Using Machine-Learning Methods and Linguistic Features of News Titles to Identify Fake News

Chak Lun Lam

MSc Computational Finance

Department of Computer Science
University College London

Year 2019

Supervisor: Dr. Denise Gorse

Abstract

We often rely on news reports to inform ourselves daily events around the world. Both positive and negative news has considerable influence on how we make decisions. Therefore, it is meaningful to develop an algorithm that can detect fake news so that we do not make wrong decision based on false information.

Fake news is a popular subject and numerous research projects have been conducted on identifying fake news. Apart from the traditional fact-checking method, neural networks have also been used. These methods are complex and require a large volume of data, so I aim to develop a simpler tool that can be used by the general Internet users.

To identify fake news, I am looking into the linguistic features of a news title. The hypothesis is that fake news articles have shorter words and excessive uses of special characters, capital letters, and abusive vocabularies in their titles. I am going to use machine-learning models, including Logistic Regression, Classification Tree, Random Forest, Support Vector Machine, to distinguish between real and fake news. At the moment, I am working with two types of news, political and gossip news. I want to see if different types of news share similar patterns.

The results suggest that it is easier to spot fake political news because those articles often have more special characters and capital letters in the titles. However, my models are not able to identify fake gossip news accurately. I suspect that the bad performance is due to imbalanced data. Therefore, I also try resampling technique on the gossip data. Overall, the results are encouraging and provide a basis for further research.

Acknowledgements

I would like to express my deep gratitude to my academic supervisor Dr. Denise Gorse for her guidance and advice. I would like to extend my thanks to the team at EY. Ms. Emma Birchenough-Dwyer, Ms. Erika Amelia and Mr. Tom Wilmots have provided useful suggestions and industrial expertise to help me shape this dissertation. Moreover, I am grateful to work with Liyah Dholiwar, Natalie Gapp, Antigone Kyriakide, Zhixuan (Shirley) Qu, and Wenjia Zhang. They have assisted me in developing my idea and finding relevant literature and codes.

Finally, I wish to thank my parents and friends for their support and encouragement throughout my study.

Table of contents

Chapter 1 Introduction

The American Dialect Society selected ‘fake news’ as word of the year for 2017. This shows the magnitude of attention on ‘fake news’. We often rely on news reports to inform ourselves about daily events around the world. Therefore, the truthfulness of these reports is crucial. In this digital era, social media has become the primary source for news. The easiness of writing and sharing posts on social media makes the spread of fake news contagious. Fake news is a serious social problem because it causes harm not only to individuals but also to societies. Both positive and negative news may have considerable influence on share prices or even the outcomes of elections. For example, an investor could invest in an underperforming company if there were false-positive news on that particular company, or a voter might not vote for a candidate if the candidate were allegedly involved in a scandal. In a worst-case scenario, fake news can incite hatred and distrust among citizens. Therefore, it is meaningful to develop an algorithm that can detect fake news. Such an algorithm can help delete a fake-news post before it is widely shared on social media so that the impacts of fabricated news are minimised.

Fake news is not a new problem. During the First and Second World War, the warring countries often produced propaganda to defame the oppositions. In the 21st century, tabloid magazines feature gossips and unfounded scandalous stories about celebrities and other well-known individuals. However, nowadays the speed of the spread of fake news has become a serious problem. As a result, we need equally speedy methods to identify a piece of fake news and inform the wider public before it causes any harm. Without any such tools, distinguishing fake news from genuine news is a time-consuming process. First of all, a reader needs to determine whether or not the source is credible. He then needs to search for similar news coverage from other credible websites. Finally, he should compare the reported facts such as numbers and dates on each source. Any conflicting information signals a possibility of fake news. However, online users usually just skim posts on social networks, and do not have the time to engage in this lengthy process. They are therefore likely to spread fake news without being aware of it. This is why a simple computational tool that provides immediate and accurate identification of fake news is needed. Another advantage of a computational tool is that it minimises the impact of the reader’s bias. If the reader

is biased, he may not make an objective judgement in the first place. He may believe in all the news that aligns with his political views. While it is impossible to stop people from creating fake news, it is nonetheless possible to stop the spread of fake news. The purpose of my tool is to quickly identify a piece of fake news so that the user can halt the spread of fake news.

Among all social media platforms, Twitter is a popular choice for researchers because of its application programming interfaces (APIs). On Twitter, users can post short text messages, known as *tweets*. These tweets and the user profiles of those who post them can be easily collected via the API, allowing researchers to build up databases for future analysis. On Twitter there is a 280-character limit for each tweet, so it is likely that a user posts hyperlinks to the fake news articles rather than the whole articles. Titles of fake news articles are purposely made to be catchy because the writers want the readers to click the links and share the articles. More website visitors increase advertising revenue for these websites. Therefore, I am going to investigate the differences in grammar and structure between fake news titles and real news titles. Elements of a catchy title include, but are not limited to, extensive uses of special characters, capital letters and abusive language.

In this paper, I will discuss the usage of machine learning methods in detecting fake news. The hypothesis is that fake-news articles have shorter words and excessive uses of special characters, capital letters, and abusive vocabularies in their titles. To text this hypothesis, I am going to use logistic regression, classification tree, random forest and SVM. I will compare model performances on two different domains, political and gossip data. In the next chapter, I will briefly summarise the background to this research and related work in fake-news detection. In Chapter 3, I will discuss my machine-learning methods for fake news detection. I will present my experimental results in Chapter 4, and then draw a conclusion in Chapter 5.

Chapter 2 Background and Literature Survey

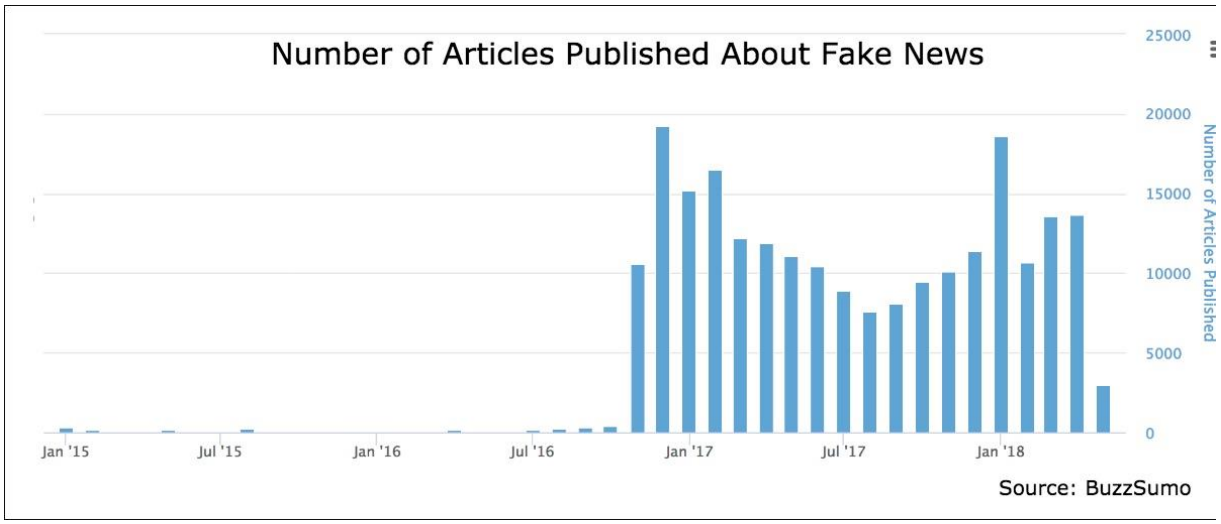
2.1 Background: Fake News

2.1.1 Definition of Fake News

Although Donald Trump often labels unfavourable reports as fake news, generally fake news refers to fabricated stories. According to the Cambridge Dictionary[1], fake news is defined as false stories that appear to be news' which 'is spread on the internet or other media'. There are many forms of fake news, such as satire, propaganda, clickbait, biased news, and sloppy journalism. Some fake news is deliberately created to mislead readers in order to influence political views or make a hoax; some is the result of careless journalism. In this paper, I define fake news as news that deviates from facts, and events that have never happened.

2.1.2 Problem of Fake News

The recent interest in fake-news identification is sparked by the US presidential election. Data from BuzzSumo[2] shows that there was a significant jump in the number of articles about fake news after Trump got elected in November 2016.



Snopes, PolitiFact, FactCheck.org, and ABC News are the initial American participants in Facebook's fact-checking program. Their fact-checking posts produced a total of 127,543 engagements on Facebook in 2017. By contrast, the top 50 fake stories in

[1] https://dictionary.cambridge.org/dictionary/english/fake-news
[2] https://buzzsumo.com/blog/content-trends-how-articles-about-fake-news-rocketed-after-trumps-election/

2017 generated a total of roughly 23.5 million engagements on Facebook according to data from BuzzSumo[3]. In other words, the fact-checking articles received only 0.5% of the engagements by the fake news. This statistic highlights the challenges in preventing the spread on fake news on social media. Social media users are more interested in reading hoaxes than fact checking. At the moment, Facebook relies on users and third-party fact-checking organisations to report inappropriate posts. It is also using machine learning to detect spam accounts.

Buzzfeed News collected the 50 most popular fake news articles on Facebook in 2017[4]. The table below shows some selected examples of fake news which can demonstrate the uses of inappropriate words and grammatical errors.

Title	Site	Facebook engagement	Publish date	Category
LAW PASSED: All Child Support in the United States Will End by Beginning of 2018	tmzbreaking.com	558,201	09-24	Politics
No more child support after 2017!!!!!!	react365.com	327,281	01-15	Politics
Angry Woman Cuts Off Man's Penis for Not Making Eye Contact During Sex - TRENDING	viralmugshot.com	981,423	02-15	Crime
Chicago Man Arrested for Slapping 25 B*tches Because He Was Tired of B*tches	viralactions.com	333,438	08-03	Crime

2.2 Technical Background

2.2.1 Matthews Correlation Coefficient

Accuracy is often used to rank the performance of a model. Essentially, it shows the percentage of correct predictions. However, it is not a good measure when the dataset is not well balanced. A better alternative to accuracy is Matthews Correlation

[3] ibid.
[4] https://www.buzzfeednews.com/article/craigsilverman/these-are-50-of-the-biggest-fake-news-hits-on-facebook-in?bfsource=relatedmanual

Coefficient (MCC). It takes false positives and negatives into account and is a better performance metric if the classes are of very different sizes. The MCC is a value between −1 and +1. A coefficient of +1 represents a perfect prediction, 0 is no better than random prediction, and −1 indicates total disagreement between prediction and observation.

Both accuracy and MCC can be calculated directly from the confusion matrix, with the following formulae:

- True positive = TP, true negative = TN, false positive = FP, false negative = FN
- $Accuracy = \frac{TP+TN}{N}$
- $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

2.2.2 Decision Trees

There are two types of decision trees, which are the classification tree and regression tree. For fake news identification, a classification tree is needed because the model predicts the class of a dependent variable based on several input variables. Moreover, a classification tree works well with non-continuous or non-linear data.

Essentially, each input is a dimension on a *p*-dimension feature space, then a set of splitting rules segments the feature space into smaller non-overlapping regions. Ideally, each region should contain only one class of data. I use the *Gini impurity* to partition the data. The Gini impurity is a measure of how often a randomly chosen element will be incorrectly labelled if it is randomly labelled according to the distribution of labels. The Gini impurity can be computed by the following formula:

$$G = \sum_{i=1}^{C} p(i)\big(1 - p(i)\big)$$

in which *C* is the total number of classes, *i* is each data point, and *p(i)* is the probability of randomly picking that data point.

When training a decision tree, the best split is chosen by maximising the *Gini Gain*, which is calculated by subtracting the weighted impurities of the branches from the

original impurity. When there is only one class of data in a node, Gini reaches the best value, zero.

2.2.3 Random forests

Since a single tree often performs poorly, a random forest, which consists of multiple trees, is needed. The first step is to create a bootstrapped data set. The bootstrapped data set is the same size as the original data set. Nevertheless, data is randomly picked from the original data set to form the bootstrapped data set, and each data can be repeatedly picked. The second step is to build a decision tree with the bootstrapped data set and a subset of variables. The tree is set up to minimise out-of-bag error. Out-of-bag samples are the data points that are in the original data set but not the bootstrapped data set. This process is called bagging, and it should be repeated hundreds of times.

2.2.4 Logistic regression

A logistic regression produces a result similar to a linear regression, that is often used in supervised machine learning. The major difference is that logistic regression returns a binary output, “0” or “1” but a linear regression returns a continuous number. The equation of a logistic regression is $F(z) = \frac{1}{1+ exp(-z)}$, where z is the output of a regression $z = \beta_0 + \beta_1 x + \beta_2 x_2 + \cdots \beta_p x_p$. This produces a cumulative distribution function in which the probability of an observation being fake news is bounded between 0 and 1. When the probability exceeds a threshold, the observation is deemed fake news; otherwise, it is predicted as real news. By default, the threshold is set at 0.5. A higher threshold not only decreases the number of positive outputs but also increases the number of false negatives.

2.2.5 Support Vector Machine

In this machine learning methodology a *p*-dimensional space contains all the data points. Then, a hyperplane divides the space into two halves which each contains only one class of data. A hyperplane is defined as:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = 0$$

After finding the values of β, a data point x will either generate a positive (<0) or a negative (<0) value. The positive values will be in one region and the negative will be in another region. In this example, a hyperplane should split the data into 'fake news' and 'real news'.

2.3 Literature Survey: Related Work

Fake news detection is not a novel topic and previous researchers have pursued different avenues. Some identify fake news from the perspective of natural language processing, some from user profiles, while some use fact checking. During my research, I discovered that many related research projects are still in the conceptual phase. Moreover, the performance of these models depends largely on data type. Thus, a direct comparison of results here with those of others may not be possible.

2.3.1 Natural Language Processing

Natural language processing (NLP) is a common research topic. (Oshikawa, Qian and Wang, 2018) have done a survey on natural language processing for fake news detection. Their paper summarises all major fake-news detection projects and methods.

Possible machine-learning models can be divided in two major categories, non-neural network and neural network. Non-neural networks suggested by Oshikawa, Qian and Wang (2018) include support vector machine, Naive Bayes classifier, random forest and logistic regression. Popular neural networks are long short-term memory (LSTM) (Rashkin et al., 2017; Ruchansky et al., 2017; Long et al., 2017), and convolutional neural networks (CNN) (Wang, 2017; Kim, 2014; Karimi et al., 2018). Classification is a common method (Mitra, Gilbert 2015), as there is a difference in word usage between deceptive language and non-deceptive ones. Oshikawa, Qian and Wang point out that some articles do not fall into one of the two classes (fake or real) because there are occasions where the article is partly real and partly fake. This ambiguous distinction between fake and real new m binary classification inaccurate. Regression is another possible method (Nakashole & Mitchell, 2014). However, it is challenging to convert discrete labels into numeric scores. Clustering (Rubin and Vashchilko, 2012) combines supervised and unsupervised machine learnings. However, significant time and labour is still required to label each group of data.

2.3.2 Fact checking

Fact checking is a common and traditional way for tackling fake news. In the past, fact checking has been a manual process which involves reading numerous official documents and other first-hand accounts. Some researchers are looking to automate this process (Thorne et al., 2018; Vlachos and Riedel, 2014; Cohen et al. 2011). FEVER (Thorne, Vlachos, Christodoulopoulos, Mittal, 2018) compare short claims to information on Wikipedia, assuming that the information on Wikipedia is correct. Firstly, human annotators extract sentences from Wikipedia and mutate them in a variety of ways. When being presented a claim, their models use term frequency–inverse document frequency (tf-idf) to retrieve the relevant documents and evidence. The best performing version achieves 31.87% accuracy in verification when requiring correct evidence to be retrieved, and 50.91% if the correctness of the evidence is ignored. This research indicates the difficulty but also the feasibility of automatic fact checking.

A similar research is done by a group at UCL (Vlachos and Riedel, 2014). They consider statements from Channel 4 and Truth-O-Meter from PolitiFact, and then cross check them with statements that are already verified by other journalists and Wikipedia. One issue with fact-checking is that not all statements can be assessed objective. Statements regarding future, causal relation or opinion cannot be verified. Moreover, the lack of machine-readable sources is another major challenge. In order to verify a claim or story, we require evidence, justifications and sources. Unfortunately, they are usually not readable by machine.

Another similar work on computational journalism is done by Cohen, Li, Yang and Yu (2011). News aggregators sometimes fail to spot false stories because it is very expensive to run an investigation or follow an original story. They propose a cloud for the crowd which allows reporters to share infrastructure and resources. They aim to build a system which automatically assign mini-tasks to be crowdsourced. For example, each journalist is assigned to check different sources in order to verify a new claim together. Eventually, these crowd contributions can build up a cloud database overtime and facilitate automatic fact-checking.

2.3.3 Clustering

Since automatic fact checking involves writing algorithms that can understand the meanings of texts, it can be an extremely difficult task. It may be easier to find common key words in fake news instead.

One of the biggest fake-news databases is CREDBANK (Mitra & Gilbert, 2015). It consists a total of 60 million tweets about 1049 real-world events. These tweets are first annotated by 30 human annotators before the researchers use hierarchical agglomerative clustering to extract three common key words for each event. This project highlights a problem in fact-checking, the credibility of some news is contentious. Although the credibility ratings of 95% of the events are agreed by at least half of the annotators, there are 49 instances in which the annotations span the entire 5-point credibility scale: certainly true/false, probably true/false, and uncertain.

PHEME is another public database of fake news (Zubiaga, Liakata, Procter, Wong, Tolmie, 2016). The journalists from Swissinfo extract highly retweeted tweets from Twitter API and label these tweets as rumour or non-rumour. The result is a set of 330 labelled source tweets across 140 stories. For each source tweet, its replies and the descendant tweets to these replies are also extracted. Eventually, 4,512 additional descendant tweets are collected. Zubiaga et al. (2016) label each source tweet as true, false, or unverified. The dataset is publicly available to allows researchers to identify characteristics of these tweets.

Vasandani looks at another social media, Reddit.[5] In Reddit, satire fake news is tagged as “the Onion” while “Not the Onion” is for real news articles that are "too crazy to be true" yet actually are real. Similarly, she clusters the common words in fake news and absurd news respectively. She achieves the highest test accuracy score of 90% accuracy with the combination of CountVectorizer and MultinomialNB. CountVectorizer converts a collection of text documents to a matrix of token counts, and MultinomialNB is a multinomial Naive Bayes classifier for classification with discrete features.

2.3.4 Neural Network

Buntain and Golbeck (2017) conducted an inclusive research on 45 features of a tweet, including structural, user, content, and temporal features. They use 2 large

[5] https://github.com/jasminevasandani/NLP_Classification_Model_FakeNews

datasets, PHEME and CREDBANK, to train their models, and then test each model on data from BuzzFeed. The resulting models from crowdsourced workers (CREDBANK) outperform the models from journalists (PHEME). The result also suggests that a simple model can outperform a complex model. The best performing model for PHEME only has 7 features. Similarly, the best model for CREDBANK only has 12 features. Some interesting examples include tweets which share media and hashtags, and tweets with emoticon, multiple exclamation, question mark, and single- and third-person pronouns.

	PHEME	**CREDBANK**
ROC-AUC	0.7407	0.7184
Accuracy	66.93%	70.28%

Results are adapted from (Buntain and Golbeck, 2017)

2.3.5 Classification

Aldwairi and Alwahedi (2018) use classification to detect fake news. They assume clickbaits are links to fake news. They focus on the syntactic characteristics of the titles, such as the number of words and capital letters in the titles. The usage of exclamation marks, question marks, hyperbole and slang words are also considered. Bounce rate is another feature used because readers often leave a cllickbait site as soon as they have visited it, resulting in high bounce rates. If the keywords in a title do not reappear in the content, this also indicates a clickbait. Classifiers such as a Bayes network, a random tree and logistic regression are used to achieve above 90% accuracy. Despite the very impressive result, we should be cautious of the sole use of accuracy as a measure, for reasons explained earlier in the technical background section 2.2.1.

Pérez-Rosas, Kleinberg, Lefevre and Mihalcea (2017) also use linguistic features such as punctuation, readability and n-grams in their research. Unlike other research in this area, they look at multiple domains, namely technology, education, business, sports, politics and entertainment. They use an SVM classifier and achieve accuracies ranging from 75 to 85%. Their work shows that the the importance of a feature varies across different domains.

Chapter 3 Methodology

3.1 Data

3.1.1 Data Collection

There are two notable problems in analysing fake news. One is the lack of pre-labelled data, and the other one is imbalanced data. It is a time-consuming, tedious, manual process to classify each news article. Also, out of all the news, only a tiny portion is fabricated news. Therefore, a starting point is using pre-labelled datasets built by other researchers. The data I used is from a data repository FakeNewsNet[6], which is constructed by a team of researchers from Arizona State University and Penn State University. (Shu, Mahudeswaran, Wang, Lee, Liu 2018) FakeNewsNet also provides the URL of the original articles. The dataset consists of two subsets, political news and gossip news.

The political news is extracted from PolitiFact[7], which is a website that provides fact-check evaluation. Reporters and editors from the media review political statements, and then categorise them as fake or true. Apart from given a judgement, PolitiFact also provides the evidence for each verdict. Therefore, it can be viewed as a reliable fact checker. PolitiFact provides six ratings: true, mostly true, half true, mostly false, false, and “pants on fire”. My set of real news only consists of statements that are classified as “true” by PolitiFact. “True” means the statement is accurate and there is no significant clarification or additional information required. Meanwhile, only statements that are “pants on fire” are included in my fake-news dataset. These statements are not accurate and make ridiculous claims.

The gossip news comprises articles from GossipCop[8] and E! News[9]. GossipCop uses a scale of 0 to 10 to classify a story from fake to real. A score below 5 corresponds to a fake story. Since the purpose of GossipCop is to showcase fake stories, almost 90% of its stories have scores less than 5. As a result, real gossip news is collected from

[6] https://github.com/KaiDMML/FakeNewsNet, accessed 13 Aug 2019
[7] https://www.politifact.com/
[8] https://www.gossipcop.com/
[9] https://www.eonline.com/uk

E! News, which is a well-known trusted website for publishing entertainment news. Shu et al. (2018) assume all the articles from E! Online are real news sources.

The table below shows that the gossip data is imbalanced with a ratio of 1:3. This are several problems associated with imbalanced data. At the same time, there are also solutions to handling imbalanced data. These problems and solutions will be discussed in the latter parts of this chapter.

	Gossip	**Political**
Real	16817	624
Fake	5323	432
Total	22140	1056

3.1.2 Imbalanced Data

There are several solutions to imbalanced data. The most obvious one is to collect more data. However, this is not usually practical. As I have discussed before, the percentage of fake news on social media is small. Merging several datasets into one large dataset is not practical, either. Data in different datasets may be drawn from various domains. As a result, the set of features may not be identical. Moreover, the importance of each feature may not be equal across all domains. Most importantly, different researchers often construct their datasets in different formats; hence, it will be extremely time consuming merely to consolidate several datasets into one dataset. Because of these reasons, it is more convenient to turn to sampling techniques.

I use 2 different methods to balance the data. The simplest way is to discard some of the ‘real news’. Given that I have 22140 observations in gossip news, I shrink the dataset to 5323 fake news and 5323 real news. Another method is biased sampling which means I add copies of observations from the smaller “fake news” class. In this case, I use the Synthetic Minority Oversampling Technique or SMOTE. The algorithm generates synthetic samples of the minority class using k-nearest neighbours. I choose k=5 for my sample. In the end, both real-news and fake-news dataset has 16817 observations. I try both approaches on the imbalanced dataset in order to see which method gives a boost in accuracy.

3.1.3 Selecting Features

Giving the time limit on this project, I am focusing on the titles of the fake news articles. I check the grammar and structure of a title in order to decide whether a piece of news is reliable or unreliable. Since fake news titles are deliberated written in a style that can attract readers, they often have abusive and emotional vocabularies. Furthermore, rather than using professional terminologies, the writers may prefer simple and short vocabularies to ensure majority of the readers can understand the titles immediately. Moreover, it is likely that the writers do not follow grammatical rules. There may be an excessive use of special characters, especially exclamation marks. Similarly, some key words or even the whole title may be in capital letters.

Here are the 8 features in my model:

i. Number of special characters in the title [10]

ii. Number of capital letters in the title

iii. Number of words in the title

iv. Number of characters in the title

v. Percentage of capital letters in terms of words in the title

vi. Percentage of capital letters in terms of characters in the title

vii. Average number of characters in each word

viii. Occurrence of abusive or inappropriate language (binary variable) [11]

3.1.4 Shuffling and Cross Validation

I shuffle the data before training a model in order to avoid any bias or patterns in the datasets. I then use k-fold cross validations for validating the performance of my models. In my research, I use 5-fold cross validation because the amount of data is not enough for having too many subsets. Each time 70% of the data is used to train the model, and then 30% of the data is for testing.

[10] Special characters are defined as ! >< :; ‘’ “ ” ' " @#~ {} \ -_ += ^& () ?
[11] The list of inappropriate words is drawn from the list of banned words on Google, https://github.com/RobertJGabriel/Google-profanity-words, accessed 20 August 2019

3.2 Machine-Learning Models

After turning these features into a n x r matrix12, I run it through various machine-learning models. The purpose is to train a classifier that will correctly classify the test data based on their observed features.

a. Logistic Regression

Logistic regression is a particularly useful model in predicting a limited set of dependent variables. In this case, the regression returns two values, 1 or 0. 1 represents that the news is fake, and 0 means the news is real. Another advantage of logistic regression is that it measures the relationship between a dependent variable and an outcome. The parameters β tell us how the odds ratio changes when the independent variable increases by one unit.

At first, I run a logistic regression with all features. This gives a benchmark of all the parameters. We cannot expect all features to be relevant to each observation; for instance, not all fake news titles have special characters. It is likely that there are some redundancies, so putting too many features in a regression tends to cause over fitting. Therefore, I use recursive feature selection to reduce the number of features. Each time the least important feature is taken away to form a smaller set of features. This process is repeated until the out-of-sample accuracy does not decrease.

Apart from reducing the number of features, regularisation is another method to improve the performance on test data. Regularisation penalises a complex model. Since I am using a Python package *scikit-learn*, Ridge regression is applied by default.

b. Classification Tree

A classification tree is well suited to this categorisation problem because it is intuitive. An observation's features are systematically checked to determine its final category. In other words, if a given observation satisfies certain criteria, it is classified as positive. Otherwise, it is classified as negative. For each split, I try to minimise the Gini Impurity so that each leaf should contain only one class of data.

[12] n = number of observations, r = number of features

Depth and pruning:

A large tree (many nodes) represents a complex model. Similar to other models, having a large tree reduces the error in the training set but may result in a higher error in the test set. Therefore, it is necessary to find a balance between bias and variance. A tree with lower depth (shorter branch) may create bias, but it can reduce the variance when different data is used. In order to find the optimal depth, I run a number of trials. In each trial I set a different depth. The depth that gives the highest out-of-sample accuracy is used for the final model.

Pruning is necessary to check whether I can lower the depth even further. Furthermore, lower depth reduces computational time. The benefit of a small tree is more obvious when the data size is extremely large. Pruning removes sections of the tree that provide little power to classify observations. Starting at the leaves, each node is replaced with its most popular class. If the prediction accuracy is not affected, then the change is kept.

c. Random Forest

Random forest is an extension to decision tree. As we have seen before, the dataset is imbalanced so a single tree may perform poorly. A single tree can yield very different predictions every time I vary the training and test sets randomly. Therefore, I have constructed a random forest to form a single consensus prediction. To find the optimal number of trees, I have tried different numbers of bagged trees. The number of bagged trees that produces the lowest out-of-bag error is used for the final model.

d. Support Vector Machine

Support Vector Machine (SVM) is another common model for classification problems. In SVM, a hyperplane separates a p-dimensional space into two halves. In this setting, the hyperplane separates the fake news from the real news. I use the soft margin classifier (support vector classifier) because it is virtually impossible to have two distinct groups. In reality, many fake news items resemble genuine news and some fake news sounds too absurd to be real. In a support vector classifier, a certain number C of training observations are allowed to be misclassified. By introducing a certain level of bias, I can decrease the variance in the test data. C bounds the number of violations to the hyperplane. A higher C simply means a higher tolerance. An optimal

C is chosen through cross validation. As in the previous cases of cross validation, a set of *C* values is tested and the one that produces the highest out-of-sample accuracy is used.

Chapter 4 Results

4.1 Data Visualisation

4.1.1 Political data

From the histograms below (figure 1 – 4), we can see that the structures of real-news and fake-news titles are distinctive. Blue bars represent real news, light red represents fake news, and dark red represents overlapped data. Figure 1 shows that fake-news titles generally have more words. From figure 2, it can be seen there are no special characters in 60% of the real-news titles. By contrast, there are 1 to 5 special characters in most of the fake news titles.

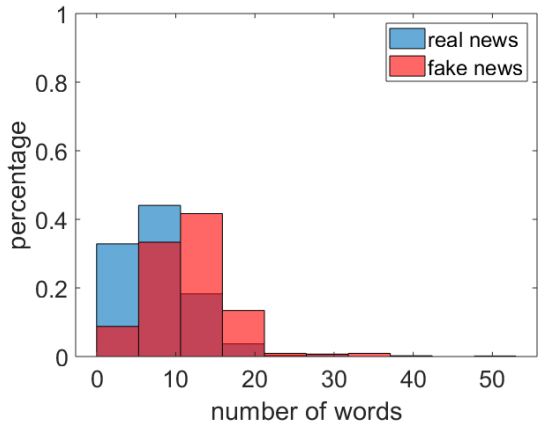


figure 1

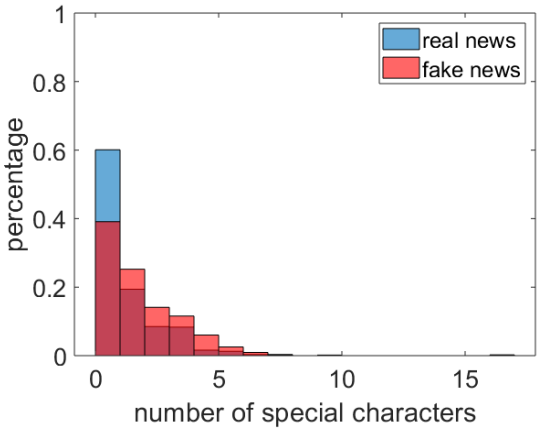


figure 2

Apart from special characters, it is not surprising that there are more capital letters in fake news titles than in real news titles. It is especially evident in figure 4 that some fake-news titles consist of capital letters only. These figures strongly suggest that building a machine-learning model with my features will help identify fake news.

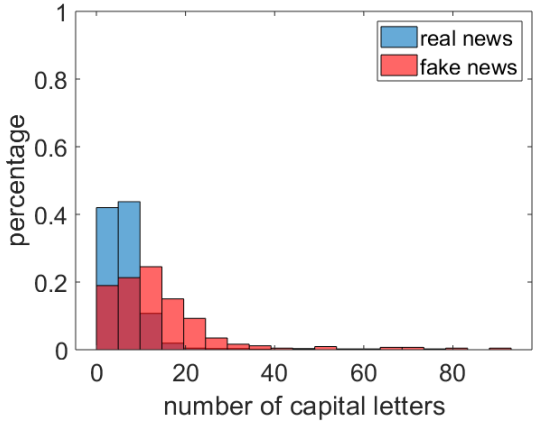


figure 3

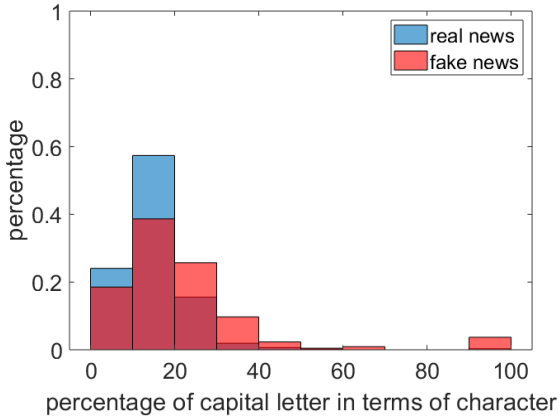


figure 4

Surprisingly, inappropriate words in these titles are rare. As we can see in the table below, the number of inappropriate words in fake news is very similar to that in real news. As a result, inappropriate wording may not be a useful method to identify fake news.

	Real news	Fake news
Number of titles with inappropriate words	2	3
Percentage of titles with inappropriate words	0.32%	0.48%

4.1.2 Gossip Data

Unlike political data, the difference between real and fake gossip news is less clear. Figures 5 to 8 show that the distributions of each feature in real-news and fake-news are similar. In particular, there is no strong evidence to suggest that fake-news titles have more words, special characters or capital letters.

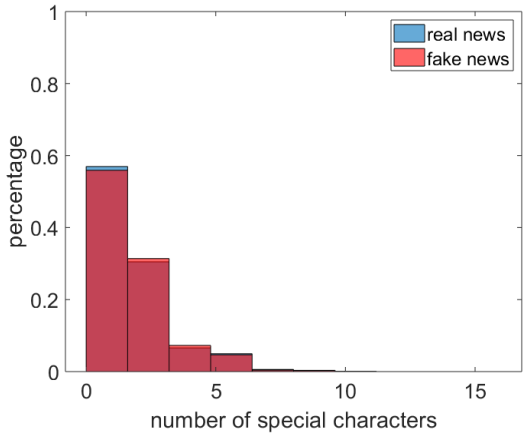


figure 5

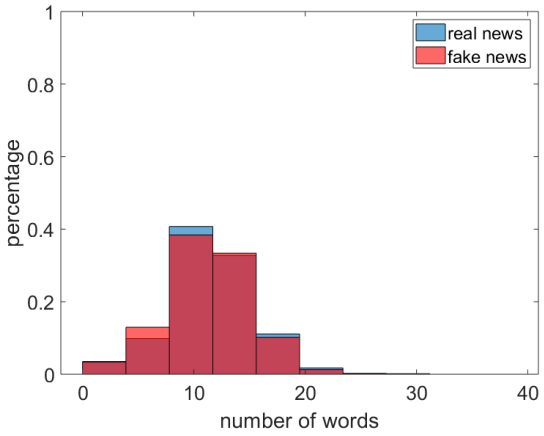


figure 6

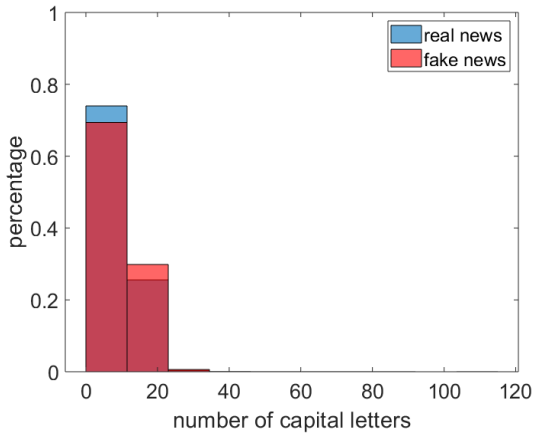


figure 7

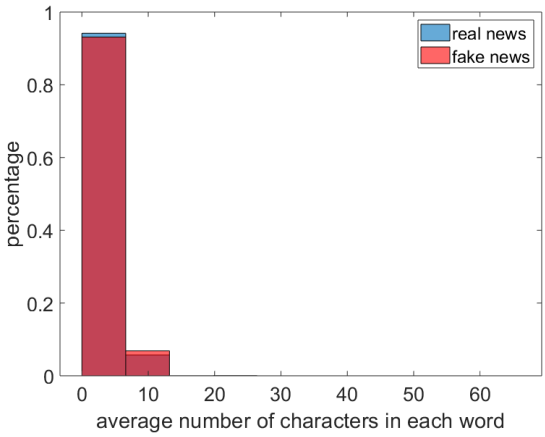


figure 8

Moreover, as in the case of political data, the percentages of titles with inappropriate words in real news and fake news are very close, as can be seen in the table below.

	Real news	Fake news
Number of titles with inappropriate words	185	59
Percentage of titles with inappropriate words	1.01%	0.83%

Overall, the results of figures 5-8 suggest that a more complex model may be needed to separate fake news from real news in the case of gossip data.

4.2 Model Performance

As discussed in Chapter 2, I will present both the accuracy and the Matthews Correlation Coefficient (MCC) for all models. Accuracy is a common performance measure which can be easily understood by readers, while the MCC is a more robust measure. Unless otherwise stated, the reported accuracy and MCC are measured with test data.

4.2.1 Political Data

a. Logistic Regression

Using a logistic regression with all 8 features, I manage to achieve 76% accuracy and 0.5071 MCC with test data. The confusion matrix below shows that a lot of the fake news is misclassified as real news. The Receiver Operating Characteristic (ROC) curve in figure 9 plots the false negative rate against the positive rate. A model should produce at least an area under curve of 0.5. Otherwise, a random classification can outperform that model. In this logistic regression, the area under the ROC curve (AUC-ROC) is 0.72. Hence, this is not a bad prediction.

		Predict	
		Real	**Fake**
Actual	**Real**	173	13
	Fake	63	68

Accuracy = 76.02%
MCC = 0.5071

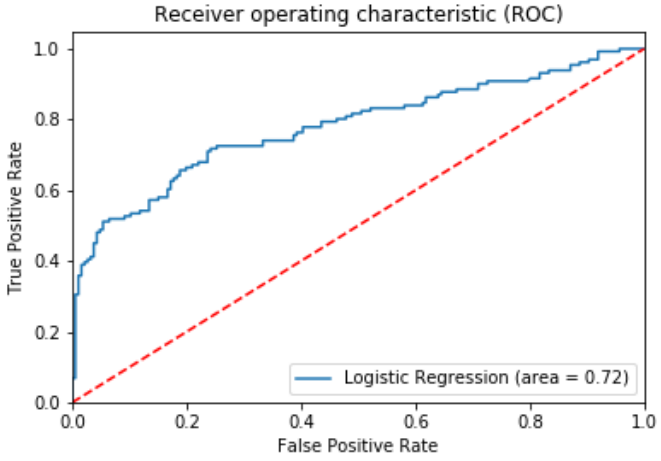


figure 9 (ROC with all features)

To improve the model performance on the test set, I try recursive feature elimination to discard unimportant features. The table below shows the model performance with different numbers of features. The top three features, the number of capital letters, the percentage of capital letters in characters, and inappropriate words, produces the highest accuracy and MCC. A larger AUC-ROC curve also shows that a smaller set of features is better.

	Number of features				
	2	3	4	5	6
Accuracy	77.60%	77.92%	77.60%	77.60	76.34%
MCC	0.5469	0.5532	0.5448	0.5448	0.5153

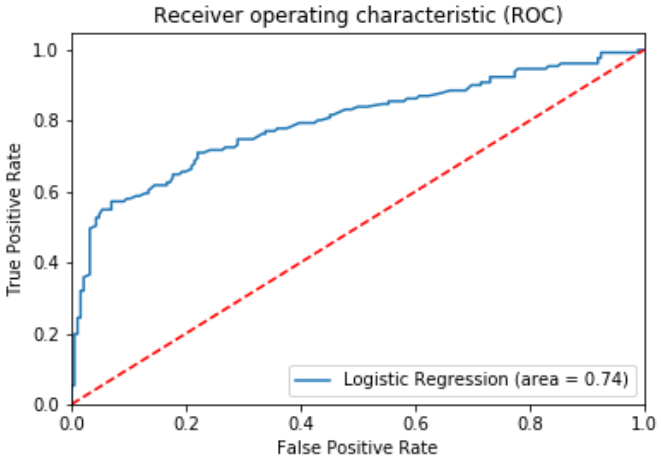
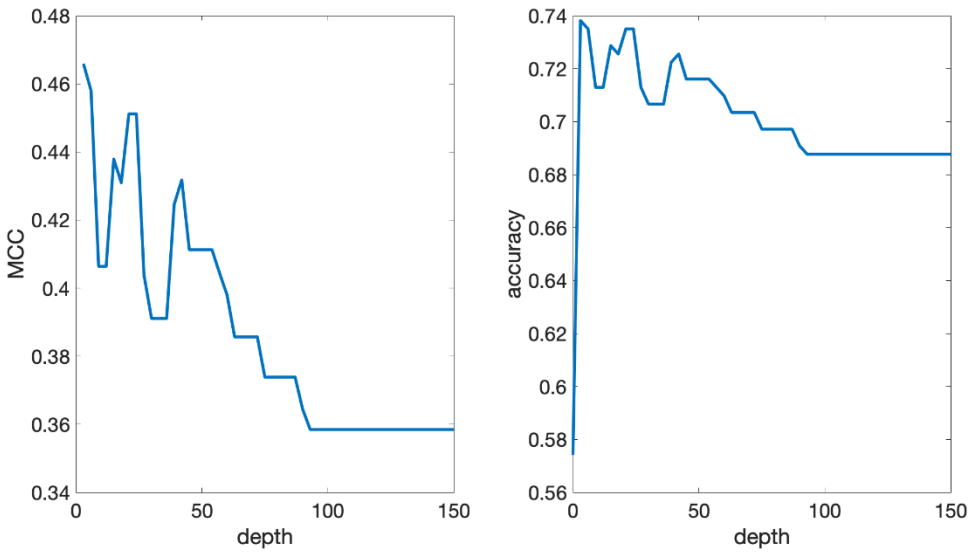


figure 10 (ROC with 3 features)

b. Classification Tree

Figures 11 and 12 show that a very small tree is enough to classify the observations in this instance. In figure 11, the MCC peaks at 3 splits and starts to fall when the tree becomes larger. Both MCC and accuracy remain constant beyond 93 splits.

figure 11 (Maximum number of splits against MCC)



(Maximum numbers of splits against accuracy) figure 12

After setting the maximum number of splits to 3 splits, the performance of a classification tree is similar to a logistic regression, as is shown by the table below:

		Predict	
		Real	**Fake**
Actual	**Real**	166	16
	Fake	67	68

Accuracy = 73.82%
MCC = 0.4659

Looking into the details of the classification tree, the set of rules is very simple.

1.
    a. if the “number of capital letter” <10.5 then class = 0 (real news)
    b. if “number of capital letter” ≥ 10.5 then node 2
2.
    a. if “percentage of characters in capital letters” <15.5 then class = 0
    b. if “percentage of characters in capital letters” ≥15.5 then class = 1 (fake news)

Figure 13 shows the importance of each feature. The number of capital letters in a title is by far the most importance feature. Combining with the splitting rules, it implies that most of the real-news titles have fewer than 10.5 capital letters.

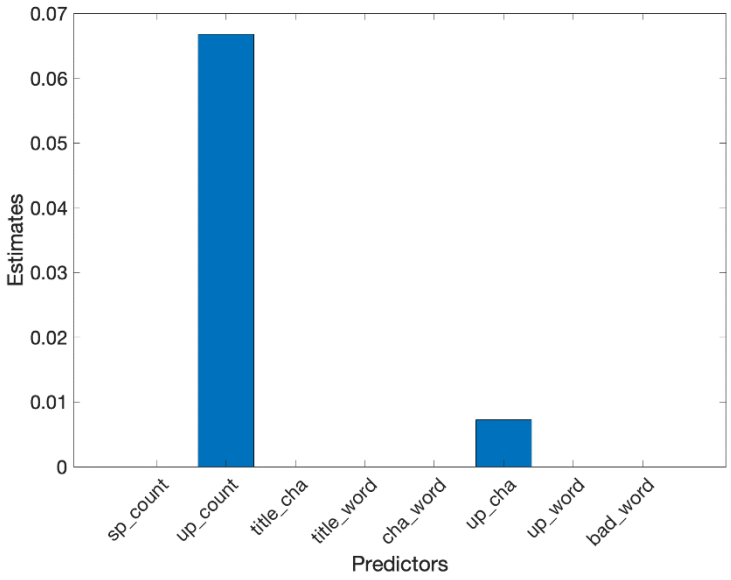


figure 13 (importance of each feature)[13]

c. Random Forest

Since a single classification tree may not generate the best result, I also try a random forest. First, I need to find the optimal number of trees in the forest. I build several forests with different numbers of trees, and then compute the MCC for each forest. Figure 14 shows that the effect of more trees on out-of-bag error starts to disappear once there are more than 20 trees in the forest. A forest with 50 trees appears to produce the lowest error, so I am going to build another forest with 50 trees and test it on the test data.

Furthermore, previous results for the classification tree show that a small tree is enough to produce high accuracy. Therefore, I also limit the depth of each individual tree to 3 splits, which is the optimal number from the previous section. The results

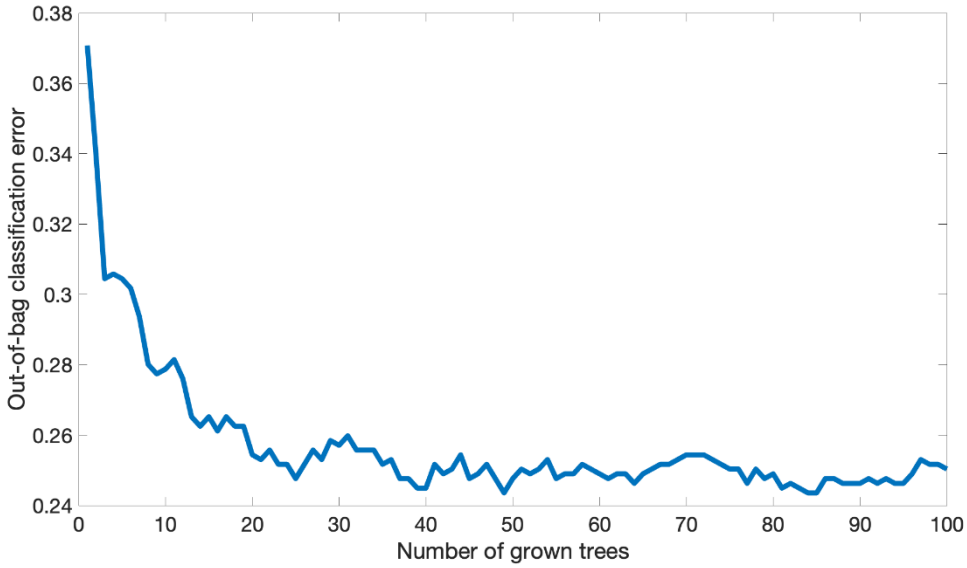
[13] *sp_count* = number of special characters; *up_count* = no. of capital letters; *title_cha* = no. of characters; *title_word* = no. of words;
cha_word = average no. of character of each word; *up_cha* = percent of capital letters in terms of character; *up_word* = percent of capital letters in terms of word;
bad_word = inappropriate words

from a random forest are similar to the results from a classification tree. Both methods result in MCCs of around 0.46 and accuracies of around 73%, as can be seen below:

	Depths	
	No constraint	**3 splits**
accuracy	70.66	73.82
MCC	0.3920	0.4623

Confusion matrix for random forest with constrained depth:

		Predict	
		Real	**Fake**
Actual	**Real**	163	19
	Fake	64	71



(number of trees in a forest against out-of-bag error) figure 14

d. Support Vector Machine

Similar to classification tree, Support Vector Machine (SVM) separates the whole feature space into subspaces which each contains only one class. Since the features are in different scales and ranges, Gaussian kernel results in higher accuracy and MCC than linear kernel. The table summarises the performance of the linear SVM and the Gaussian SVM.

	Kernel	
	Linear	**Guassian**
accuracy	70.98%	73.50%
MCC	0.4035	0.517

Confusion matrix for Gaussian SVM:

		Predict	
		Real	**Fake**
Actual	**Real**	157	19
	Fake	59	76

I then try to introduce a soft margin to the Gaussian SVM to check if the allowance of misclassified training observations will improve out-of-sample performance (see table below). I set several maximum numbers C of misclassified training observations. Surprisingly, the resulting MCCs are lower than the one from an unconstrained model. In MATLAB, the default C is set at 1.

	Maximum number of misclassifications in training set (*C*)			
	5	**15**	**25**	**50**
accuracy	71.92%	71.29%	71.61%	69.40%
MCC	0.4173	0.4048	0.4127	0.3686

At this point, logistic regression appears to be the best model despite being the simplest of the models implemented. Moreover, as in the cases of regression and tree-based models, reducing the number of features or simplifying the model can improve the performance. One may think these results suggest fake-news detection is easier than expected. The situation is however not so straightforward, as the following section will show.

It is also worthy to point out all the models produce relatively more false negatives than false positives. It means the models fail to filter out many fake-news articles. This may not be ideal since the objective is to create a tool that can spot fake news.

4.2.2 Gossip Data

a. Logistic Regression

While it was effective for political data, logistic regression does not do well in the case of gossip data. The accuracy of this regression only reflects the imbalanced distribution of data instead of measuring the real performance of the model. The model classifies almost every observation as real news. As a result, about 24% of the fake news is wrongly classified as real news. (see table on next page))Although the

regression achieves 76% accuracy, it is not a good model. A MCC of -0.01 shows that the model is not better than a random classifier. Therefore, other measures should be used to show the effect of different thresholds on accuracy.

		Predict	
		Real	**Fake**
Actual	**Real**	5026	3
	Fake	1613	0

Accuracy = 75.65%
MCC = -0.0120

The ROC curve in figure 15 shows the trade-off between type I and type II errors. Raising the threshold from 0.5 reduces the number of false positives, but at the same time increases the number of false negative. Ideally, the ROC curve should touch the top left corner because a perfect model should identify all positive observations without misclassifying any negative observations. Hence, the ideal AUC-ROC is 1. However, in this logistic regression, the AUC-ROC is 0.50. Thus, this is not a good prediction.

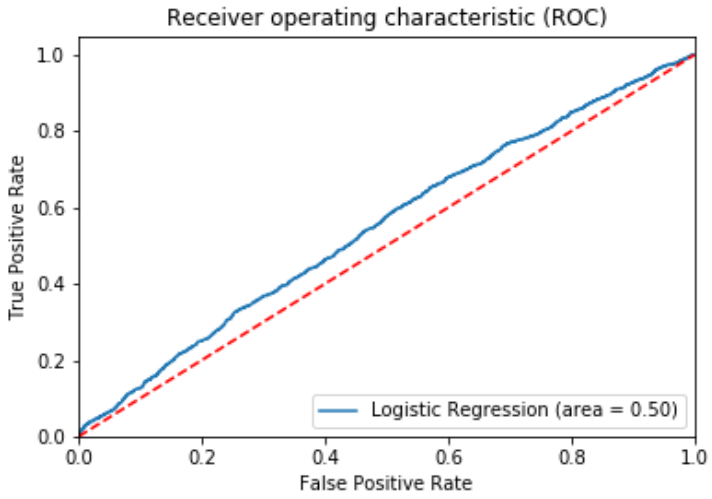


figure 15

After recursive feature elimination, the set of features is reduced to the number of capital letters, number of words, average number of characters in each word, and presence of inappropriate word(s). However, the performance of the model has not improved. The confusion matrix shows the significant presence of false negative.

		Predict	
		Real	**Fake**
Actual	**Real**	5025	4
	Fake	1613	0

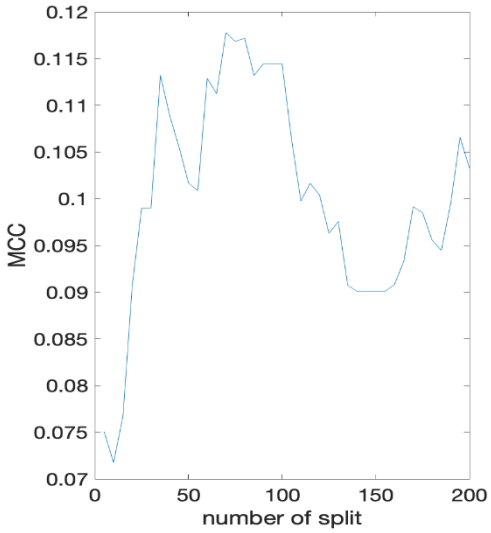
Accuracy = 75.67%
MCC = -0.0139

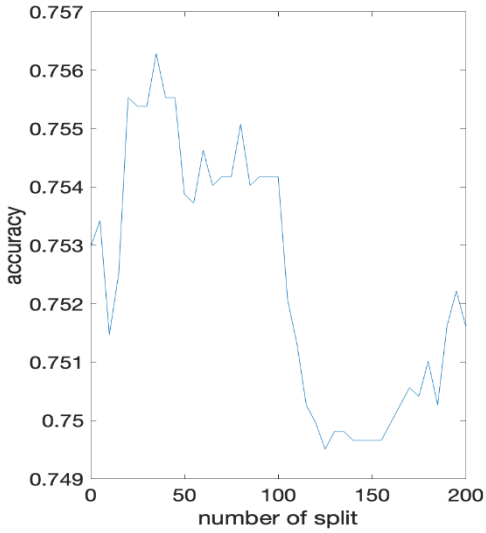
Again a high (76%) accuracy from this regression is only possible because about 76% of the observations are negative. This reflects the highly imbalanced nature of the data set. Therefore, given the poor performance of logistic regression, I am going to try other machine-learning models and resampling methods.

b. Classification Tree

The figure shows that a larger tree yields higher accuracy and MCC until the tree grows too large. The MCC peaks at 70 splits and starts to fall when the tree is larger. The curve is not necessarily smooth, because the model is trained to minimise error in the training set instead of maximising the MCC in the validation set.

figure 16 (number of splits against MCC)





(number of splits against accuracy) figure 17

Confusion matrix for unconstrained tree:

		Predict	
		Real	**Fake**
Actual	**Real**	4893	109
	Fake	1524	117

Accuracy = 75.42%
MCC = 0.1178

After setting the maximum number of splits to 70, the result of a tree is marginally better than the result of regression. After pruning and decreasing the depth from 14 to 8, I am able to increase the MCC slightly. Figure 18 shows the importance of each feature (predictor), which is estimated by summing changes in the risk due to splits on every feature and dividing the sum by the number of branch nodes. The percentage

of capital letters in words is by far the most importance feature. The number of capital letters and the number of characters in each title are also important.

Confusion matrix for tree after pruning:

		Predict	
		Real	**Fake**
Actual	**Real**	4937	65
	Fake	1547	63

Accuracy = 75.73%
MCC = 0.1250

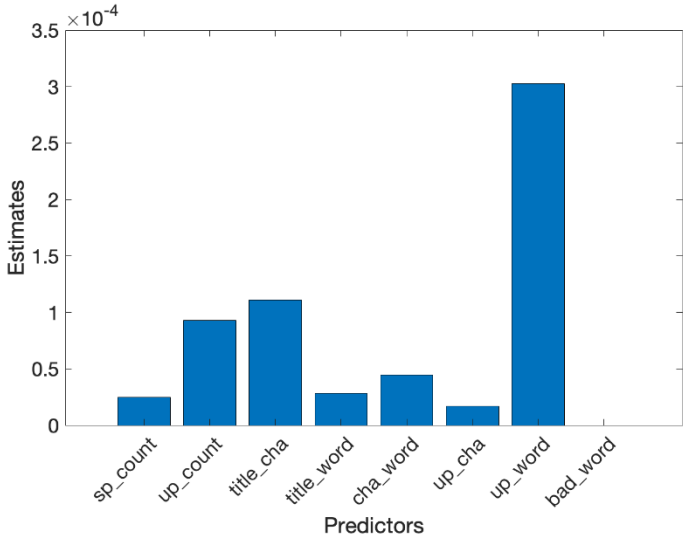


figure 18 (importance of each feature)

c. Random Forest

In figure 19, adding more trees in the forest drastically lowers the out-of-bag error at first, though when there are more than 60 trees this reduction of error becomes minimal. Therefore, I build a random forest with 100 trees. Comparing to a single decision tree, the forest produces a similar MCC even though the accuracy has fallen by a few percent.

		Predict	
		Real	**Fake**
Actual	**Real**	4488	514
	Fake	1308	333

Accuracy = 72.57%
MCC = 0.1295

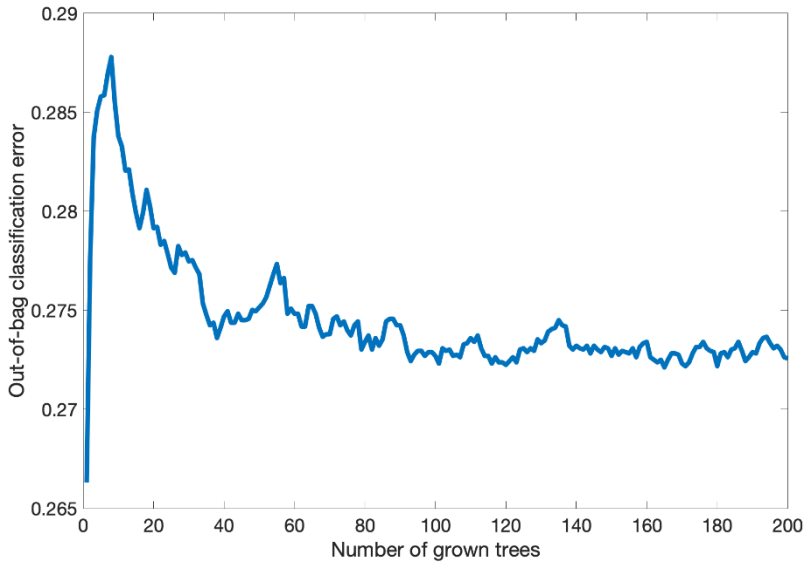


figure 19 (number of tree against out-of-bag error)

Another interesting result is that the importance of features has changed in comparison to a single decision tree (figures 18 & 20). The most important feature changes from the percentage of capital letter in terms of word. Apart from this feature and inappropriate words, the importance of other 6 features are comparable.

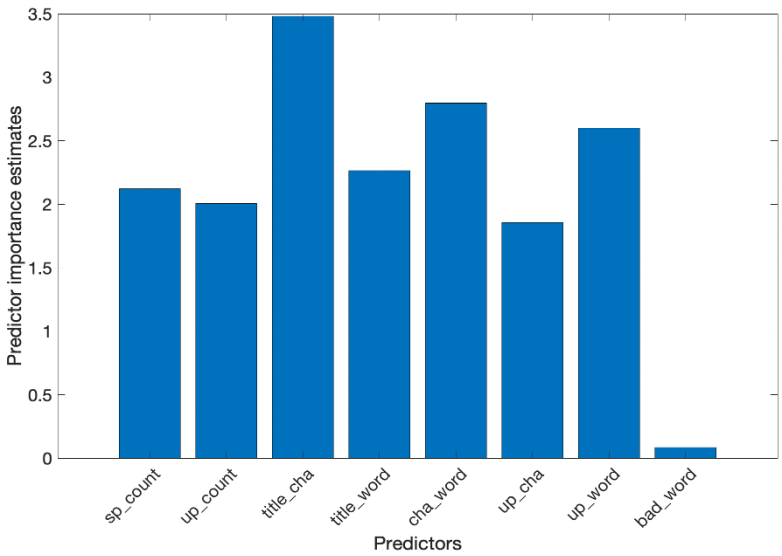


figure 20 (importance of each data in the forest)

d. Support Vector Machine

A Support Vector Machine (SVM) with Gaussian function produces a similar result to the tree-based models in section b & c. A 5-fold cross validation shows that the cross-validated classification error is 23.44%, as can be seen below.

		Predict	
		Real	**Fake**
Actual	**Real**	4969	33
	Fake	1577	64

Accuracy = 75.76%
MCC = 0.1165

Similar to the SVM model for political data, a soft margin in this case does not improve the out-of-sample accuracy.

It is apparent that none of the models implemented are able to classify gossip data accurately. It is possible the bad result is a result of imbalanced data. Therefore, I use SMOTE to increase the portion of fake news.

4.2.3 Gossip Data: SMOTE & Shrinking Real-news Dataset

After resampling, I now have two equally sized real- and fake-news gossip data sets. Since random forest produced the highest accuracy and MCC with the previous imbalanced gossip data set, I try this model with the new balanced data, as detailed below. It will be seen that after SMOTE, there is little improvements in either the accuracy or MCC. This suggests that the initial poor performance may not be the result of imbalanced data. Instead, the features needed to identify fake gossip news appear to be different from fake political news.

a. Random forest after SMOTE

After trying different number of trees, the result in figure 16 suggests that a small number of trees is needed for the random forest. Once the number of trees has gone past 100, growing more trees does not reduce the out-of-bag error significantly. I therefor build a random forest with 100 trees and test the model on the test data. The resulting MCC is better than the MCC without SMOTE, but by a small margin only. Similar to the original random forest, the number of characters is the most important feature in the SMOTE forest, closely followed by the number of special characters. Apart from inappropriate words, other features have similar importance.

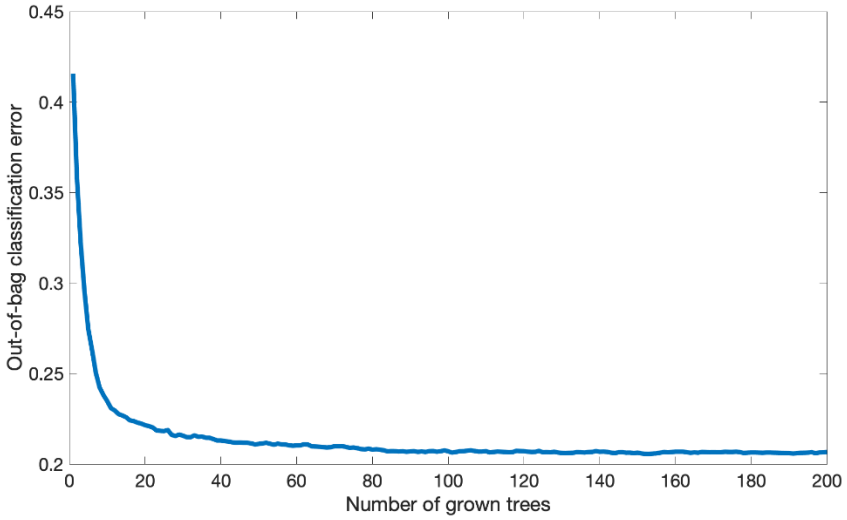


figure 16 (number of trees against out-of-bag error)

Confusion matrix for random forest with 100 trees (SMOTE):

		Predict	
		Real	**Fake**
Actual	**Real**	4174	864
	Fake	1096	508

Accuracy = 70.49%
MCC = 0.1535

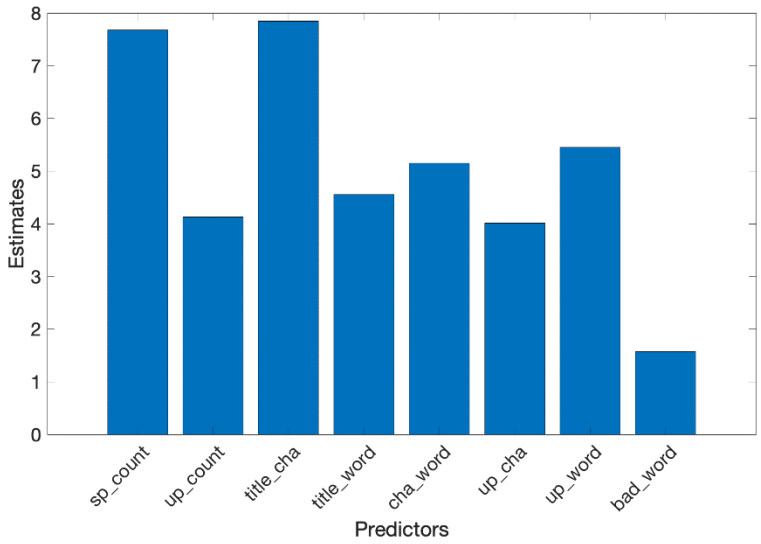


figure 17 (importance of each feature in the forest)

Afterwards, I limit the number of splits in each individual tree in order to avoid overfitting. The table below shows that the MCC has not improved because of this.

	Maximum number of splits								
	5	**15**	**25**	**50**	**70**	**90**	**110**	**150**	**200**
MCC	0.1345	0.1403	0.1437	0.1329	0.1367	0.1379	0.1521	0.1415	0.1415

b. Random forest after shrinking size of real-news dataset

I also try to shrink the size of the real-news dataset to 5323 observations, which is the same number as the fake news data. Since there are 16817 real-news data points, I split the real news into 3 subsets and pair each subset with all fake news. Hence, each of the 3 resulting datasets has similar amounts of fake news and real news. Unfortunately, the results are not promising. Using a random forest, the MCCs from 3 trials are 0.1330, 0.1264 and 0.1245 while the average accuracy is around 55%. One of the confusion matrices, shown below, shows that almost half of the data is false negative or false positive.

		Predict	
		Real	**Fake**
Actual	**Real**	905	715
	Fake	670	904

It can thus be seen that gossip news appears to be much harder to classify into real versus fake than political news. It is possible that real and fake gossip news closely resembles each other. One possible explanation is that gossip articles are often featured in tabloids, and people read tabloids for entertainment. As a result, readers of gossip news may not care about the grammars and wordings of the articles. Moreover, gossip websites rely on marketing revenue so both real and fake gossip-news articles are written in catchy styles.

Chapter 5. Conclusion

5.1 Conclusion

In this paper, I have discussed the current problems of fake news and suggested a potential solution to halt the spread of fake news. Since social-media users may not have the knowledge and time to fact check everything they read, fake news can easily cause readers to receive wrong information and make wrong decisions. For this reason, it is meaningful to build a tool that can spot fake news quickly. I suggest that the titles of fake-news articles are purposely written in a catchy style in order to attract readers; hence the grammars and structures of these titles are potentially different from those of real news. I use several machine-learning models such as logistic regressions, classification trees, random forests, and support vector machines to pick out such differences. At the moment, my data consists of political and gossip news. The preliminary results show that it is much easier to identify fake political news than fake gossip news. These results imply that one model does not fit all types of news. It is possible that online users read gossip news as entertainment, so the titles of both real and fake gossip-news articles are written to attract readers. As a result, even real gossip-news articles do not necessarily follow grammatical rules. Furthermore, in both cases of political and gossip news, my models sometimes misclassify fake news as real news. Thus, we should be cautious when interpreting the performance or applying the model. Nonetheless, this research can form the basis for the development of a fake-news filter which can analyse the news titles and filter out potential fake news.

5.2 Limitation and Future Work

In regard to future work, this research can be expanded from two domains to more domains. Fake news in other domains such as finance, science and health could cause adverse effects as well. Moreover, I can extend my research from a focus on titles to inspecting the whole articles. Normally, a title is only a short sentence and does not convey too much information about the content; consequently, it may be difficult to spot a fake-news article simply from looking at a dozen of words. Therefore, I can apply similar features, such as the use of special characters, and the existence of inappropriate words, to the contents of articles. Moreover, I can check whether the words in the titles appear again in the contents. Although the current data repository I

am using, FakeNewsNet, provides the URLs to the articles, downloading and cleansing all the articles will require a lot more time and computational power. Therefore, this idea can be explored in-depth in future research.

5.3 Final Thoughts

The results from this research show the difficulty as well as feasibility in spotting fake news. With the help of social media, writing and spreading fake news is extremely easy. At the same time, there are online tools that can generate sophisticated fake-news articles. Therefore, continuous research should be conducted so that online communities can possess tools to prevent the spread of fake news. Fake-news identification is definitely a meaningful topic for future research because of the harms brought by fake news.

Reference

"Fake news" is 2017 American Dialect Society word of the year. (2018, January 5). Retrieved from American Dialect Society: https://www.americandialect.org/fake-news-is-2017-american-dialect-society-word-of-the-year

Aldwairi, M., & Alwahedi, A. (2018). Detecting Fake News in Social Media Networks. *Procedia Computer Science Volume 141*, 215-222.

Arkaitz Zubiaga, M. L. (2015). Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *arXiv:1511.07487*.

Buntain, C., & Golbeck, J. (2017). Automatically Identifying Fake News in Popular Twitter Threads. *arXiv:1705.01613*.

Cambridge Dictionary. (n.d.). Retrieved from fake news: https://dictionary.cambridge.org/dictionary/english/fake-news

Choen, S., Li, C., Yang, J., & Yu, C. (n.d.). Computational Journalism: A Call to Arms to Database Researchers. *https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37381.pdf*.

FakeNewsNet. (2019, August 13). Retrieved from GitHub: https://github.com/KaiDMML/FakeNewsNet

Karimi, H., Roy, P., Saba-Sadiya, S., & Tang, J. (2018). Multi-source multi-class fake news deteection. *27th International Conference on Computational Linguistics*, 1546-1557.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Long, Y., Lu, Q., Xiang, R., Li, M., & Huang, C.-R. (2017). Fake news detection through multi-perspective speaker profiles. *Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 252-256.

Mitra, T., & Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. *ICWSM*, 258-267.

Nakashole, N., & Mitchell, T. M. (2014). Language-aware truth assessment of fact candidates. *52nd Annual Meeting of the Association for Computational Linguistics (Volume1: Long Papers)*, 1009-1019.

Oshikawa, R., Qian, J., & Wang, W. Y. (2018). A Survey on Natural Language Processing for Fake News Detection. *arXiv:1811.00770*.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). Automatic Detection of Fake News. *https://www.aclweb.org/anthology/C18-1287*.

Rashkin, H., Choi, E., Jin, Y., Volkova, S., & Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. *2017 Conference on Empirical Methods in Natural Language Processing*, 2931-2937.

Rayson, S. (2018, May 22). *Content Trends: How Articles About Fake News Rocketed After Trump's Election*. Retrieved from BuzzSumo: https://buzzsumo.com/blog/content-trends-how-articles-about-fake-news-rocketed-after-trumps-election/

Rubin, V. L., & Vashchilko, T. (2012). Identification of truth and deception in text: Application of vector space model to rhetorical structure theory. *Workshop on Computational Approaches to Deception Detection*, 97-106.

Ruchansky, N., Seo, S., & Liu., Y. (2017). Csi: A hybrid deep model for fake news detection. *2017 ACM on Conference on Information and Knowledge Management*, 797-806.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018, September). *FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media.* Retrieved from https://arxiv.org/abs/1809.01286

Silverman, C., Lytvynenko, J., & Pham, S. (2017, December 28). *These Are 50 Of The Biggest Fake News Hits On Facebook In 2017*. Retrieved from BuzzFeed News: https://www.buzzfeednews.com/article/craigsilverman/these-are-50-of-the-biggest-fake-news-hits-on-facebook-in?bfsource=relatedmanual

Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., & Mittal, A. (2018). The Fact Extraction and VERification (FEVER) Shared Task. *arXiv:1811.10971*.

Vasandani, J. (2019). *Using NLP and Classification Models to distinguish between fake news and absurd news*. Retrieved from GitHub: https://github.com/jasminevasandani/NLP_Classification_Model_FakeNews

Vlachos, A., & Riedel, S. (2014). Fact Checking: Task definition and dataset construction. *https://www.aclweb.org/anthology/W14-2508*.

Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Appendix

A. All codes and data are at: https://github.com/chaklam14/fake-news-detection.git

B. Logistic regression for political data:

Results: Logit

Model:	Logit	Pseudo R-squared:	0.189
Dependent Variable:	class	AIC:	1174.9098
Date:	2019-09-04 23:08	BIC:	1214.6077
No. Observations:	1056	Log-Likelihood:	-579.45
Df Model:	7	LL-Null:	-714.41
Df Residuals:	1048	LLR p-value:	1.5889e-54
Converged:	1.0000	Scale:	1.0000
No. Iterations:	7.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
sp_count	0.0253	0.0528	0.4795	0.6316	-0.0781	0.1287
up_count	0.3683	0.0363	10.1404	0.0000	0.2971	0.4394
title_cha	-0.0461	0.0155	-2.9764	0.0029	-0.0765	-0.0158
title_word	0.1046	0.0753	1.3894	0.1647	-0.0430	0.2523
cha_word	0.0024	0.0363	0.0650	0.9482	-0.0689	0.0736
up_cha	-0.1020	0.0351	-2.9077	0.0036	-0.1708	-0.0332
up_word	-0.0057	0.0061	-0.9281	0.3533	-0.0177	0.0063
bad_word	0.4426	0.9464	0.4677	0.6400	-1.4123	2.2975

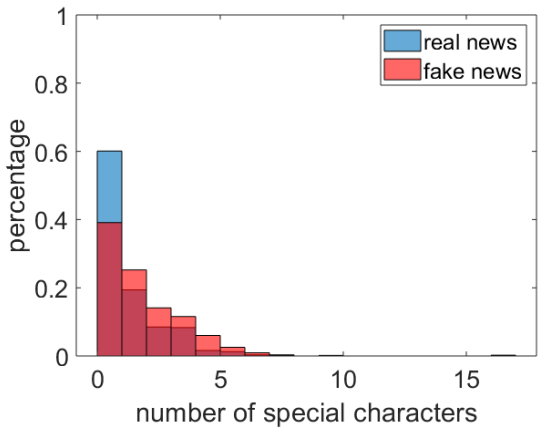
C. Logistic regression for gossip data:

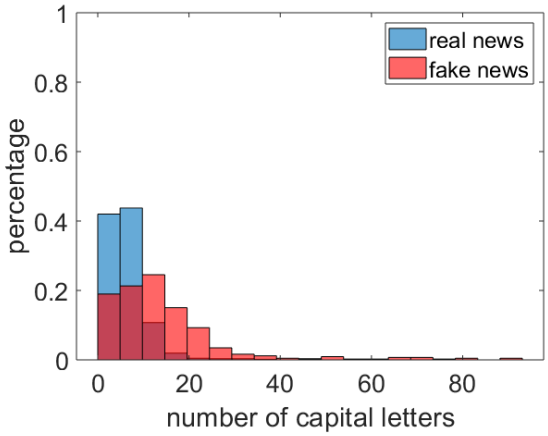
Results: Logit

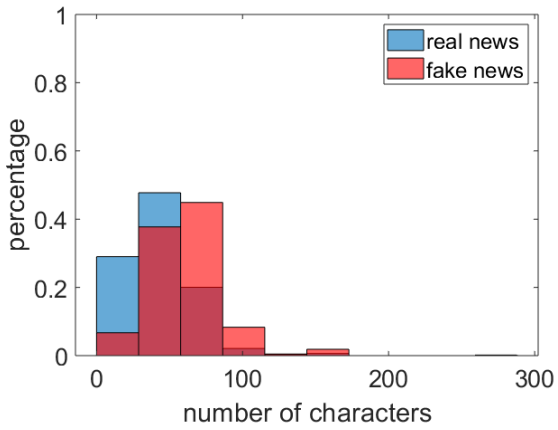
Model:	Logit	Pseudo R-squared:	0.005
Dependent Variable:	class	AIC:	24313.5877
Date:	2019-09-03 23:08	BIC:	24377.6288
No. Observations:	22140	Log-Likelihood:	-12149.
Df Model:	7	LL-Null:	-12212.
Df Residuals:	22132	LLR p-value:	4.5557e-24
Converged:	1.0000	Scale:	1.0000
No. Iterations:	5.0000		

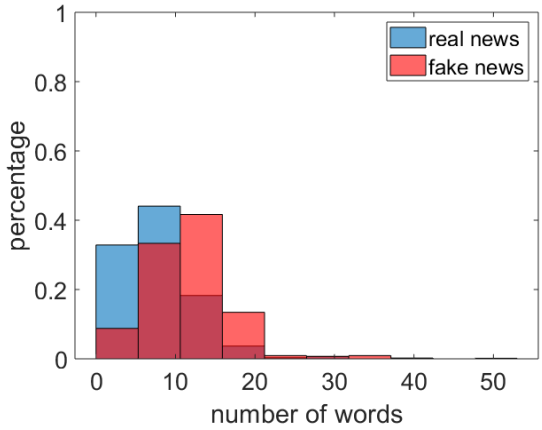
	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
sp_count	-0.0040	0.0109	-0.3657	0.7146	-0.0253	0.0173
up_count	0.0286	0.0101	2.8322	0.0046	0.0088	0.0484
title_cha	0.0150	0.0041	3.6718	0.0002	0.0070	0.0230
title_word	-0.1079	0.0186	-5.7993	0.0000	-0.1444	-0.0714
cha_word	-0.1992	0.0202	-9.8482	0.0000	-0.2389	-0.1596
up_cha	-0.0666	0.0120	-5.5681	0.0000	-0.0901	-0.0432
up_word	0.0123	0.0025	4.8811	0.0000	0.0074	0.0173
bad_word	-0.1339	0.1706	-0.7846	0.4327	-0.4683	0.2005

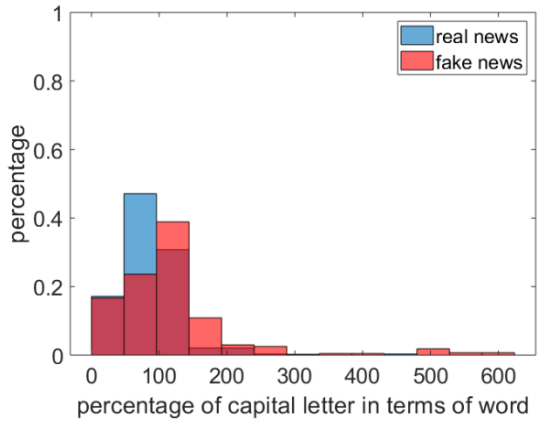
D. Data distribution for features of political news

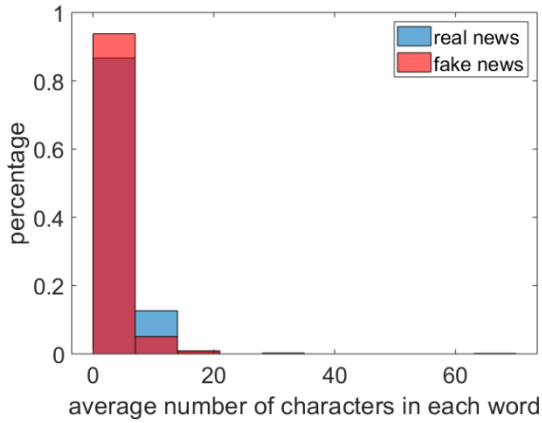


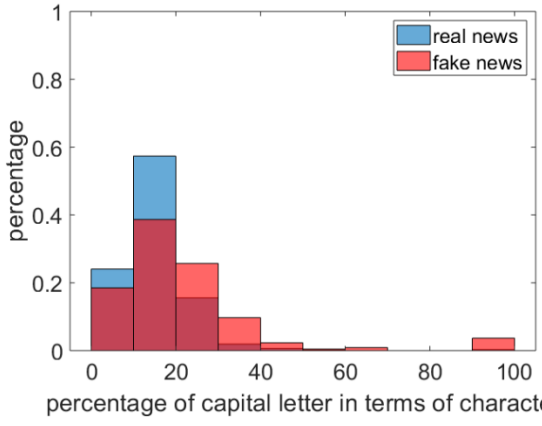




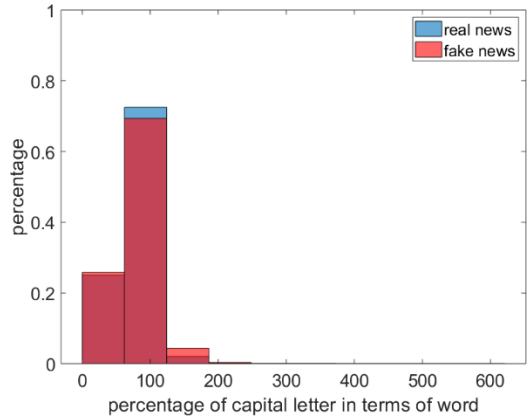


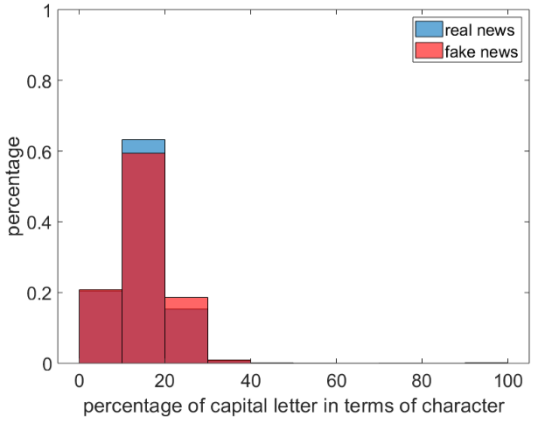


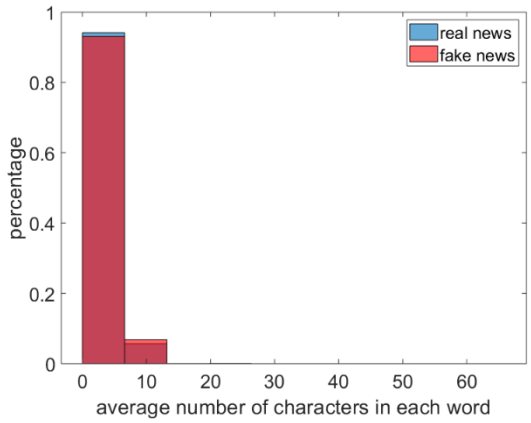


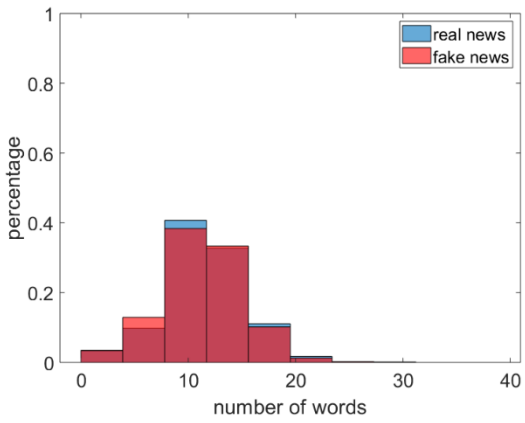


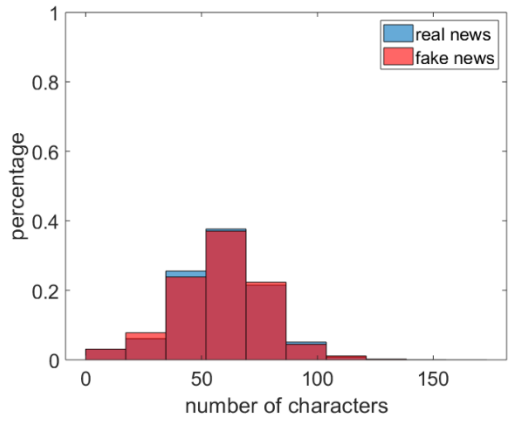
E. Data distribution for features of gossip news:

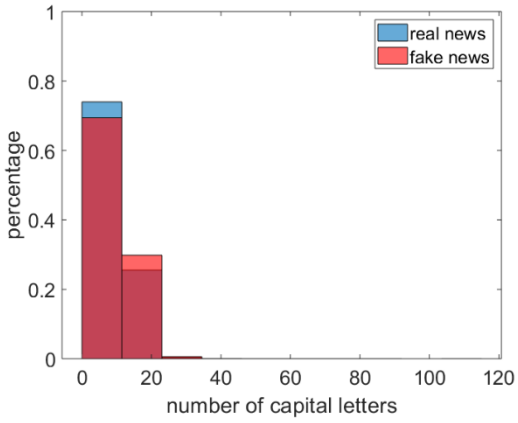


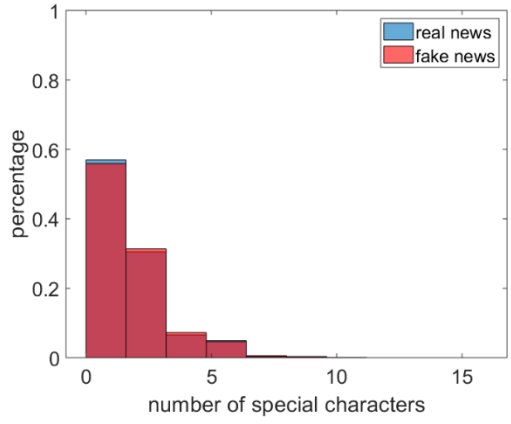












MSc Computational Finance Project Dissertation 2019

Project Summary

Project Title: Fake News Detection

Stakeholders: Technology Department, EY

Industrial Supervisors:

Emma Birchenough-Dwyer(emma.dwyer@uk.ey.com) – Senior Manager, Risk & Compliance Technology

Tom Wilmots (tom.wilmots@uk.ey.com) – Financial Services Technology Consultant

Erika Amelia (erika.amelia@uk.ey.com) – Management Consultant

Team Members:

Liyah Dholiwar (liyah.dholiwar.18@ucl.ac.uk)

Natalie Gapp (natalie.gapp.18@ucl.ac.uk)

Antigone Kyriakide (antigone.kyriakide.15@ucl.ac.uk)

Zhixuan Qu (zhixuan.qu.18@ucl.ac.uk)

Wenjia Zhang (wenjia.zhang.18@ucl.ac.uk)

Project Outline:

Objective 1: Identify the ways to spot fake news for example: sentiment analysts- is it overly emotional, fact check – stats/ dates/ numebrs etc, look for duplication of articles are there discrepancies across publications.

Objective 2: Apply machine learning to categorise each article as True or False. Build an app that anyone can use to fact check media via different platforms and content sources.