Fake News Detection Based on Linguistic Features of Headlines:

Do first impressions matter?

Student details

Name: Christos Tzouvaras

Student Number: 2009992

THESIS SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY

DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE

SCHOOL OF HUMANITIES AND DIGITAL SCIENCES

TILBURG UNIVERSITY

Thesis committee

Supervisor: dr M.Louwerse

Second reader: dr. Brouwer

Tilburg University

School of Humanities & Digital Sciences

Department of Cognitive Science & Artificial Intelligence

Tilburg, The Netherlands

02/12/2022

Word count: 7856

Abstract

Based on recent research, there is evidence that fake news and real news articles possess different linguistic features. These differences extend to their titles. This study investigated the following question: to what extent are machine learning models able to identify fake news articles, using features solely extracted from news headlines? In addition, it examined how well these models generalise to articles of different topics. To that effect, three different datasets were used, each containing different news topics. The first dataset on which the algorithms were trained and evaluated consisted of general world news. The subsequent two datasets upon which the generalisability of the models was evaluated consisted of political news and entertainment news, accordingly. The highest accuracy for the first dataset was achieved by random forests and k-nearest neighbours at 0.83, support vector machine achieved the highest accuracy for the second dataset at 0.70 and naive Bayes for the third dataset at 0.62. The results show that it is possible to discern fake news from their titles. However, the generalisation of the algorithms was poor. Overall, k-nearest neighbours was the best performing algorithm, with the highest accuracy score in the first dataset and falling closely behind the top performing algorithms (in terms of accuracy) for the other two datasets.

Data Source/Code/Ethics Statement

This study used three different datasets. None of the data creation process involved human participants. All of the datasets consist of news articles. They are publicly available, created and used in peer-review published scientific research. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data or code. All images used in this thesis, when not produced by the author, were reproduced with permission of the original authors.

Problem Statement & Research Goal

The aim of this study is to create an automated process to flag down potential fake news articles. In this manner, it would be possible to simultaneously caution news consumers to the credibility of the articles as well as hinder the propagation of fake news through social media and word of mouth. Fake news has gained a lot of traction and attracted a significant amount of attention in the latter half of the past decade. Despite the recent attention, fake news is not new. The novelty lies in the speed and width with which such news can spread and propagate throughout social media in this age of information.

The World Health Organisation (WHO) has reported that the recent pandemic was followed by a rise in misinformation related to COVID-19 (Pan American Health Organization, 2020). This rise was termed as an infodemic (Pan American Health Organization, 2020). An infodemic pertains to a sudden increase of information on a specific topic (Pan American Health Organization, 2020). WHO suggests that similarly to the way a virus spreads over a large region infecting and affecting a significant proportion of the population, misinformation may also spread (Pan American Health Organization, 2020). During the early stages of the pandemic there was an onslaught of misinformation about the origin, means of transmission, precautions, potential cures and even the existence of the virus (Siwakoti et al., 2021). Instances of the tangible consequences of such misinformation can be seen in the shortage of medicine such as hydroxychloroquine and cytokine (Solomon et al., 2020). These drugs were wrongly believed to be effective against the virus due to some early-stage uncontrolled trials. Unfortunately, the results of the studies were not properly communicated with the public, resulting in a sudden rise in the demand of these drugs which created a shortage. This shortage negatively impacted patients who were in need of these drugs. The rise of misinformation during the pandemic was so severe that the World Health Organisation launched an information platform to counter the problem (Zarocostas, 2020).

In 2016, the Oxford Dictionaries named ‘post truth’ as the word of the year. Its dictionary definition: “relating to or denoting circumstances in which objective facts are less influential in shaping public opinion than appeals to emotion and personal belief” ("Word of the year 2016", 2016). This should come as no surprise given that in that year the UK Brexit referendum and the US presidential elections took place. In both instances, there was fake news circulating, attempting to escalate societal divisions and set societal groups against each other (Marshall & Drieschova, 2018). It has been argued that such misinformation may have influenced the results of the US presidential election (Bovet & Makse, 2019) and the Brexit referendum (Marshall & Drieschova, 2018). Due to the amount of misinformation circulating, it has been proposed that we are now living in a post-truth era (Marshall & Drieschova, 2018).

A lot of research has been conducted on fake news in the past years. However, such research is highly technical and reserved for people relevant to the field. Laymen may understand the danger of fake news but would not be able to understand the research conducted into fake news detection. The state of the art currently in fake news detection is deep learning pre-trained language models (Antoun et al., 2020). Such models generally use latent features, which are not observable. These models are treated like black boxes, we know what goes in and what comes out, but we do not have a great understanding of what happens inside the box. These attributes make research into fake news inaccessible to the general public. In an attempt to increase opacity, this study examined the use of machine learning algorithms with non-latent features.

Furthermore, there are certain gaps in literature that this study identified and would aim to address. To the author’s best knowledge, there is limited literature examining the predictive power that headlines may possess in fake news detection tasks. In addition, three separate datasets were used. One to train the algorithms and create classification models and

the other two to assess the generalizability of said models. In this manner the study examined whether the topic of news in one dataset would affect performance in a dataset with a different topic.

The aforementioned points are what I believe to be the novel additions of this study to the scientific community and its general contribution on a societal level. Thus, the goal of this research is to create a tool for fake news detection based solely on news articles headlines. The research questions are:

- RQ1: To what extent can fake news articles be detected by the linguistic features of their titles?
- RQ2: To what extent does the domain of a news article affect the performance of a model?
- RQ3: Which is the best performing algorithm overall and why?

The findings of this study support that news articles can be classified into fake and real based on the linguistic features of their headlines. There were a plethora of features extracted, but only a handful were selected to build the models. The selected features were: number of verbs, number of proper nouns, emotiveness (counts of adverbs + adjectives / by counts of noun + verbs), compound sentiment score and neutral sentiment score. Using these features, k-nearest neighbours (KNN) achieved the highest accuracy score of 0.84 for the first dataset. The generalisation of all algorithms in the other datasets was poor with the best performing algorithm on the second dataset being support vector machine (SVM) with an accuracy of 0.70, and naive Bayes (NB) achieving the highest accuracy on the third dataset at 0.62. These results suggest that domain does have an effect on the performance of the algorithms. Lastly, there was not a single algorithm to outperform all others in the three datasets. However, KNN achieved the highest score for the first dataset and generalised relatively well to the other datasets.

Literature Review

Fake news is an often used term, but how would one operationalise the term? Rubin, Chen and Conroy (2016), claim that fake news can be divided into three categories: serious fabrications, large-scale hoaxes, and humorous fakes. Serious fabrication is deceptive reporting which uses exaggerations and sensationalised headlines (also known as "clickbait") to attract the attention of the public and sway opinions (Rubin, Chen, Conroy, 2019). Such publications can be unverified or purposefully misleading. Large-scale hoaxes are a deliberate falsification or fabrication in an effort to deceive audiences. Humorous fakes or satire are created for entertainment purposes and audiences are usually aware that they are actually consuming fake news.

There are two distinct approaches to fake news detection, propagation-based and content-based (Zhou et al., 2020). Propagation-based approaches detect fake news through examining information related to the manner in which fake news spread through social media. Examples of such information is who reproduces the fake news, to whom they are being transmitted to, how individuals that spread fake news relate to each other (Zhou et al., 2020). Content-based approaches on the other hand aim to detect fake news based on their content (Zhou et al., 2020). These approaches rely upon knowledge, style or latent features. Knowledge approaches tend to extract knowledge from news articles and then compare to already existing knowledge. Latent feature approaches use deep learning techniques to represent news articles when examining differences between true and fake news articles. While these approaches often yield the best results, latent features make their use from ono-specialists a challenge. Style based approaches utilise extracted features to distinguish fake news from truthful ones. Such an approach was used by Zhou et al (2020). The researchers adopted an interdisciplinary approach to fake news detection. They examined

theories from forensic and social psychology in an attempt to identify emergent patterns amongst fake news. The benefit of such an approach is that the features that are identified from these patterns are non-latent (hand-crafted), which can serve to increase the interpretability of fake news detection algorithm models. In the course of the study they compared fake news, deception and clickbait articles. When compared to truthful news articles, fake news and deception articles showed the following similarities: greater percentage of unique verbs, obscene vocabulary, emotional words. Fake news articles showed shorter words and longer sentences. Clickbait headlines were shown to be more prevalent in fake news articles rather than truthful ones. Fake news headlines were more prone to negative emotions and higher sentiment scores, characteristics which are shared by clickbait articles. In addition, both fake news headlines and clickbait headlines were more likely to display a greater number of words compared to real news. The study by Zhou et al (2020) informs of similarities between fake news, deception and clickbait. These similarities can be used as features to discriminate between fake news and truthful news. More importantly, it shows that news articles headlines features can be used to discriminate between fake and true news.

Further evidence that supports these results come from Asubiaro and Rubin (2018). The researchers compared features between fabricated and legitimate news articles. The features in question are: word count, number of informal words, verifiable facts (such as proper nouns, geographic locations, dates and times), frequency of pronouns , frequency of demonstratives (‘this’, ‘that’ ‘these’), sentiment and emotiveness (counts of adverbs + adjectives / by counts of noun + verbs). A paired sample t-test showed that when compared to real news, deceptive news were more likely to have longer paragraphs, a greater number of informal words and less verifiable facts. False news headlines were more likely to display a larger count of words, nouns, demonstratives, a greater display of emotiveness and more punctuation marks.

However, a study by Horne and Adali (2017) contradicts some of the findings from Asubiaro and Rubin (2018). Horne and Adali (2017) examined the differences between fake news, satire and legitimate news. A statistical analysis suggested that fake news articles showed great differences in their titles when compared to trustworthy news titles. Satire titles exemplified similar but exaggerated differences to legitimate news as fake titles did. These differences were: larger count of proper nouns, larger count of words, and more possessive nouns. Asubiaro and Rubin (2018) make a distinction between nouns and proper nouns, whereas Horne and Adali (2017) do not make such distinction. This may account for the conflicting results of the two studies.

Regardless, grammatical tagging appears to be an important feature for deception detection in brief text. Jamil et al. (2019) used open domain data in an effort to identify deception in short texts, while disregarding content and domain of the texts. Several features were extracted and information gain (IG) theory (a measure of reduction in entropy in a changing dataset) was implemented to choose the most informative features, such as n-gram, part-of-speech tagging and production rules. The results showed that a combination of part-of-speech with production rules while using NB as a classifier, produced an accuracy of 70 %.

A study that examines the best features for fake news detection was published by Gravanis, Vakali, Diamantaras and Karadais (2019). The authors used content based features to create models for fake news detection. They tested the performance of several popular algorithms and how their performance could be enhanced using ensemble algorithms. The authors uncovered further linguistic features that can be used in fake news detection. They concluded on 54 features which can be seen in Table 1 below. The authors found that the best performing algorithm was SVM with an accuracy of 0.95. Bagging came second with 0.94,

followed by KNN at 0.92 and NB at 0.88. These results serve as the baseline against which the performance of the models created in this study will be compared.

Table 1. Best performing linguistic features for fake news detection

A/A	Description	A/A	Description
1	# syllables	30	Tentative
2	# words	31	Certainty
3	# sentences	32	Sensory and Perceptual Processes
4	# big words	33	Social Processes
5	# syllables per word	34	Space
6	# short sentences	35	Inclusive
7	# long sentences	36	Exclusive
8	Flesh Kincaid grade level	37	Motion Verbs
9	avg # of words per sentence	38	Time
10	sentence complexity	39	Past tense Verb
11	number of conjunctions	40	Present tense Verb
12	emotiveness index	41	Future tense verb
13	rate of adjectives and adverbs	42	# Noun phrases
14	# affective terms	43	avg # clauses
15	% Words captures, dictionary words	44	avg word length
16	% Words longer than six letters	45	avg noun phrase length
17	% Total Pronouns	46	pausality
18	% First Person Singular	47	modifiers
19	% Total First Person	48	# modal verbs
20	% Total Third Person	49	passive voice
21	% Negations	50	Objectification
22	% Articles	51	Generalize Terms
23	% Prepositions	52	Group Reference
24	Positive Emotions	53	Lexical Diversity
25	Negative Emotions	54	Content word diversity
26	Cognitive Processes	55	Redundancy
27	Causation	56	Typographical error ratio
28	Insight	57	Spatio - temporal information
29	Discrepancy		

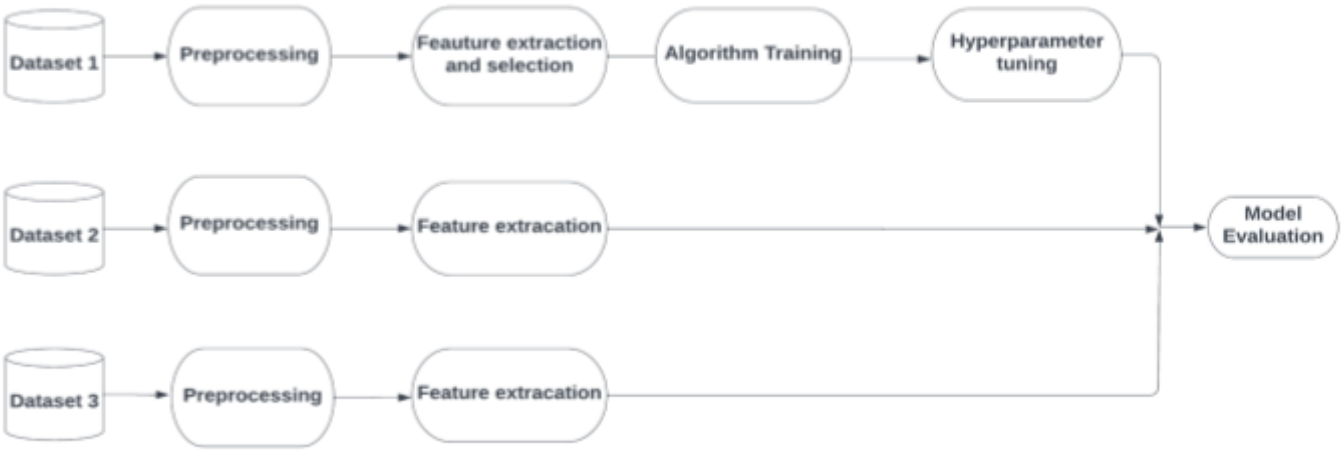
Note. This table was produced by Gravanis, Vakali, Diamantaras, and Karadais (2019) displaying best performing linguistic features. From “Behind the cues: A benchmarking study for fake news detection” by Gravanis, G., Vakali, A., Diamantaras, K., & Karadais, P. (2019), *Expert Systems with Applications, 128*, 201-213.

Methodology & Experimental Setup

The aim of this study is to examine whether fake news articles can be identified based on headlines' linguistic features. The extracted features will be used as input for four different algorithms. The algorithms' performance and generalizability will be evaluated on three different datasets.

This section will describe the methodology and the experimental setup of this study. Figure 1 acts as a visualisation of the research methodology.

Figure 1. Methodology flowchart



Datasets

Dataset 1

The dataset that will be used for training, validation and testing was used in two studies by Ahmed et al. (2017,2022). It contains 21,417 news articles labelled as real news and 23,481 articles labelled as fake news. Real news were sourced from the news website Reuters. The fake news were collected from sites that PolitiFact.com (a political fact checking website) deemed untrustworthy (Ahmed et al., 2017). Both real and fake news consist of political and world news. A length of at least 200 characters was used as criteria for selecting articles by the creators of the dataset. Each article holds the following information: article title, text, type, date of publication.

This dataset was chosen for this study based on the requirements for a fake news detection corpus as posited by Rubin, Chen, and Conroy (2016). The authors define certain criteria that a dataset should fulfil in order to be considered suitable to fake news detection tasks. The criteria are:

1. *Availability of both truthful and deceptive cases.* There should be instances of truth and deception in order for a machine learning algorithm to be able to distinguish patterns. This dataset does indeed contain instances of pre-annotated truthful and fake news articles in the dataset.
2. *Digital textual format accessibility.* The dataset should contain primarily text. Other mediums such as images can be present, videos and recordings should be transcribed. The dataset in question contains exclusively text.
3. *Verifiability of ground truth.* The dataset was previously used in studies published in peer-reviewed journals and the truthful articles were sourced from a respectable news source (Reuters.com).

4. *Homogeneity in lengths.* The articles should be similar in length for meaningful patterns to emerge. This is not a concern, as only the titles will be analysed and not the entire article. Nevertheless, the creators of the dataset used a 200 word criteria to exclude short articles.
5. *Homogeneity in writing manner.* The topics of the news articles in a dataset should cover similar topics. The dataset contains news articles from general topics with a focus on contemporary global news and political issues.
6. *Predefined time frame.* The dataset collection data-frame must be taken into consideration. Articles that are greatly separated in time may display a greater variation. The articles collected were published between the years 2016 and 2017.
7. *The manner of news delivery*. The manner in which the news is provided. This is not an aspect of the dataset that was examined.
8. *Pragmatic concerns* such as suitability of size, copy-right, privacy and accessibility. The dataset is publicly available and of a suitable size for training machine learning algorithms
9. *Language and culture*. The dataset is in English sourced from all over the world, with a focus on North American news. Due to that I do not consider language and culture to be an issue.

The creators of the dataset cleaned and processed the data, but purposely chose to maintain intact any mistakes and punctuation in the fake news set. FOr the purposes of this study, the dataset was split into training, validation and testing sets. The training set consisted of roughly sixty percent of the instances (26,923) with the validation and testing sets containing approximately twenty percent each (8,975). This split was chosen to maximise the number of training instances while maintaining the same number of instances for validation and testing. The training set will be used to train the algorithms, the validation set will be used

to tune hyperparameters and the test set will be used to evaluate the performance of the algorithms.

Dataset 2-3

This dataset was created by Shu et al. (2020) in a study about fake news in social media. The data is separated into two categories, political news (dataset 2) and entertainment news (dataset 3). The political news dataset contains 624 real news and 432 fake news collected from the site Politifact.com. The entertainment news set contains 16,817 real news articles and 5,323 fake ones collected from Gossipcop (a news entertainment fast-checking website, which no longer exists) and E!online (an entertainment news reporting website). Each article holds the following information: id, news_url (the website the article was originally published on), title, tweet_ids (tweet that cited the news article). These datasets will be used to examine the generalizability of the models trained on the first dataset.

Feature extraction and selection

As mentioned above, the dataset was already preprocessed and cleaned by the dataset owners. Extra steps were taken for the use of the dataset in this study. The capitalisation was corrected, the text was tokenized and part of speech tags were extracted using the Natural Language Toolkit (Bird, Klein, & Loper, 2009). Using the part of speech tags, further features were created such as emotiveness (counts of adverbs + adjectives / by counts of noun + verbs), modifiers and counts of different parts of speech. In addition, a sentiment analysis was performed using Valence Aware Dictionary and sEntiment Reasoner (VADER) (Hutto & Gilbert, 2014). VADER is a rule-based algorithm for sentiment analysis. It outputs scores for four categories, positive, negative, neutral and compound. Compound score is the sum of positive, negative and neutral scores resulting from the sentiment analysis, normalised between negative one and positive one (Hutto & Gilbert, 2014). VADER was chosen due to various advantages that it possesses. It is computationally inexpensive, fast and easy to

implement. In addition, VADER outperformed human raters in a sentiment classification task regarding tweets with f1 classification accuracy = 0.96 for VADER and 0.84 for human raters. It displayed the most favourable generalisation abilities when compared to other sentiment classifiers (Hutto & Gilbert, 2014).

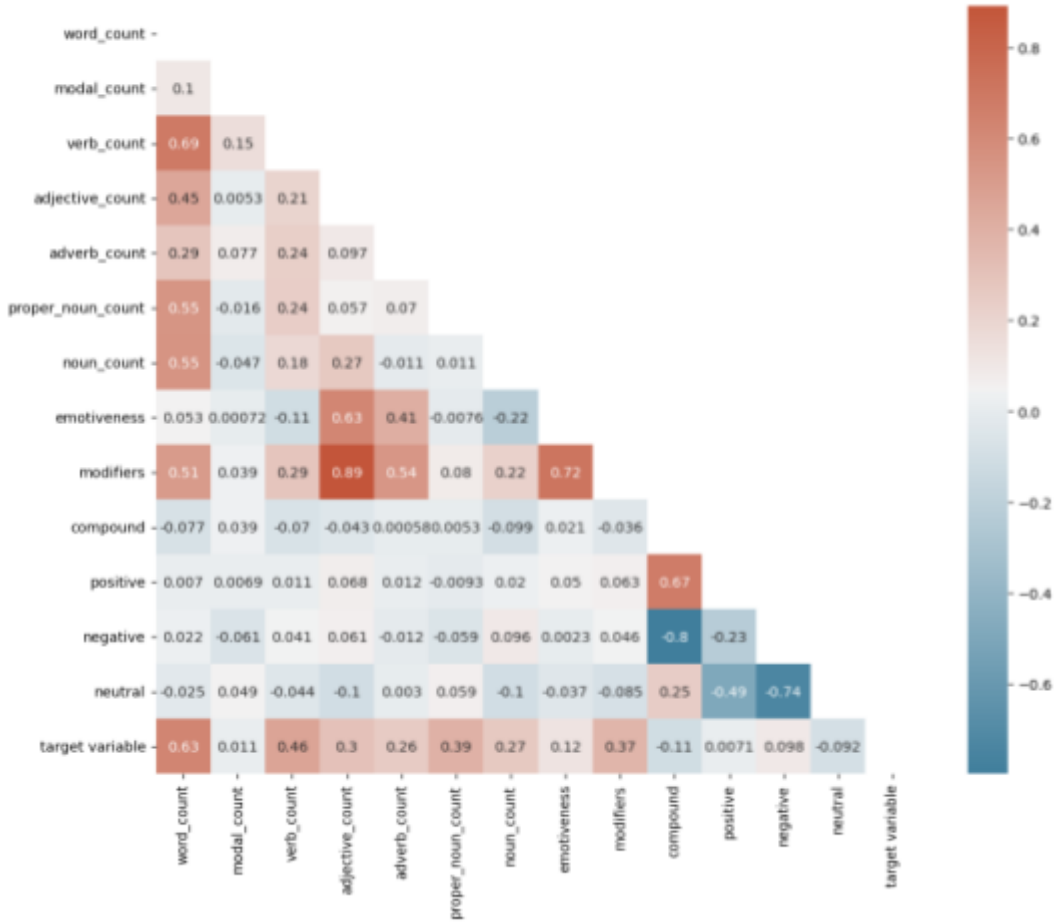
After all features were extracted, rows with NaN values were removed and the data was scaled. Twenty five rows were removed from the primary dataset, 518 rows were removed from the entertainment dataset and 246 from the political dataset. A correlation heatmap (Table 2) shows the strength of the relationships between the extracted features and the target variable. A lot of the feature variables are moderately to highly correlated with each other which indicates multicollinearity. Collinearity means that at least 2 predictor variables are closely correlated. This is a problem because collinearity undermines the statistical significance of that variable (Allen, 1997). When multiple variables are closely correlated, it is called multicollinearity. A better way to examine multicollinearity is called Variance Inflation Factor (James, Witten, Hastie, & Tibshirani, 2021). The formula for calculating the Variance Inflation Factor (VIF) of each variable is

$$\mathrm{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}, \tag{1}$$

where $R^2x_j|x_{-j}$ is the R^2 from a regression of X_j onto the rest of the variables. Any variable with a variance inflation factor value above five is considered to introduce a problematic degree of multicollinearity (James, Witten, Hastie, & Tibshirani, 2021). To address this problem, I calculated variance inflation factor values for all the variables. I removed the variable with the highest variance inflation factor value and recalculated the values. I repeated this process until all the remaining variables have variance inflation factor values below five. The selected features which will be used to train the models are: number of verbs, number of proper nouns,

emotiveness (counts of adverbs + adjectives / by counts of noun + verbs), compound sentiment score and neutral sentiment score.

Table 2. Correlation heatmap between extracted features and target variable



Algorithms

The classifiers that will be used in this study are k-nearest Neighbours (KNN), support vector machine (SVM), random forest (RF) and naive Bayes (NB).

K-nearest neighbours

KNN is a supervised classification algorithm which can be used in both classification and regression problems. The main principle of KNN is the assumption that data points which are close to each other are more similar than data points which are further away from each

other. In that fashion, data points in proximity with each other must be homogeneous, therefore belonging in the same class.

KNN assigns class membership to each new instance according to how similar that instance is to each of its neighbours. Similarity is measured according to a distance function. The “k” in KNN is the hyperparameter that reflects the number of nearest neighbours that classification will be based upon. When the value of k is too low, the algorithm tends to overfit, becoming too sensitive to noise in data and displaying a high variance and low bias.When the value of k is too high the algorithm tends to underfit. It ignores the nuanced patterns in data, displaying high bias and low variance. For these reasons, selection of k plays an important role in the performance of the algorithm.

This algorithm is a lazy learner, which means that it stores data and does not do any kind of learning until a query is performed. It is also non-parametric, which means that it makes no assumptions about the data. According to Guo et al. (2001), KNN is amongst the most effective classification methods on the Reuters corpus - a collection of newswire stories employed in text classification.In addition, the algorithm is easy to implement and easy to understand (Guo et al., 2001). This will complement my decision to use non-latent parameters in an effort to increase transparency (Zhou et al., 2020).

However, there are certain drawbacks. Since calculations take place during classification, classifying new data can be time-consuming and computationally expensive (Guo et al.,2001). The algorithm is sensitive to outliers and class imbalance. In addition, it does not perform well with large numbers of input variables.

Naive Bayes

NB is a classifier suitable for text classification, based upon the Bayes theorem, named after eighteenth-century British mathematician Thomas Bayes and is based upon conditional probability. Conditional probability is the probability that something will happen, given that

something else has already occurred. Applying the conditional probability, it is possible to calculate the probability of an event using its prior knowledge. The formula for conditional probability is as follows:

$$P(A|B) = \frac{P(B|A) * P(B)}{P(B)} \qquad (2)$$

P(A) is the prior probability of event A occurring. P(B) is the probability of event B occurring, independently of event A and prior knowledge. P(B|A) is the probability of event B occurring given that event A also occurs. P(A|B) is the probability of event A occurring when we know that event B has already happened.

It is called naive because it makes the assumption that every pair of features is conditionally independent, given the estimate of the class variable ("Naive Bayes", 2022). Such a condition is often violated when working with real world data (Zhang, 2004). However, it has been shown that for classification tasks NB performs equally with independent features and dependent ones (Domingos & Pazzani, 1997 ; Rish, 2001).

Granik and Mesyura (2017) used a bag-of-words model with a NB classifier for a Facebook posts fake news detection task. They were able to achieve a 75.9% classification accuracy. It could be worthwhile to compare the performance of the NB model that was created for this study to the aforementioned results, and examine how well headlines perform as input data.

One major disadvantage of NB is the assumption of independence which affects the conditional probabilities. This would not affect results or accuracy of the algorithm as a classifier as discussed above (Domingos & Pazzani, 1997 ; Rish, 2001), however it would reduce its interpretability, since conditional probabilities are unreliable in this case.

Support vector machine

An algorithm that performs well in cases where the feature space exceeds sample size is SVM. This algorithm is a powerful and versatile tool, useful for linear and nonlinear

regression, classification and outlier detection. The algorithm outputs a decision boundary that divides data points into classes. That does not sound dissimilar to the way that a lot of algorithms work. What sets this algorithm apart is the use of support vectors. Support vectors are the data points (from each class) closest to the separation line (Géron, 2019). The distance between the line and the support vectors is called margin. A maximal margin equals optimum hyperplane (Géron, 2019).

The algorithm performs particularly well with complex small- to medium-sized datasets. Its ability to handle high dimensional spaces means that it generalises well, a crucial aspect of a fake news detection tool. Studies comparing algorithms show that this algorithm tends to excel in classification tasks with linguistic cues (Ahmed et al., 2017; Gravanis et al., 2019). Horne and Adali (2017) used SVM to predict fake news titles based on the linguistic features of their headlines. They achieved 0.78 cross-validation accuracy over a 0.57 baseline.

Despite all the advantages there is one disadvantage which is crucial to this study. The algorithm offers no probabilistic explanation for the classification. This would decrease the interpretability of an SVM model. Solutions that circumvent this issue tend to be computationally expensive.

Random forests

Random forests (RF) is an ensemble learning method. Such methods are made up of a set of classifiers. RF as the name would suggest, is made up of multiple decision trees. Let us examine how RF produces results. Each tree randomly samples a subset of the data with replacement, also known as bootstrapping (Géron, 2019). In turn, they each generate independent results. The final result is the mode of results produced by all the trees combined (Géron, 2019). RF is easy to implement and can be used for both classification and regression problems, making it a versatile algorithm.

Using multiple decision trees can alleviate some of the issues that a decision tree algorithm is prone to, such as bias and overfitting. However, the number of trees created can produce computational load. In addition, RF can be time consuming during training, depending on the number of trees. That is why one of the main hyperparameters of RF is the number of trees created.

In a study examining the performance of different algorithms in the detection of fake news in twitter, RF achieved an accuracy of 0.96, outperforming both NB (0.92) and SVM (0.95), (Khanam, Alwasel, Sirafi, & Rashid, 2021).

Evaluation metrics

In this binary classification task, I attempt to label news articles according to their headlines. The predicted labels can either be zero (corresponding to a credible article) or one (corresponding to a deceptive article). In order to compare the predicted labels with the actual ones, I will make use of confusion matrices.

Table 1 *Confusion matrix*

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

A confusion matrix is a performance measurement for classification problems such as the one at hand. It is a table that shows the four possible combinations when comparing predicted and real labels. The combinations are: true positive (TP), true negative (TN), false positive (FP), false negative (FN). TP and TN are instances where the predicted labels match the real ones ($y_{predicted} = y_{real}$). FP and FN are instances where the predicted and real labels do not match ($y_{predicted} \neq y_{real}$). Using the confusion matrix, we can derive evaluation metrics such as accuracy, precision, recall and F-score. Accuracy (equation 1) is the ratio of all correct predictions to the total number of instances. Accuracy answers the question: How well does a model predict real values?

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

Precision is the ratio of true positive instances to the sum of positive predicted instances. Precision answers the question: Out of all the positive predicted labels, how many are correctly predicted?

$$precision = \frac{TP}{TP + FP} \quad (4)$$

Recall is the ratio of correctly classified instances to the sum of true positives and false negatives. Recall answers the question: Out of all the actual positive labels, how many did the algorithm classify correctly.

$$recall = \frac{TP}{TP + FN} \quad (5)$$

Lastly, there is F-score, a measure which combines precision and recall. F-score is the harmonic mean of a model's precision and recall. It can be adjusted to give more weight to either precision or recall.

$$F_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

Hyperparameter tuning

Using the extracted features and the aforementioned algorithms, I created models to make predictions, using the default hyperparameters for each algorithm. In an attempt to improve performance, I used GridSearchCV. In GridSearchCV one specifies parameter values of an estimator and the algorithm outputs the parameters that produce the best results. This is achieved by performing an exhaustive cross-validated grid-search over the parameter grid. A stratified 30 fold (ten splits with three repetitions) was chosen as a cross-validation splitting strategy.

For KNN, the tuned parameters were:

- *n_neighbors* with values from five to one hundred in incrementals of 5
- *weights* with two specified values (uniform and distance*)*
- *metric* with three specified values (minkowski, euclidean, and manhattan).

The best performing values were 50 *n_neighbors*, manhattan for *metric* and distance for *weights*.

For RF, the tuned parameters were:

- *max_features* with values of sqrt, log2 and None.
- *min_samples_leaf* with values ranging from one to five
- *criterion* with values of gini and entropy
- n_estimators with values ranging from ten to one hundred with intervals of ten

The best parameters values were log2 for *max_features*, two for *min_samples_leaf*, gini for *criterion* and one hundred for n_estimators.

For SVM, the tuned parameters were:

- *kernel* with values rbf, linear, sigmoid and poly
- *gamma* with values scale and auto

The best performing values were scale for *gamma* and rbf for *kernel.*

For NB, the only tuned parameter was:

- *var_smoothing* with values 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10, 1e-11, 1e-12, 1e-13, 1e-14, 1e-15.

The best performing value for this parameter was 0.001.

Figure 2. Performance before and after hyperparameter tuning

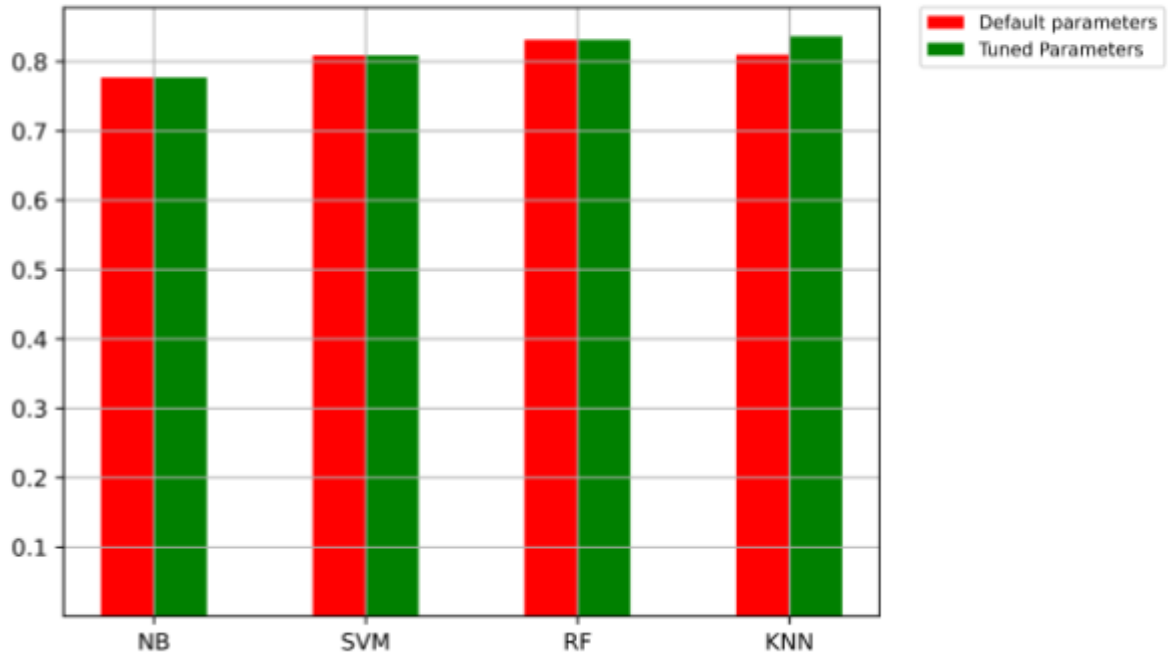


Figure 2 shows the accuracy of the algorithms with the default and tuned parameters. KNN was the only algorithm whose performance increased after tuning the hyperparameters.

Results

The aim of this study is to examine the extent to which it is possible to identify fake news articles based on the linguistic features of their headlines. Twelve separate features were created, but not all the features were used as input. Due to issues of multicollinearity a total of five features were selected and seven features were excluded. Variance Inflation Factor values were used as criteria for feature exclusion. The linguistic features of news titles that were deemed useful for fake news detection are : emotiveness (counts of adverbs + adjectives / by counts of noun + verbs), number of proper nouns, number of verbs, compound sentiment score and neutral sentiment score. For the first dataset, KNN and RF were the best performing

algorithms with an accuracy of 0.83. For the second dataset, SVM achieved the highest accuracy score of 0.70. NB was the best performing algorithm for the third dataset with an accuracy of 0.62

In the following section the results of the algorithms will be presented for each dataset. This includes the accuracy, recall and precision scores of each model for positive labels as well as the confusion matrices with the true positives, true negatives, false positives and false negatives. At this point I should note that a positive label (1) stands for fake news articles and a negative label (0) stands for real news articles.

Dataset 1

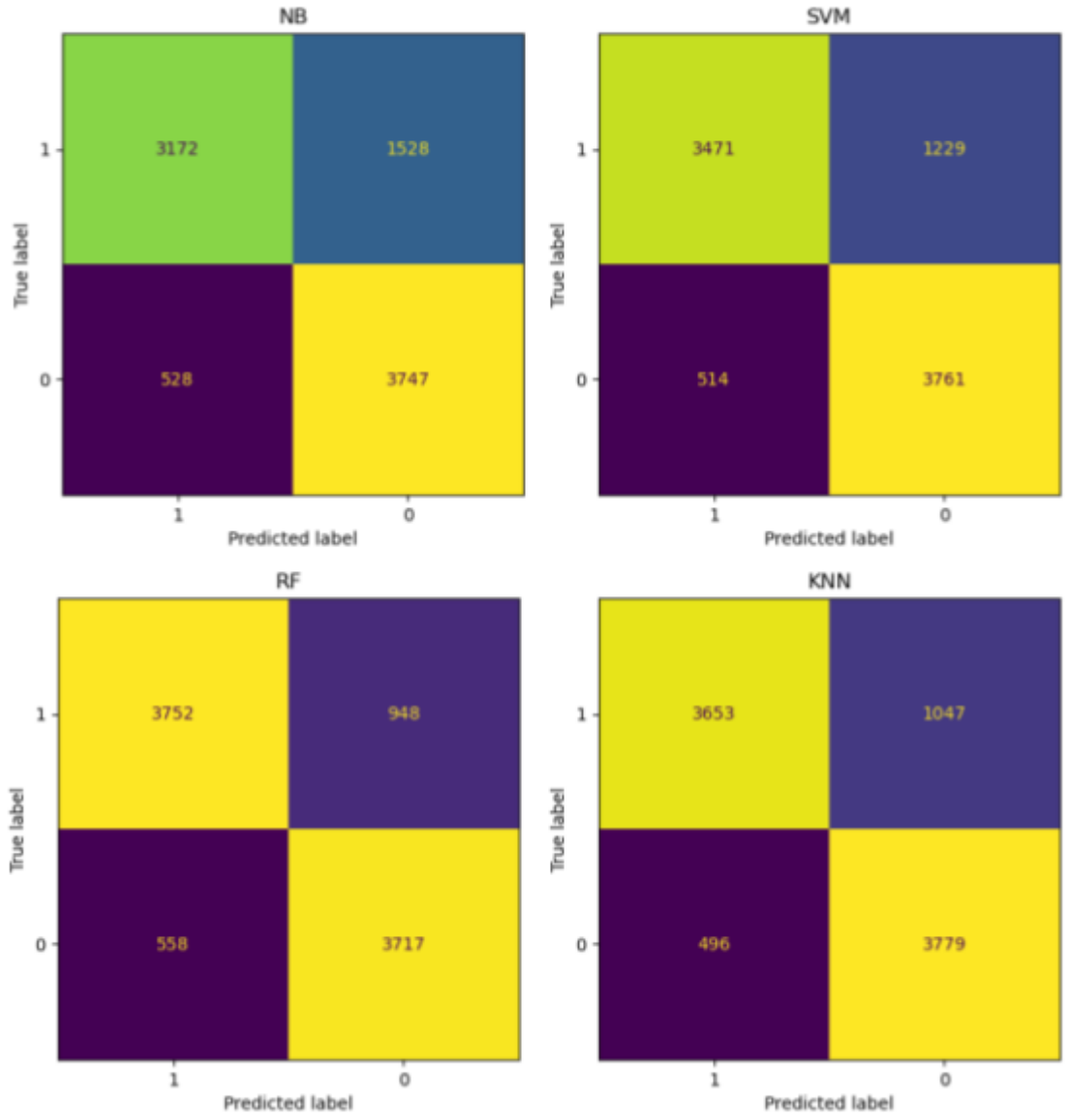
All four algorithms performed equally well on the first dataset. I will use as a baseline the results from a benchmark study by Gravanis et al., (2019). It should be noted that in that study the authors used content based features extracted by the main body of articles. Therefore, the comparison I will make will indicate how well does fake news prediction based on article headlines perform against predictions on article bodies'. The results for the first dataset are: RF and KNN had an accuracy of 0.83, followed by SVM with an accuracy of 0.81 and NB with an accuracy of 0.77. The baseline results were 0.92 for KNN, 0.88 for NB, 0.95 for SVM and 0.85 for bagging (which will serve as the baseline for RF) as mentioned in the literature review (Gravanis, Vakali, Diamantaras, & Karadais, 2019). Below I will look at the performance of each algorithm separately. A look at Figure 3 and 4 allows for a closer examination of the classification results of the models.

Naive Bayes

NB had an accuracy of 0.77, a precision of 0.86, a recall of 0.67, and an f1-score of 0.76. It displayed the highest number of false negatives. Essentially, NB classified the largest number of fake articles as truthful. In addition, it had the least amount of true positives with a big difference compared to the other articles, approximately three hundred instances. This

means that NB misclassified the largest number of real news instances as fake. In classification tasks such as fake news detection, the obvious goal is to identify fake news. At the same time, misclassifying real articles as fake could potentially be more problematic than failing to identify fake news.

Figure 3. Confusion matrices for dataset 1



Support vector machine

SVM had an accuracy of 0.81, a precision of 0.87, a recall of 0.74 and an f1-score of 0.80. It showed no outstanding results, as it achieved the second highest number of true

negatives, second highest of true positives and the second highest number of false negatives. In simple words, while SVM managed to score relatively high in correctly classifying fake news and real articles, it failed to identify several instances of fake news.

Random forests

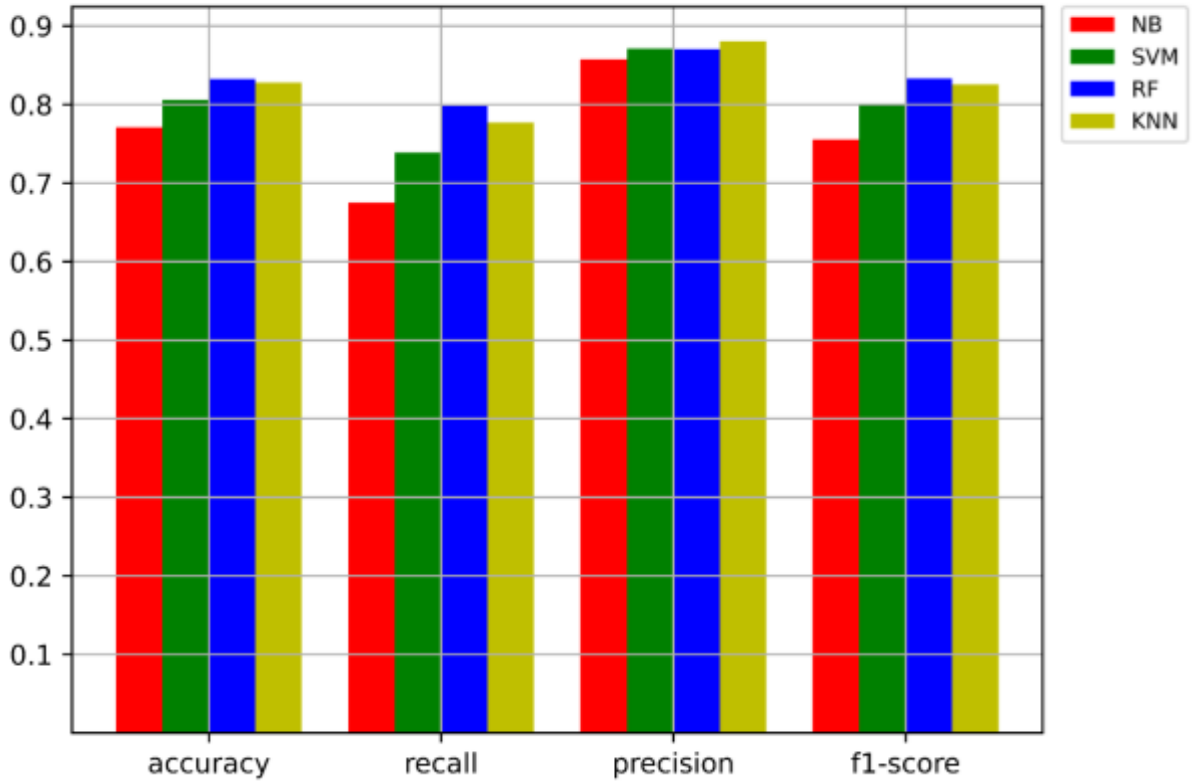
RF had an accuracy of 0.83, a precision of 0.84, a recall of 0.80, and an f1-score of 0.83. This algorithm achieved the highest number of true positives, but had the highest number of false positives. In short, RF had the highest accuracy, tied with KNN. While it managed to correctly classify the most correct instances of fake news out of all the algorithms, it also had the most instances of real news being mislabeled as fake.

K-nearest neighbours

KNN (along with RF) had the highest accuracy of 0.83, the highest precision score of 0.88, a recall of 0.78 and an f1-score of 0.83. It had the highest number of true negatives and the lowest number of true positives. These results indicate that this algorithm, when compared to the other three algorithms, excelled in classifying fake news and misclassified the least

amount of true articles as fake.

Figure 4. Evaluation metrics for the first dataset



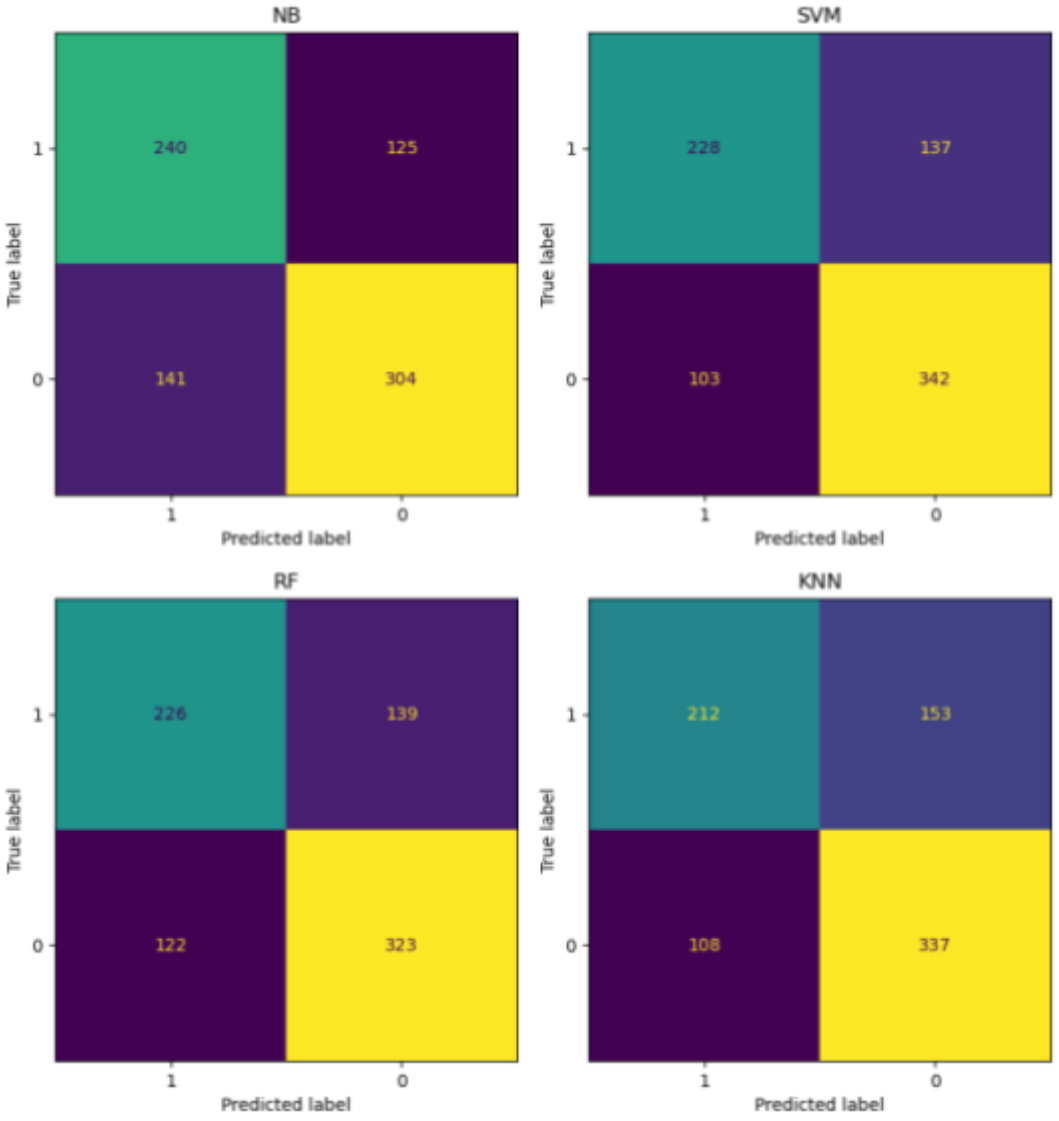
Dataset 2

Performance of the algorithms in the second dataset is considerably worse compared to the first dataset. SVM achieves the highest accuracy at 0.70 accuracy, KNN and RF come second at 0.68, while NB has an at 0.67. Let us examine the evaluation metrics per algorithm as well as the classification matrices, visualised in Figure 5 and 6 respectively.

Naive Bayes

NB had an accuracy of 0.67, a precision of 0.63, a recall of 0.66, and an f1-score of 0.64. NB achieved the highest number of true positives while having the highest number of false positives. In other words, the algorithm performed well in identifying fake news, but classified more truthful articles as fake compared to the rest of the algorithms. In addition, it correctly identified the least instances of real news.

Figure 5. Confusion matrices for dataset 2



Support vector machine

SVM had an accuracy of 0.70, a precision of 0.69, a recall of 0.62 and an f1-score of 0.65. It achieved the highest number of true negatives and the lowest number of false positives. This means that it correctly identified the most instances of real news and misclassified the least amount of real news articles as fake, when compared to the performance of the other algorithms for this dataset. Its fake news detection ability came in second as it achieved the most amount of correctly identified fake news instances after NB.

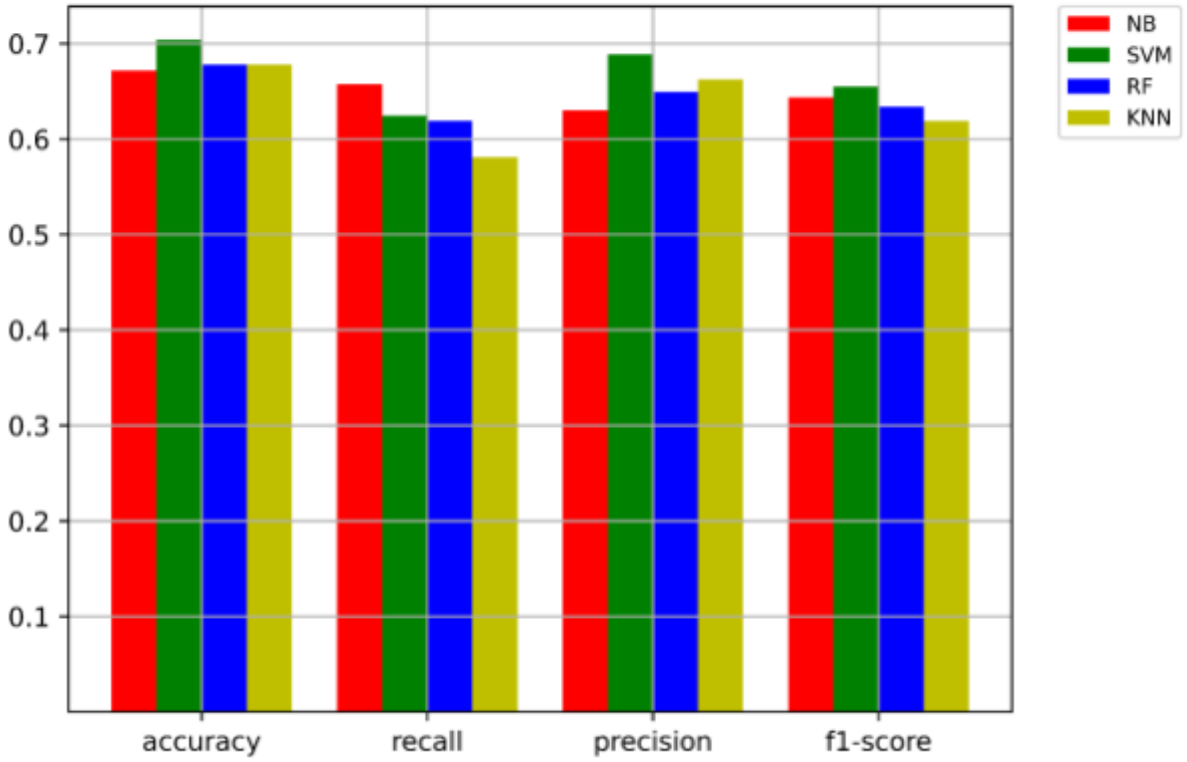
Random forests

RF had an accuracy of 0.68, a precision of 0.65, a recall of 0.62 and an f1-score of 0.64. This algorithm achieved the second lowest number of true positives and the second highest number of false positives. It misclassified a relatively large amount of fake news as real, while also misclassifying a large amount of real news as fake compared to the other algorithms. It also had relatively low numbers of correctly identifying real and fake news. Its performance is below average compared to the other algorithms.

K-Nearest Neighbours

KNN had an accuracy score of 0.68, a precision score of 0.66, a recall of 0.58 and an f1-score of 0.62. The algorithm had the second highest number of true negatives and the lowest number of true positives. This means that while the algorithm was the best amongst all other four algorithms (for this dataset) in correctly classifying real news, it also performed the worst in correctly classifying fake news. This is also evident from the algorithm's low recall score, which indicates that out of every 100 instances of fake news the algorithm is able to accurately identify only 58 of them.

Figure 6. Evaluation metrics for the second dataset



Dataset 3

Performance in this dataset is worse than the previous two. Given the fact that this dataset has imbalanced classes, I should use f1-score as a measure of performance. However, when rounding all scores up to two decimals, all the algorithms had the same score of 0.34. For that reason I chose accuracy scores to assess the performance of the algorithms. NB achieves the highest accuracy score at 0.62, followed by KNN at 0.61, SVM at 0.60 and RF at 0.57. Below I will go into more detail regarding the performance of each algorithm, visualised in Figure 7 and 8.

Naive Bayes

NB achieved an accuracy score of 0.62, a precision score of 0.29, a recall score of 0.42 and a f1-score of 0.34. It had the highest number of true negatives as well as the highest number of false negatives. This means that while NB outperformed all the other algorithms in identifying truthful news for this dataset, it also misclassified the highest number of fake news

as real. In addition, it has the lowest number of true positives, correctly classifying the least amount of fake news.

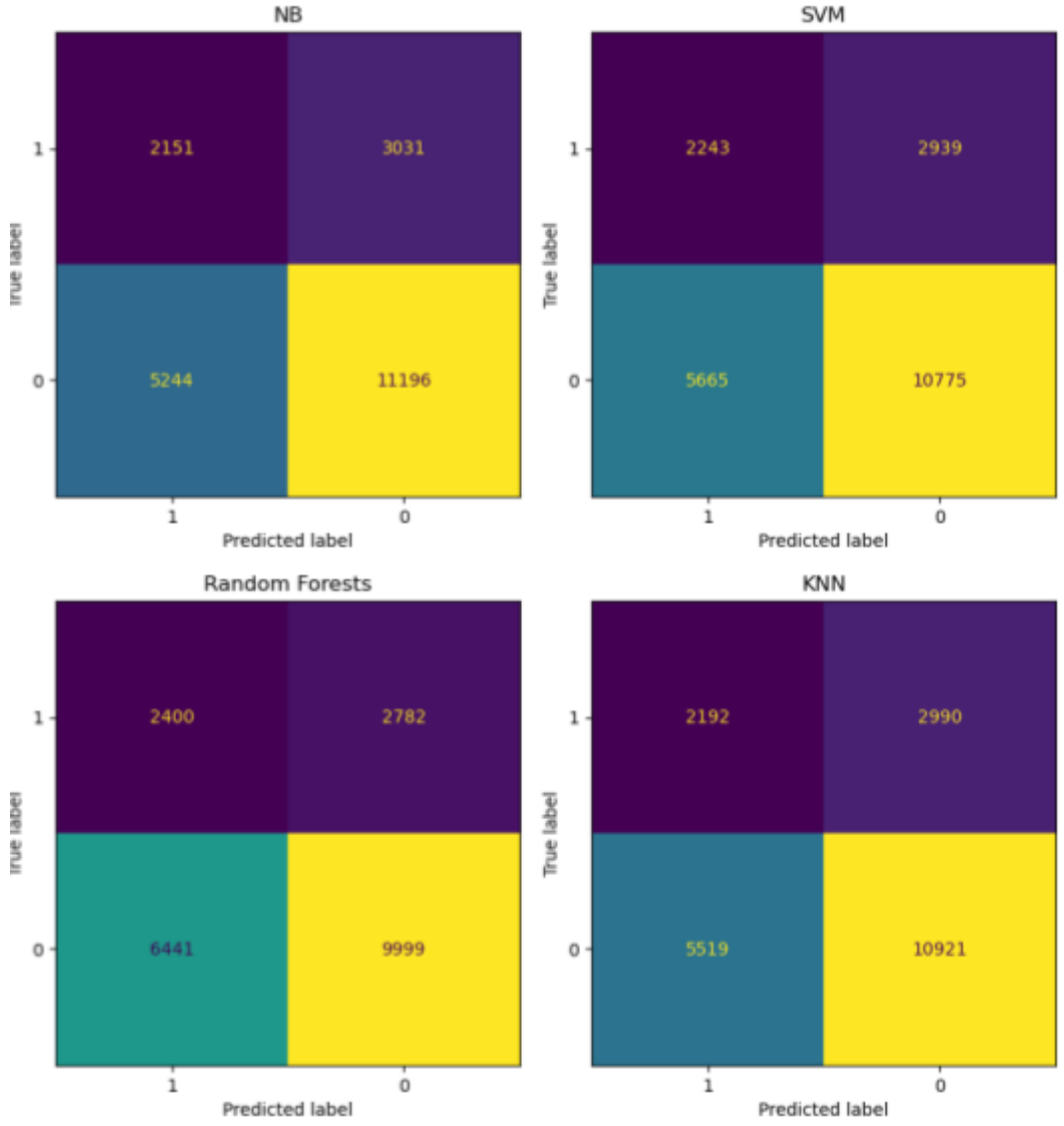
Support vector machine

SVM achieved an accuracy score of 0.60, a precision score of 0.28, a recall score of 0.43, and an f1-score of 0.34 . Its confusion matrix shows that this algorithm's classifications fall in the middle compared to the other algorithms. This algorithm was able to correctly classify the second highest number of fake news for this dataset, with RF performing slightly better in that regard. In addition, SVM has the second highest number of false positives, which means that a large number of real news articles were mistakenly classified as fake.

Random forests

RF reached an accuracy of 0.57, a precision score of 0.27, a recall score of 0.46, and an f1-score of 0.34. RF has the highest number of true positives. However, this performance was offsetted by its high number of false positives, higher than all other algorithms for this dataset. This means that while RF correctly classified the most instances of fake news, it also misclassified the highest number of fake news as real. In addition, it had the lower number of true positives, meaning that it correctly identified the least instances of real news.

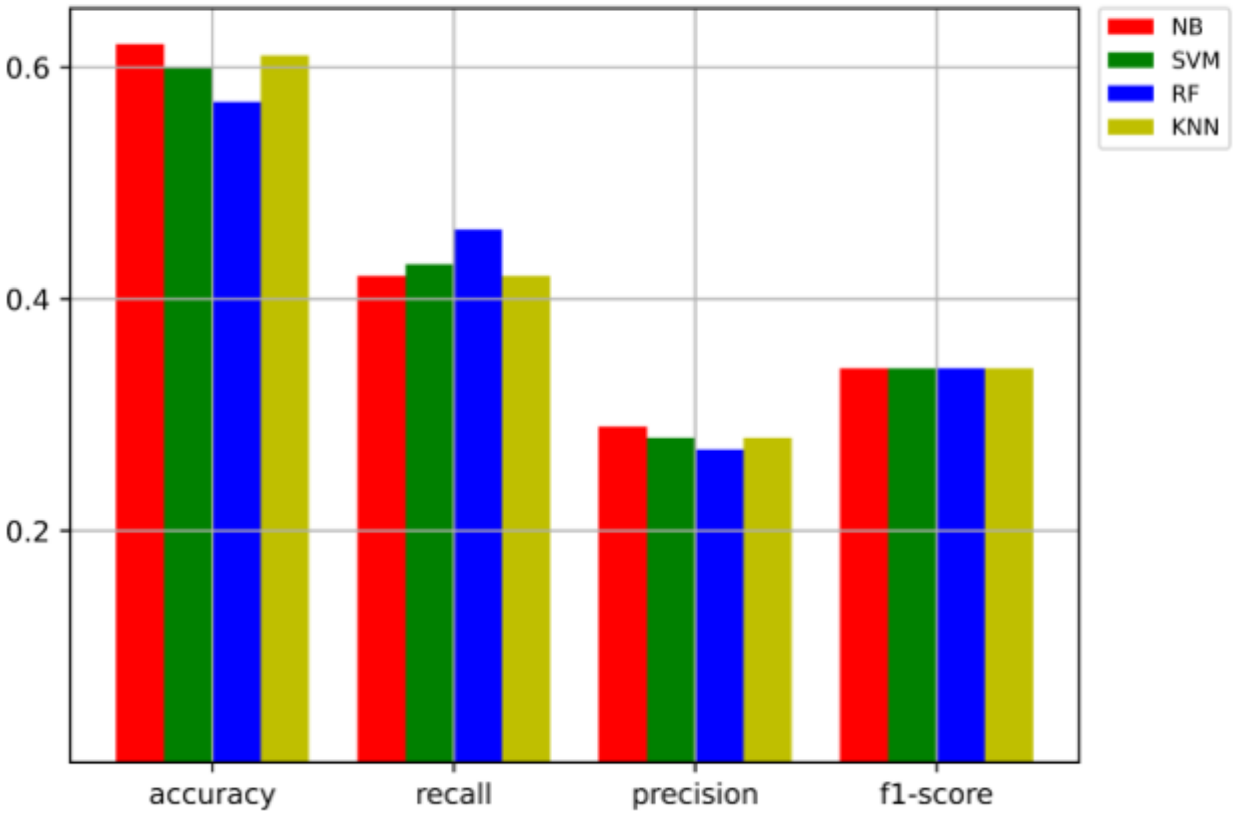
Figure 7. Confusion matrices for the third dataset



K-nearest neighbours

KNN achieved an accuracy score of 0.61, a precision score of 0.28, a recall score of 0.42, and an f1-score of 0.34. It achieved the second highest number of true negatives, second highest in false negatives and second lowest number in true negatives. KNN was able to correctly classify the second highest number of real news, but it did not perform as well in identifying fake news, misclassifying the most instances of fake news as real, after NB.

Figure 8. Evaluation metrics for the third dataset



Discussion

In this paper it was shown that titles can be used to differentiate between fake and real news articles. All four algorithms had comparable performances in the first dataset with RF and KNN achieving the highest accuracy (0.83). When taking into consideration the fact that the training time for RF was five times greater than KNN (nine minutes and fifty minutes respectively), one would deem the latter to be the better algorithm, based on speed and computational resources. Performance was in general lower for the other two datasets. For the second dataset, the highest performing algorithm was SVM with an accuracy of 0.70. This dataset was composed of political news and bore a fair resemblance to the first dataset, which was made of world news including but not limited to political news. The third dataset consisted of entertainment news centred around the life of celebrities. The highest performing

algorithm was NB with an accuracy of 0.62. It is evident that the generalisation performance of all algorithms, for this dataset, was the worst.

The results of this study exemplify that it is possible to detect fake news articles based solely on their headlines. In addition, it shows that linguistic features combined with sentiment analysis can be used as features for a task such as this. All models performed worse than the baselines in all three datasets. I believe one of the reasons for this is that there is a limited amount of information one can extract from short text such as a headline. A larger body of text will always allow for a more in depth analysis, with more features to be extracted and more patterns to uncover. The study used as a baseline examined the full body of articles and used a total of 57 features as opposed to the five features that I used.

Let us revisit the research questions again and examine the answers this study provided.

- RQ1: To what extent can fake news articles be detected by the linguistic features of their titles?

Through this study and in combination with the literature presented, it was established that there are linguistic features in news articles that can be used in fake news detection.

- RQ2: To what extent does domain influence prediction performance?

It was shown that all algorithms performed better in the first dataset on which they were trained. Performance declined for the second dataset, which bore a moderate level of similarity with the first dataset. Lastly, the algorithms showed the worst performance for the third dataset which was the least similar to the other two datasets.

- RQ3:Which is the best performing algorithm overall ?

For the task of fake news classification in the first dataset, KNN had the highest accuracy together with RF. KNN was selected as the best performing algorithm however, because of the economy that it allows in computational time and power. When I checked the

generalizability of the models, SVM had the best performance for the second dataset, and NB for the third dataset. KNN showed a stably good performance, outperforming all other algorithms in the first dataset, while generalising relatively well to the other datasets, with its accuracy and f1-score being slightly less than the top performing algorithms for each dataset. Therefore, even though there is no clear winner in terms of performance, I would argue that KNN is the most consistent algorithm across all three algorithms.

These results show that it is possible to use machine learning algorithms to identify fake news articles based solely on their headlines, potentially creating tools that could be used to quickly assess the reliability of news articles. News media consumers could use such tools on a daily basis, performing on the spot checks of headlines that grab their attention. While this approach is certainly not foolproof and could not replace manual fact checking, it could prove to be a useful tool in hindering the propagation of fake news.

A limitation of the study is the datasets that were used. Such datasets are hard to find and it is particularly difficult and time consuming to create such a dataset yourself, as one would have to fact-check and annotate each article manually. What this means, is that the truthfulness (or not) of the articles was not checked by me. I used a pre-annotated dataset, which was published in peer-reviewed journals.

Further research could examine what more information can be extracted from news headlines and how such information can be manipulated into creating features for identifying fake news.

Conclusion

The main aim of this study was to examine to what extent machine learning algorithms are able to discern fake news articles based solely on their headlines. Through the literature presented, in combination with the experiments performed, I have concluded that it is indeed possible to distinguish between fake and real news articles by the linguistic features

of their titles. Based on the results, this is not as effective a strategy as examining the full text as or using deep learning algorithms. However, the models proposed here do increase opacity in fake news detection tasks and could potentially be used as an early warning against fake news.

References

Ahmed, H., Traore, I., & Saad, S. (2017). Detecting opinion spams and fake news using text classification. *Security and Privacy*, *1*(1). https://doi.org/10.1002/spy2.9

Ahmed, H., Traore, I., & Saad, S. (2022). Detection of online fake news using n-gram analysis and machine learning techniques. https://doi.org/10.1007/978-3-319-69155-8_9

Alonso, M., Vilares, D., Gómez-Rodríguez, C., & Vilares, J. (2021). Sentiment analysis for fake news detection. *Electronics*, *10*, 1348. https://doi.org/10.3390/electronics10111348

Antoun, W., Baly, F., Achour, R., Hussein, A., & Hajj, H. (2020). State of the art models for fake news detection tasks. *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. doi:10.1109/iciot48696.2020.9089487

Asubiaro, T., & Rubin, V. (2018). Comparing features of fabricated and legitimate political news in digital environments (2016‐2017). *Proceedings of the Association for information science and technology*, *55*(1), 747-750. https://doi.org/10.1002/pra2.2018.14505501100.

Bovet, A., & Makse, H. (2019). Influence of fake news in twitter during the 2016 US presidential election. *Nature Communications*, *10*(1). https://doi.org/10.1038/s41467-018-07761-2.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, (29), 103–130. doi: https://doi.org/10.1023/A:1007413511361

Frank, M., & Feeley, T. (2003). To catch a liar: challenges for research in lie detection Training. *Journal of applied communication research*, *31*(1), 58-75. https://doi.org/10.1080/00909880305377

Géron, A. (2019). Random forests. In *Hands-on machine learning with scikit-learn, keras and tensorflow: Concepts, tools, and techniques to build Intelligent Systems* (pp. 197-198). Sebastopol, CA: O'Reilly.

Géron, A. (2019). Linear SVM classification. In *Hands-on machine learning with scikit-learn, keras and tensorflow: Concepts, tools, and techniques to build Intelligent Systems* (pp. 153-154). Sebastopol, CA: O'Reilly.

Granik, M., & Mesyura, V. (2017) Fake news detection using naive Bayes classifier. *IEEE First Ukraine conference on electrical and computer engineering (UKRCON)*, pp. 900-903. https//doi.org/10.1109/UKRCON.2017.8100379

Gravanis, G., Vakali, A., Diamantaras, K., & Karadais, P. (2019). Behind the cues: A benchmarking study for fake news detection. *Expert Systems with Applications, 128*, 201-213. doi:10.1016/j.eswa.2019.03.036

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. https://doi.org/10.1007/978-3-540-39964-3_62

Horne, B.D., & Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. https://doi.org/10.48550/arXiv.1703.09398

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). Collinearity. In *An introduction to statistical learning: With applications in R* (2nd ed., pp. 99-103). New York, NY: Springer.

Marshall, H., & Drieschova, A. (2018). Post-truth politics in the UK's Brexit referendum. *New Perspectives*, *26*(3), 89–105. https://doi.org/10.1177/2336825X1802600305

Morrison, J., Naik, R., & Hankey, S. (2018). Data and democracy in the digital age. *The Constitution Society*. Retrieved 27 September 2022

Naive Bayes. (2022). Retrieved 13 October 2022, from https://scikit-learn.org/stable/modules/naive_bayes.html?highlight=naive+bayes

Pan American Health Organization. (2020). *Understanding the infodemic and misinformation in the fight against COVID-19* [Fact sheet]. https://iris.paho.org/bitstream/handle/10665.2/52052/Factsheet-infodemic_eng.pdf

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal Of Modern Computing*, *5*(2). https://doi.org/10.22364/bjmc.2017.5.2.05

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence* (p./pp. 41--46), .

Rubin, V. L., Chen, Y., & Conroy, N. K. (2016). Deception detection for news: Three types of fakes. *Proceedings of the Association for Information Science and Technology, 52*(1), 1-4. doi:10.1002/pra2.2015.145052010083

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, *8*(3), 171-188. https://doi.org/10.1089/big.2020.006

Word of the year 2016. OUP Academic. (n.d.). Retrieved October 23, 2022, from https://global.oup.com/academic/content/word-of-the-year/

Zarocostas, J. (2020). *How to fight an infodemic* (Vol. 395, pp. 1614-1616, Rep.).The Lancet. https://doi.org/10.1016/S0140-6736(20)30461-X

Zhou, X., Jain, A., Phoha, V., & Zafarani, R. (2020). *Fake news early detection: An interdisciplinary study.* https://arxiv.org/abs/1904.11679v2.