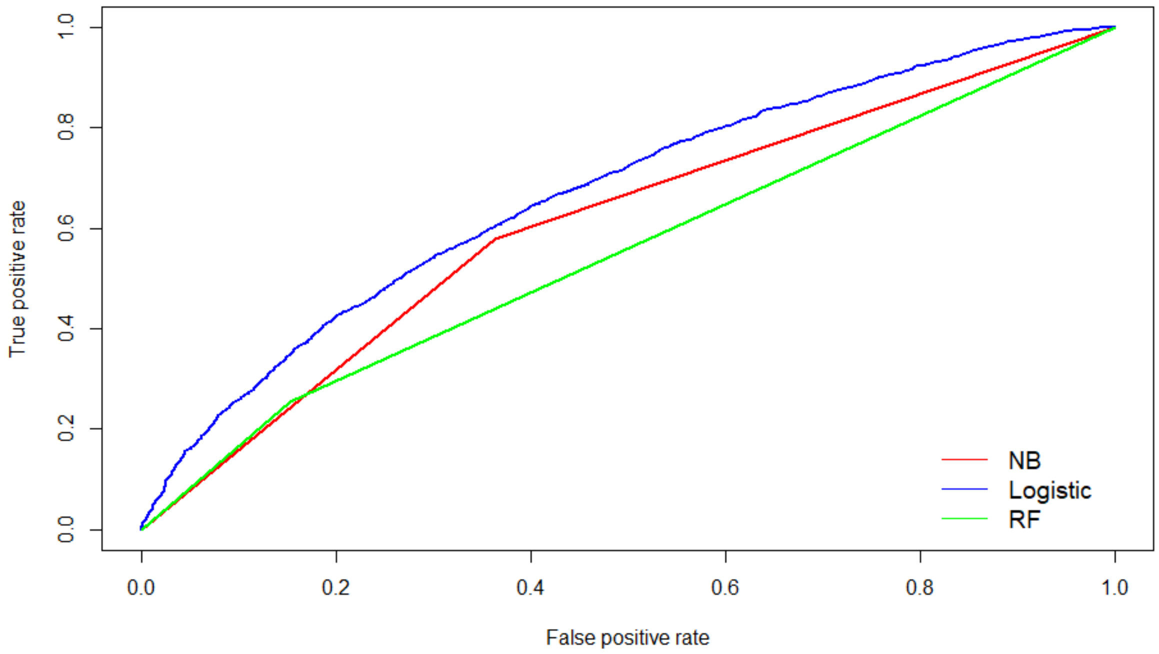

Figure 7:The ROC comparing the performance of the models.

Model	Accuracy	F1 Score
Naïve Bayes	0.568	0.633
Logistic Regression	0.641	0.762
Random Forest	0.641	0.753

Table 5: The Accuracy and F1 Score of the models.

Of the three models, the Logistic Regression Model performed the strongest, with the highest accuracy and F1 score, as well as a clearly stronger ROC.

4. Conclusion

Using the dataset provided an NLP model was created that produces an accuracy of 64% and an F1 Score of 76%. This was achieved using a logistic model trained using upsampled training data. This data must first be processed via a corpus and DTM to strip it of any non-essential details, such as numbers, punctuation, and white spaces.

References

R Core Team (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>

RStudio Team (2022). *RStudio: Integrated Development Environment for R*. RStudio, PBC, Boston, MA <http://www.rstudio.com/>