UNIVERSITY of STIRLING

Division of Computing Science and Mathematics
Faculty of Natural Sciences
University of Stirling

Detecting toxicity in online discussion

Helen Graham

Dissertation submitted in partial fulfillment for the degree of
Master of Science in Big Data

September 2018

Abstract

Problem: There is growing concern that a toxic culture in online discourse presents a barrier to diverse participation in the digital world. Those who operate social media platforms or online knowledge repositories may seek to detect and moderate toxic content, in order to avoid losing members of their user communities, or discouraging potential users from participating in the first place. Given the rapid flow of information being contributed to such platforms, some degree of automation would assist the task of moderation.

Objectives: This project aimed to create a classifier that can detect toxicity in online comments, based on the words they contain, and other features, such as the sentiment they convey.

Methodology: A pre-labelled dataset, the Wikipedia Detox dataset, was used. This contains around 150,000 comments taken from article and user talk pages on Wikipedia, and annotated for whether they are toxic, defined in this case as whether they would make someone want to leave a conversation. Features were extracted from the comments; a vector of the words used in the comment (transformed via principal components analysis), along with other characteristics such as the mean sentiment score, and the presence of features such as repeated punctuation or capital letters that might indicate that a hostile tone is being used. These features were used together with the toxicity labels to train machine learning models. Four algorithms were used: regularised logistic regression, random forest, naïve Bayes and support vector machines (SVM).

Achievements: The work presents a contribution to identifying toxic content online. The classifiers built using the logistic, random forest and SVM algorithms achieved a reasonable level of success in predicting whether a comment was toxic or not. All three had similar areas under the resulting ROC curves (89-93%) and F1 scores (92-95%). The models had different strengths; the logistic model was the better at successfully identifying toxic posts, but the random forest and SVM models were less likely to erroneously classify a non-toxic post as toxic. Areas identified for future work included improved detection of sarcasm, and the use of more cutting edge word embedding methods. However the work also illustrates the inherent difficulty in classifying a subjective phenomenon, and the reliance of the models on consensus in classification. This is challenging to reconcile with a context in which a minority may be alienated by behaviour that is not considered toxic by a majority.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project, except in the following cases where the code used to carry out this analysis draws on, adapts, personalises and extends examples from books, online courses or tutorials:

- The tokenization and visualisation of text data (Section 3.2) was in part adapted from examples in Silge and Robinson's *Tidy Text Mining* book [1] and from the courses 'Text Mining: Bag of Words' and 'Sentiment Analysis in R: The Tidy Way' provided by the online training provider Data Camp (www.datacamp.com)
- The implementation of machine learning in the `caret` package in R (Section 3.3.2) was in part adapted from examples in Data Camp courses 'Machine Learning Toolbox' and 'Supervised Learning in R: Case Studies'
- The implementation of PCA was guided by two useful blog posts [2] [3]

Signature

Date

Acknowledgements

Firstly, my thanks go to my supervisor, Carron Shankland, for steering me in the right direction.

Thanks also to Amy Isard, who gave me a really useful heads up about some of the literature in this area.

Undertaking an MSc in Big Data was a bit of a change of direction in my life, and I want to thank my partner Jeff for going with the flow and believing in me.

Finally, I want to dedicate this dissertation to… Stack Overflow! It can certainly be a toxic place a lot of the time (and I applaud the efforts being made to change this), but goodness knows how I would have written this dissertation without it.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Background and context to the problem

Digital technology is a rapidly growing sector of the UK labour market, with employment in this sector increasing by 13.2% between 2014 and 2017, and the opportunities provided by the sector are potentially lucrative, with an average salary of £42,578 compared with £32,477 in non-digital jobs. However, the sector remains dominated by men; just 19% of workers in the sector are female [4].

One factor that has been consistently cited as a barrier to women's participation in the tech workforce is a negative culture that alienates and excludes them. A survey of 600 tech industry professionals by IT recruitment consultancy Harvey Nash found that 29% of female respondents reported experiencing an unwelcoming work environment, compared with 7% of male respondents [5]. Women who work in the sector report that this manifests itself in various ways, from having their competency questioned in a way that their male colleagues do not, to the presence of scantily clad models at industry events [6]. This issue of exclusion extends beyond workplaces to exclusion from digital spaces more generally. Megarry [7] argues that, just as street harassment constrains women's use of public spaces, the harassment they receive online is a form of exclusion. Digital sexism, in which aggressors try to intimidate, shame and discredit women's contributions to digital public spaces, constrains what women can talk about online and how they talk about it [8][9][10][11].

Women's exclusion from digital workplaces and online spaces extends to the open source community and other sources of online 'volunteering'. Concern about this issue has been raised recently by two key players in this area; online encyclopaedia Wikipedia, and developer community Stack Overflow. Both of these are male dominated spaces. In the 2018 Stack Overflow survey of its developer members, 93% of respondents were male [12]. A study that tried to infer the gender of Stack Overflow users from their names, pictures and associated websites found only a slightly higher proportion of 12% female users, and they also found lower levels of engagement with the site; female users ask and answer fewer questions, and have fewer reputation points [13]. Similarly, only around 1 in 10 of those who write and edit articles on Wikipedia are women [14].

Both Wikipedia and Stack Overflow are key online repositories for knowledge. Who contributes to and curates these sites is pertinent because the content will be shaped by – and reflect the biases and worldviews of – the user communities. Both organisations have expressed concern that an unacceptably high level of toxicity in discussions on their platforms might be alienating some users, and in particular those from under-represented groups. Stack Overflow recently acknowledged [15] that the environment on their platform was unacceptably poor:

> *Too many people experience Stack Overflow as a hostile or elitist place, especially newer coders, women, people of color, and others in marginalized groups.*

In effect, they are concerned that the culture among their site users could be putting off new entrants to the sector, in particular those from under-represented groups. In response, they decided to conduct further research into this issue. They asked their staff to rate a sample of postings, identifying those that were [16]:

> *...unwelcoming in a way that isn't flagrant hate or abuse but would still make you think twice about participating in our community... [this might] include condescension, snark, sarcasm, and the like.*

57 staff members rated 3992 comments, of which 0.3% were outright abusive and 7.4% fell into this 'not abusive but unwelcoming' category. A typical example of such a comment was: *"This is becoming a waste of my time and you won't listen to my advice. What are the supposed benefits of making it so much more complex?"*. Stack Overflow intend to use the data from their rating exercise to inform a "human-in-the-loop machine learning" solution to addressing the issue.

In a similar piece of work, the Wikipedia detox project[1] is currently being undertaken by Wikipedia owner the Wikimedia Foundation, in conjunction with technology incubator Google Jigsaw. It was initiated in response to concerns about the impact of abusive behaviour on the participation and retention of Wikipedia editors. It aims to understand the nature and impact of this behaviour, and develop tools for detecting it. As part of the project, a corpus has been created of comments made on the user and article talk pages of Wikipedia, which have been annotated for the presence of personal attacks, aggression, and toxicity (defined as an unpleasant comment that makes you want to leave a discussion). 11.7% of comments were found to fall into this category [17]. The data has been used by Wikimedia to build a classifier to detect attacks and aggression, and a prototype is available that takes an input of text and estimates the probability that it contains an attack or aggression.

The backlash against Stack Overflow's work can be seen in the Stack Exchange Meta discussion boards. There are a number of recurring themes in the critique; denial that there is a problem [18], requests for better evidence because what has been presented is considered too anecdotal [19], and concern trolling (derailing discussion of equality issues with concern that doing anything to address them might reduce the quality of the product) [20]. Previous attention to Wikipedia's gender gap resulted in a similar backlash in the media and online, with commenters denying that this was a problem and blaming women for not participating [14]. This backlash suggests that toxicity is in part a form of gatekeeping, driven by those whose identities feel under threat by the opening of 'their' domain to people not like them [21][22][23].

1.2 Scope and objectives

In light of the way that toxicity presents a barrier to diverse participation in the digital world, intervention on the part of those who operate online platforms is required if they aspire to make that world less intimidating to under-represented groups. Given the fast flow of information being contributed, some degree of automation would assist this task. This research therefore aims to create a classifier that will detect toxicity in online postings.

This work is an example of supervised learning. Each case has a set of measurements on some predictor variables, and an associated outcome label, and the aim is to fit a model that relates the outcome to the predictors, with the ultimate aim of being able to predict the response on future, unlabelled cases [24]. This is in contrast to unsupervised learning, where there is no outcome label available, and the aim is to discern underlying patterns, clusters or relationships within the data.

[1] https://meta.wikimedia.org/wiki/Research:Detox

The type of analysis undertaken here comes under the various headings of text mining, natural language processing (NLP) and sentiment analysis. NLP is often seen as a deeper or more advanced version of text mining [25]:

> *Text mining is the discovery and extraction of interesting, non-trivial knowledge from free or unstructured text... Natural language processing (NLP), is the attempt to extract a fuller meaning representation from free text.*

This work is attempting to estimate the toxicity of a piece of text, which could be understood as its meaning; is it benign and neutral, or hostile towards its intended recipients, and what makes it so. The work also encompasses elements of sentiment analysis [26]:

> *...the field of study that analyses people's opinions, sentiments, evaluations, appraisals, attitudes and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes.*

The work considers whether the sentiment behind a comment is toxic, and at a more basic level to what extent sentiments such as negativity and anger are indicative of a toxic post.

The standard methodology of a text mining study [27] effectively defines the sub-objectives of the project:

1. Define and understand the problem of toxic behaviour online, by considering previous approaches to the detection of similar phenomena
2. Obtain some suitable data to investigate the problem
3. Clean the data and extract the relevant features for use in a machine learning model
4. Train a model to classify comments as toxic or not and evaluate its success
5. Use the best model obtained to provide assistance to those engaged in moderating online discussion, by offering a credible estimate of toxicity on unseen text

1.3 Achievements

This work attempts to address the issue of toxic behaviour discouraging online participation, by contributing towards efforts to automate the detection of such content. In doing so, it adds to a growing literature on the detection of unpleasant behaviour online.

The work presents an analysis of a relatively new dataset that is yet to be explored to its full potential, and builds on rather than replicates previous work. Classifiers are built that are relatively successful at predicting the outcome of interest on the type of data on which it is trained. The success of the classifier is derived in part from using both the text of a comment itself, and 'metadata' about the comment, as features in the model. The feature set used in the modelling process takes this text classification exercise beyond a basic 'bag of words' approach, to incorporate the analysis of sentiments and other linguistic features that represent something about the way the author of a comment is expressing themselves, such as the use of capital letters or punctuation that might indicate a particular tone.

However the work also exposes some of the pitfalls inherent in classifying this type of comment, such as the difficulties of detecting subtleties such as sarcasm and indirect insults, and the more fundamental difficulty of building a classifier around a subjective outcome. It also highlights the way that classifiers may struggle to perform well in a domain other than that in which they have been trained.

1.4 Overview

The remainder of this dissertation is structured as follows:

- Chapter 2 presents a summary of recent research using supervised machine learning to classify text from online discussions for the presence of unpleasant features.
- Chapter 3 presents the data that was used in the analysis, and explains how it was cleaned and prepared for use in a machine learning model. It also outlines the machine learning algorithms that were used, and how the models were built and evaluated.
- Chapter 4 gives an overview of the data before presenting the results of the machine learning process.
- Chapter 5 concludes the dissertation with a summary and discussion of its findings.

2 State of the art

In the work carried out by Stack Overflow and Wikimedia on understanding online toxicity, both went about the task in a similar way; making an annotated corpus and training a classifier to detect the outcome of interest. This work is located in a wider body of contemporary research that uses a supervised machine learning approach to detecting unpleasant behaviour online. This review considers the data sources and methods that others have used to go about this task, in order to inform the development of the methodology in this project.

The scope of the review here is text classification problems within the domain of interest; unpleasant online speech that might upset or alienate someone. Therefore this review considered literature that applies machine learning techniques to the detection of online abuse, sarcasm, nastiness, impoliteness, insults, hate speech, bullying, attacks, aggression and toxicity. It considers how previous researchers have chosen their datasets, features and algorithms, and how this has informed the work undertaken in this project.

2.1 What data has been used?

Some researchers, in looking at phenomena such as online abuse, have looked for data in places that there is likely to be discussion and conflict, such as Twitter [28][29] and online news discussion boards [30][31]. However, not all have sought out obvious sites of conflict; other sources include question and answer sites such as ask.fm and Stack Overflow [32][33] and Wikipedia talk pages [17][34]. The experience of toxic conversation even in arenas where it might not be expected is perhaps the more interesting phenomenon to investigate, because it is the ubiquity of this culture that makes it such a pressing issue; it cannot be avoided in any way other than opting out altogether.

Having chosen a site of interest, the next question is how to generate a supervised learning corpus from the raw data. Examples (e.g. comments, tweets, discussion board postings, or parts thereof) are labelled as having or not having a characteristic of interest (e.g. abuse, sarcasm). Sometimes a working definition is agreed beforehand, for example Waseem and Hovy [35] define a tweet as offensive if it contains at least one of a list of eleven features, including sexism, racism, the promotion of hate speech, or the deliberate distortion of the truth about a minority group. Conversely, Danescu-Niculescu-Mizil et al. [36] simply ask annotators to indicate a point on a continuous sliding scale between ‘very impolite’ and ‘very polite’.

When it comes to the process of annotation itself, some have chosen to do their own labelling (or task a student Research Assistant with this job) [37][33][35][38]. Others have chosen to use a crowdsourcing platform [28][30][36][31]. Using a crowdsourcing platform has some advantages over attempting to annotate a corpus within a small research team. Labelling is a time consuming task, so by outsourcing it, the researcher is left with more time to build and tune their models, to make them as good as possible. It allows the work to be spread across more annotators; few researchers will have the dozens, or potentially hundreds of annotators at their disposal that can together create a very large corpus. Opening up the exercise to a broader cross-section of people also allows a rating to be reached by consensus between those who do not necessarily share the same world views. With too few annotators, the results of a subjective exercise such as rating a comment as abusive or not may end up reflecting the biases and prejudices of the annotators.

However, to outsource this stage of the research in this way is effectively to conceptualise this classification process as low-skilled drudge work, when arguably it is paramount because the quality of these classifications will underpin the success or failure of the models. Crowdsourcing does raise issues of quality. Almost anyone can participate in the process, and their motivation is not the ultimate success of the classifiers the researcher will build, but rather to annotate as many examples as possible, as quickly as possible, to maximise the income they receive from the task. For this reason, some degree of quality control is required, for example requiring annotators to rate test examples in line with the broad consensus on these before being allowed to contribute their own ratings [17].

Because these hand-annotated corpuses are labour-intensive to create, they are often deposited publicly for reuse. For example Zimmerman et al. [39] reuse the corpus created by Waseem and Hovy [35] referred to above. Joshi et al. [40] use a publicly available corpus called the Internet Argument Corpus, while Binns et al. [34] use the Wikipedia detox corpus that was created by Wulczyn et al. [17], that also forms the basis of the analysis in this project.

Those using data from Twitter may choose to use hashtags as a ready-made label, for example using tweets tagged as sarcastic in trying to classify sarcasm [29][41][42][43]. This approach, in addition to being quicker than manual labelling, has the benefit that the author of the text has explicitly signalled their intent, thus removing the subjectivity of asking a third party to classify the tweet. However, it is likely that some tweets will be sarcastic, but not classified as such in the corpus because they do not have the hashtag, and therefore represent false negatives within the corpus itself. It also relies on a high level of accuracy in the use of the sarcasm hashtag, when in fact hashtags can be very ‘noisy’ [41].

2.2 What features have been extracted from the text?

The starting point in every case is to use the words in the text as features. This may be achieved by simply counting the frequency with which each occurs, but more complex analyses also consider their relation and proximity to each other, and information about them; what type of words they are, and whether they convey any particular opinion, orientation or sentiment.

The first step in turning text into features is to define the unit of interest. In many cases this is simply the word, or unigram. However, multiword units may also be considered in order to take into account combinations of words that describe a specific concept [44]. Some studies have restricted their analysis to unigrams [45], while others have considered bigrams (two words) [30], or trigrams (three words) [46]. Although the inclusion of these larger units can add distinct concepts to the feature space, it also dramatically increases the size of the feature set.

In the opposite direction, some researchers consider the frequency of character ngrams; sequences of n characters or more, even if they do not represent full words [47][35][17]. This has been found to be quite a successful approach, because it can capture the essence of a word with different spellings or conjugations. For example Waseem and Hovy [35] find that ngrams such as ‘sla’, ‘slam’ and ‘isl’ are highly indicative features for detecting racism, because they will be present in many different words pertaining to Islam (Islam, Islamic, Islamist, etc.). This is akin to stemming, a text preparation method that reduces similar words to a common stem, so that they are not treated as independent concepts.

Having defined the word or character ngram of interest, the simplest approach to turning these into features that can be used in a model is to create a binary word vector for each case. All the words present across the whole corpus are taken to be the vocabulary of the corpus, and each case is represented by a binary string with a 1 if the corresponding vocabulary word is present in that particular case, or a zero if it is not. It is also possible to represent words by the number of times they occur, rather than simply whether they occur or not. However, a more common approach is to represent the number of occurences (the term frequency, or TF), multiplied by the inverse of the word's frequency in the corpus as a whole (its inverse document frequency, or IDF). The resulting TFIDF therefore increases as a term's frequency within a case increases, but is offset if a term is very common in the corpus as a whole; this means that the highest scores are terms that occur frequently within a single case, but infrequently across the corpus as a whole, and thus potentially provide more information [27].

Given that most of the vocabulary will not occur in any given case, the vectors that result from this exercise are likely to be extremely sparse; i.e. contain mostly zeros. This raises the question of whether to undertake dimensionality reduction to reduce the number of features, and if so how. Some analyses have just used all of the available ngrams; for example Buschmeier et al. [45] use every distinct word, along with other features, and end up with a set of almost 22,000 features. Others employ feature selection prior to model training; for example, Sahay et al. [37] pick the words that best explain the outcome of interest using SelectKBest feature selection, which conducts a chi squared test of association between the outcome and each feature, and selects only the k features that are most strongly associated. A third option is to use a machine learning model that incorporates feature selection in the way it works, such as penalised regression, which weights irrelevant features down to zero [43].

An alternative approach to dealing with high-dimensional word vectors is to project them onto a smaller feature space. This not only reduces the need to discard information, but also has the additional advantage of being able to capture some of the inter-relationships between the words. One method that has been used in the literature to perform this task is word2vec [28] [47], which takes into account the distances between words. This approach represents words as a location in pre-defined multidimensional space, which has a much smaller number of dimensions than the number of words (Chatzakou et al. [28] use a space with 300 dimensions, Nobata et al. [47] with 200 dimensions). This space itself needs to be trained in a separate unsupervised learning process, but pre-trained embeddings are available for use by researchers who do not wish to undertake this step. Similar approaches used in the literature include GloVe embeddings [31] and paragraph2vec [48].

In addition to using ngrams or word embeddings, some look for specific words or combinations of words that might be particularly indicative of their outcome of interest. For example, Chatzakou et al. [28] look for specific hate or curse words from a crowdsourced list, in their endeavour to identify instances of cyberbullying. Justo et al. [46] look for specific phrases relating to sarcasm (e.g. "I'm so sure", "oh yeah") and nastiness (e.g. "your ignorance is", "nonsense", "idiot"). Dadvar et al. [38] look for profanity and the use of second person pronouns to detect hate, while Danescu-Niculescu-Mizil et al. [36], in classifying politeness, look for the display of specific politeness strategies such as gratitude, deference, hedging and apologising.

Other specific words of interest may be those that represent the emotional tenor of a comment, and sentiment analysis can identify these. This involves identifying the presence of positive, negative or other emotions, or in some cases the strength of these emotions, by cross-reference of the corpus against a lexicon that contains these words and a corresponding category or score. In applied work, a researcher would be unlikely to spend time creating a lexicon from scratch, but rather use one of a number of established lexicons. For example, Danescu-Niculescu-Mizil et al. [36] and Buschmeier et al. [45] use Hu and Liu's [49] lexicon, which classifies words as positive or negative, while Chatzakou et al. [28] use a tool called SentiStrength [50], which assigns a score between -5 and +5.

The presence of such words can be incorporated into a feature set in a number of ways. One possibility is to construct an indicator of the number of times particular semantic features occur, or how much they occur on average [33]. Another is to look for the presence of multiple sentiment words together, which may be particularly indicative of an emotional orientation [45]. The presence of emoticons may also be a clue about the mood the author was trying to convey [28][41][40]. Certain features of the way people use language have also been used as potential indicators of aggression or other emotions in trying to ascertain the tenor of a comment. For example the use of capital letters [28][40], unusually intense or frequent use of punctuation [45] or repeated letters in words [43].

A final feature type is the use of meta information about a comment or tweet, or about the person posting it. For example, the time at which a tweet is posted might be useful if particular types of online interaction are more common at certain times of day, or those posting from verified accounts may be less likely to engage in toxic behaviour if they are at greater risk of real-life repercussions [28]. Information about the length of posts, sentences and words may be useful if those engaging in certain types of behaviour are more likely to communicate in a particular manner [46][35]. How well (or badly) users and posts have been rated by other users of a site may also be useful contextual information [32][51].

2.3 Which algorithms have been used?

A range of machine learning algorithms have been deployed in these classification exercises across the literature. Model choice is often made without a great deal of explicit rationale in the literature reviewed here, which is predominantly constituted of short journal articles and conference papers that preclude a detailed discussion of these issues. However, each have features that make them valid options for the type of problem or dataset at hand.

2.3.1 Logistic regression

A popular starting point is a logistic regression model, due to its relative simplicity, speed and ubiquity, although it is often followed by the implementation of a more complex model that is expected to perform better [37][41][45][35][34][17][48].

Logistic regression predicts the probability that a categorical variable takes a particular value, based on one or more predictors. In many cases in the literature, there are only two categories – whether a comment has a given property or not – and the model is predicting the probability that a comment has this property. Logistic regression extends the basic linear model where outcome y is some linear function of a set of predictors X (Equation 2.1).

$$y = \beta_0 + \beta_1 X \tag{2.1}$$

Because probability is bounded between 0 and 1, but equation (1) could give a result outside of this, the logistic function restricts outputs to between 1 and 0 (Equation 2.2).

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \tag{2.2}$$

After this transformation, it is the log odds of probability, rather than the probability itself, that is a linear combination of predictor variables (Equation 2.3).

$$log(\frac{p(X)}{1 - p(X)}) = \beta_0 + \beta_1 X \tag{2.3}$$

The values of β_0 and β_1 are estimated using a method called maximum likelihood. These parameters are estimated such that the predicted probability, if you substitute these numbers into the model, is as close as possible to the actual outcome [24].

When the set of predictors is large, they are unlikely to all be important. The aim is to find the most parsimonious model, both for ease of interpretability, and to prevent overfitting, to maximise prediction accuracy on new data. Rather than using a subset of variables, penalised regression uses all the predictors and reduces the size of the coefficients. A number of estimates of the coefficients are made, from no penalty to zero, and the best selected, as judged by the resulting model accuracy on unseen data. Coefficients can be shrunk either by reducing their number (lasso regression) or their overall magnitude (ridge regression).

2.3.2 Support vector machines

Support vector machines are a popular choice in text classification problems [37][33][30][41][40] [45] [36][51][52]. The choice of this model is generally rationalised in terms of its use and success in previous text classification applications. Lantz [53] also suggests that the recent implementations of good SVM algorithms in popular and well-supported libraries in different programming languages is also likely to be behind the increased usage of this type of model, as the mathematics behind these algorithms might otherwise be too complex for most researchers to implement.

The intuition, however, is relatively straightforward. The goal of a SVM is to find the hyperplane that divides data points in an n-dimensional space into two classes that are as homogenous as possible. The best hyperplane creates the largest possible separation between the classes [53]. However, because most data cannot be perfectly separated in this way, SVMs ‘allow’ some data points to be on the wrong side of the hyperplane; a hyperplane that almost separates the classes is created, to trade off some fit to the training data in order to generalise better on unseen data [24].

2.3.3 Random forest

Another popular option is random forest models [28][45][54]. These are an extension of decision trees, which follow a ‘divide and conquer’ approach to classification, splitting the dataset into progressively smaller branches until all cases in the terminal nodes have the same label. They are easy to interpret and quick to implement, but they cannot usually compete with other algorithms on predictive accuracy, and tend to overfit. [24]

Random forests are a way to improve these tree-based classifiers; multiple trees are estimated, and rather than simply choosing the strongest predictor as a basis for each split, each split can only consider a random subset of predictors. Introducing randomness in this way produces a set of trees that are less correlated with each other, and so when an average is taken across the trees it is less variabale, and thus more reliable [24]. The resulting performance of these models, and their implementation across a variety of packages, makes them a popular choice in machine learning applications in general [55].

2.3.4 Naive Bayes

Another model employed in the literature is Naive Bayes [46][45]. Again its use is typically justified by researchers on the basis that it is commonly, and often successfully, applied to text classification problems. It is also a simple and relatively quick model to implement, even on large datasets [53]. Naive Bayes is so called because it assumes that the predictors are independent of each other. This is a strong assumption and seems unlikely in a lot of cases, but the algorithm has been extensively used in text classification due to its relatively high success in this area. If what is seen as its key disadvantage – the independence assumption – is not a hinderance, then its speed, simplicity and ability to handle noisy and missing data make it a good choice, and this may explain its popularity [53].

The algorithm is based on calculating conditional probabilities, as per Bayes theorem. In this case, the task is to estimate the probability that a comment has a particular property (for example that it is abusive), conditional on what is known about the words and features it contains. The elements on the right hand side of Equation 2.4 can all be calculated from the labelled data; the probability of observing these words in an abusive comment, the probability of a abusive comment, and the probability of observing the words and features.

$$P(\text{abusive}|\text{features}) = \frac{P(\text{features}|\text{abusive})P(\text{abusive})}{P(\text{features})} \tag{2.4}$$

Because we assume that the probabilities are independent, they are additive, therefore the model takes the form in Equation 2.5. The probability of status L for comment C, given the evidence provided by feature set F is the sum of the probabilities of observing each feature given status L, multiplied by the probability that a comment takes status L. This is then multiplied by a scaling factor $\frac{1}{Z}$, which converts the likelihood values into probabilities.

$$P(C_L|F_1, ..., F_n) = \frac{1}{Z}p(C_L)\prod_{i=1}^{n} p(F_i|C_L) \tag{2.5}$$

2.3.5 Other models

Other studies have taken a deep learning approach, using convolutional neural networks [42][39], recurrent neural networks [31] or multilayer perceptrons [17]. This class of model connects inputs to outputs via a network of ‘hidden’ layers, which weight the inputs in order to produce the correct classification for the output [56]. They are somewhat hyped but powerful models whose popularity has increased in recent years [55]. They take the text classification problem in a slightly different and more cutting edge direction to the previously outlined models, but require a substantial amount of computational resources.

2.4 Evaluation and implications for the present analysis

As all of these studies are looking for different things in different domains, there is limited usefulness in drawing inferences about what would be the best approach in this study. However, a few useful generalisations can perhaps be made. On features, most of those who go beyond words to look at additional features get an improved model. Although it is possible to get a good model with just words, the most successful are often those using more complex or computationally intensive word embeddings rather than a simple bag of words. On algorithm choice, accuracy metrics (such as those outlined in the next chapter) typically range from somewhere in the 70s (on a 0-100 scale) for more intangible characteristics like sarcasm, up to the 80s and 90s for more clear cut ones such as abuse. Researchers have had success with a range of algorithms, especially support vector machines, naive Bayes and random forest, and neural networks achieve similar levels of success; in short, there does not seem to be a single best model for this type of problem.

Like Binns et al. [34], the analysis in this project is based on the Wikipedia detox datasets, although it advances beyond this by considering more than just the words themselves, but also other features. The above literature presents a number of options in terms of feature choices, which can be narrowed down by whether they are available in the chosen dataset and with the computing power available. The final selection of features, outlined in next chapter, tries to include any of the above that are possible. The literature has also provided a blueprint for a machine learning strategy; use several models, including a logistic model for a baseline, but also others that have previously been successful.

3 Data and methods

The methodology of this project was outlined in Section 1.2, which put forward a five step process for conducting a text classification problem. The previous chapter tackled step 1: understanding the problem. This chapter outlines how steps 2 to 4 were undertaken, from data and model selection to implementation and evaluation, with the following chapter presenting the outcomes of these steps. The chapter also presents and discusses the ethical issues in the project.

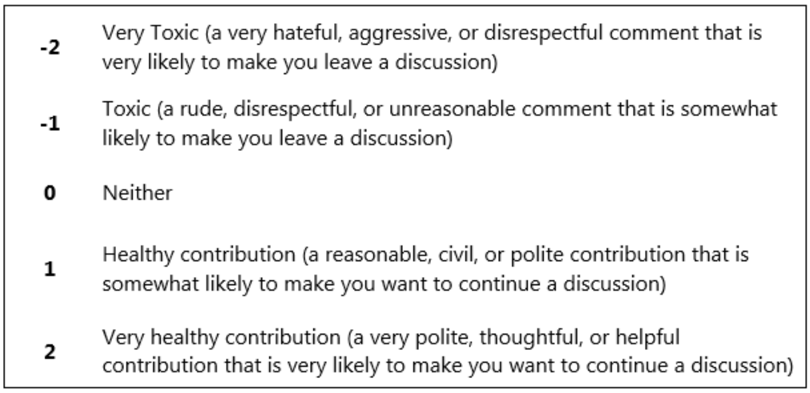
All data manipulation and machine learning was carried out in R, using RStudio. As R is a popular language in which to perform these tasks, there is a great deal of online support via courses, tutorials, package documentation and community support. Several of these informed the analysis here; these were listed in the Attestation section of this document.

A key limitation in using R is memory; as R stores data in RAM, there is a limit to the size of file it can work with or model it can build. However, this is less likely to be an issue with annotated corpora, which are unlikely to be very large due to the labour intensity of producing them, and indeed in this case the working object was only 75MB. Perhaps the main disadvantage is R's slower speed, which meant that subsamples of the data had to be used when training models.

3.1 Obtaining a suitable dataset

The first stage in the process was to obtain some suitable data. Given the limited timeframe of this project, rather than spend time creating a new purpose-built corpus from scratch, an existing relevant dataset that was already labelled and freely available was chosen to train the classifier.

The Wikipedia Detox project and the corpus created as a result were introduced in Chapter 1. The data produced during the Detox project is publicly available, in three separate datasets annotated for for personal attacks, aggression and toxicity. This analysis makes use of the third of these; the toxicity dataset, which contains around 160,000 annotated English language comments from Wikipedia. The comments have been taken from discussion pages pertaining to articles and users on the site. Annotations were carried out by workers on the online crowdsourcing platform Crowdflower, who were asked to score comments on the scale presented in Figure 3.1.



-2	Very Toxic (a very hateful, aggressive, or disrespectful comment that is very likely to make you leave a discussion)
-1	Toxic (a rude, disrespectful, or unreasonable comment that is somewhat likely to make you leave a discussion)
0	Neither
1	Healthy contribution (a reasonable, civil, or polite contribution that is somewhat likely to make you want to continue a discussion)
2	Very healthy contribution (a very polite, thoughtful, or helpful contribution that is very likely to make you want to continue a discussion)

Figure 3.1: Scale on which annotators were asked to judge examples as toxic

A number of precautions were taken by the creators of the data to maximise the quality of the annotations [17]. Each comment was annotated by at least 10 different annotators. Each annotator had to first of all correctly annotate 7 out of 10 test examples, with further test questions randomly interspersed throughout to maintain quality. Inter-annotator agreement was measured using Krippendorf's alpha and was found to be in line with similar crowdsourced datasets. Despite these protective measures, the data remains vulnerable to questions over its validity due to its non-expert evaluation. However, it offers a rare source of labelled data on which to perform supervised learning in this area.

The models were trained and evaluated initially on the data taken from the *user* page discussions. Data from the *article* page discussions was subsequently used to evaluate how well the classifier might generalise to another domain.

3.2 Tidying and feature extraction

The data was cleaned and a set of features extracted, representing the words used in the comments as well as 'metadata' such as the presence of stylistic features and sentiment words. This section outlines this process, which is also summarised in Figure 3.2.

The dataset is provided as two tab-separated files, one each for the comments and the annotations, which can be matched by a comment ID. Two pieces of information supplied about the comment were pertinent. The first was whether it came from a user or article talk page; the former was used for model training (and initial evaluation), with the latter used in an additional evaluation stage. The second useful piece of information was whether the comment was made by an editor that was logged in, and therefore whether the comment was anonymous or not. Previous research has suggested that the anonymity afforded by the internet may exacerbate abusive behaviour [57][58][59][60][61]. Therefore this information was stored as a feature of the comment for use in the machine learning model.

The comments data had already been partially cleaned, with Wikipedia markup and HTML stripped out, so there were only minor data preparation tasks to perform. Tokens indicating a tab or new line were removed, as this was unlikely to offer useful information, and any URLs or email addresses in the comments were also removed.

At this stage, prior to tokenization, numeric variables were created representing the length of the post, and the average length of each word used in the post. A number of comment features were also extracted using regular expressions. A binary indicator was constructed to denote the presence or absence of each of the following features in a comment:

- any words in all capitals
- any repeated punctuation
- any words with repeated letters (e.g. sooo, zzzz)

Tokenisation was then carried out using the R `tidytext` package. There are a number of text mining utility packages available in R, which are useful because they perform the heavy lifting of the tokenisation; a body of text can be quickly and easily split into the tokens of interest. The `tidytext` package is designed to create data frames that conform to so-called 'tidy' data principles, which means the data frames output after tokenisation can be easily used with other popular and user-friendly packages such as `dplyr` for manipulating data and `ggplot` for creating plots. The `tidytext` package offers a number of options for defining the token of interest, but in this case simple unigrams were used; words, as delimited by spaces. The package converts tokens to lower case and strips out punctuation between, but not within, words. Any token containing only numbers was removed, but words containing both numbers and letters were left in, as they may contain some information, for example if users are swapping letters for numbers to avoid detection of abusive language. Stop words (the default list supplied by the `tidytext` package) were removed. Other pre-processing steps that are sometimes taken at this stage include stemming and/or lemmatization of words, replacing contractions with full versions, and correcting spelling or grammatical errors. However, these steps were not undertaken here, because the presence of a shortened, colloquial or misspelled version of a word might be relevant information about the comment.

After tokenization, a vocabulary was constructed of any word appearing at least 50 times, and each comment was turned into a vector of the TFIDF of each word (as outlined in the previous chapter, this is the frequency with which the word appears in a comment, offset by the frequency with which it appears across all comments). This yielded a large set of features (3,276 unique words), resulting in the need for feature selection or dimensionality reduction. Principal components analysis was carried out as a quicker and less computationally intensive method of dimensionality reduction than the word embedding methods employed elsewhere in the literature. This is method that creates a new, uncorrelated set of predictors from a large set of original (and potentially highly correlated) predictors, by looking for the combinations that together explain the most variance [62]. Carrying out this procedure reduced the size of the feature space from 3,276 words to 50 components. All components were retained at the this stage, as feature selection took place within the modelling process, through methods such as regularisation of logistic models.

Finally, some sentiment features of each comment were extracted. The mean sentiment score for each comment was calculated, based on the sentiment scores in the AFINN sentiment lexicon [63], which assigns a score of between -5 and 5 to around 2,500 words. The presence of eight other sentiments, as found in the NRC sentiment lexicon [64], was also counted, and comments were categorised on the basis of whether they contained at least three words pertaining to these sentiments. The extent to which the presence of these words distinguished toxic from non-toxic posts was ascertained (see Figure 4.4 in the next chapter), and the three most distinguishing words were chosen for inclusion in the model (these transpired to be trust, disgust and anticipation).

The transformation of a comment into its corresponding feature set is summarised in the diagram shown in Figure 3.2. The post in that example had a length of 51 words, and a mean word length of 3.7 letters. It was made by a user that was logged in, so was is not anonymous. It did not contain any repeated letters or punctuation, and it had just one all-caps word, which seems to be an abbreviation. After tokenisation and the removal of stop words, there were 14 unique words in the comment. Three of these had sentiment categories and scores attached to them; for example the word 'luck' has three associated sentiments in the NRC lexicon (anticipation, joy, and surprise), and is given a score of +3 in the AFINN lexicon. The sentiment scores averaged out to a score of 2 for the comment as a whole. The words in the comment do not appear in the feature set, but rather the scores for the 50 components created as a result of principal components analysis.

The text, sentiment and other features were brought together for each comment, which was then united with its toxicity label, derived from the annotations dataset. In the annotations dataset, each comment received at least 10 annotations, coded as 1 for toxic (i.e. the annotator had assigned it a score of less than zero on the scale presented in Figure 3.1), and zero for non-toxic. This was averaged across all annotators to produce a mean score, and then classified as toxic overall if the mean exceeded 0.5, and therefore a majority of annotators agreed it was toxic. So for example, a comment with 10 annotations needed 6 of the annotators to classify it as toxic, producing a mean rating of 0.6, in order to be considered a toxic comment. Variations on this threshold were considered, and the results of this are presented and discussed in Section 4.5.

The resulting dataset was used to train and evaluate machine learning models. The process of text processing, and the machine learning exercise the resulting data was used in, is summarised in Figure 3.3.

3.3 Machine learning

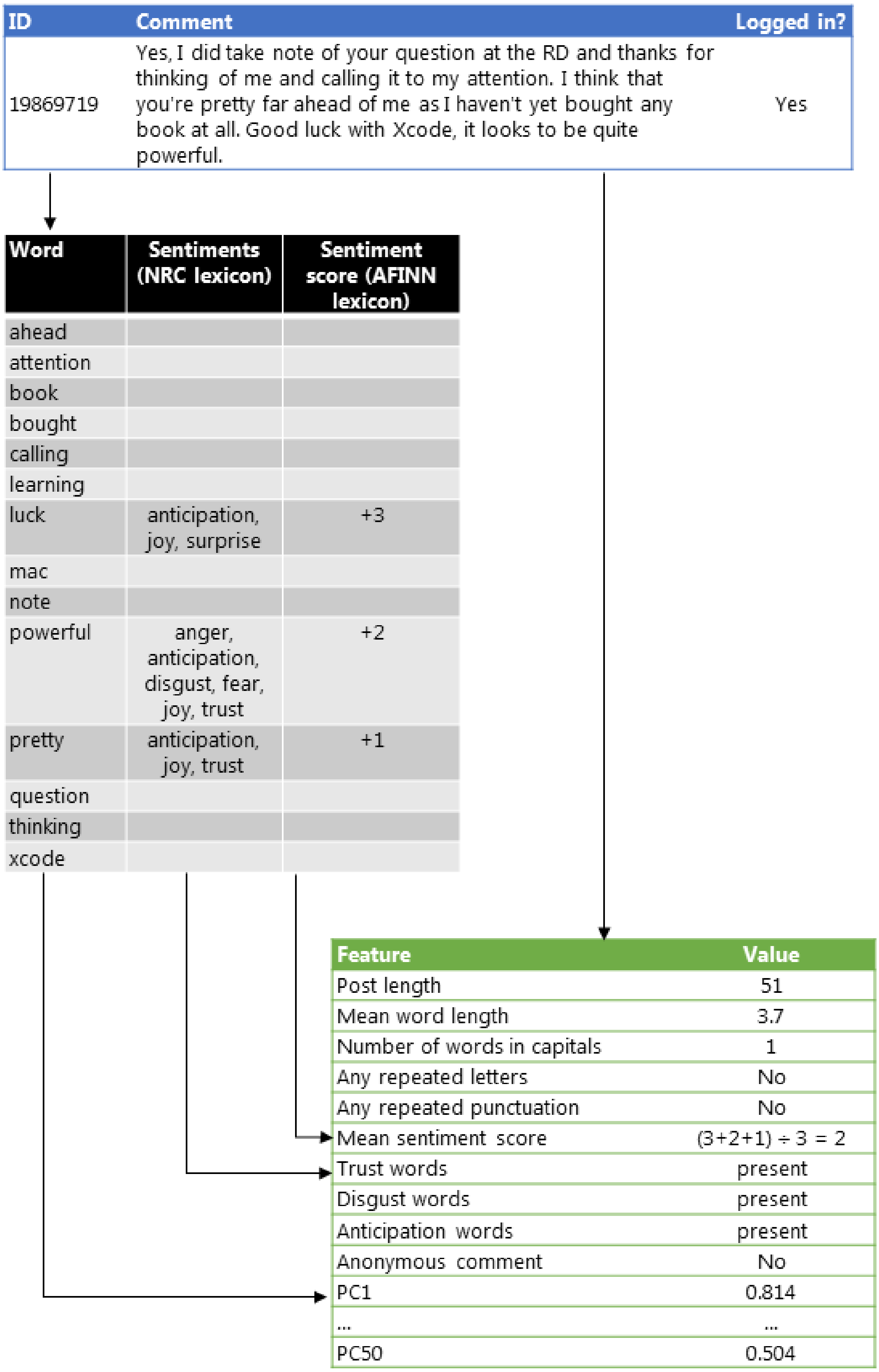
Having extracted the features from the text, the next step was to train machine learning models. Four machine learning algorithms were selected, based on their widespread use in previous similar studies; logistic regression, random forest, naive Bayes and support vector machines (SVM).

3.3.1 Model training and validation

Prior to training models, a random sample of 20,000 comments (out of the 95,000 in the dataset) was taken to enable models to be built, as at larger sample sizes than this, models could not be trained without computer failure, or took excessively long. Predictors were scaled and centred (to take a zero mean and standard deviation of one), as this has been shown to improve the numerical stability of calculations [62].

The data was split into training and test sets. The aim of the modelling process is not to perfectly fit a model to the data on which it was trained, but rather to perform well at categorisation when presented with unseen data. Therefore a training set was used to train the model, with a test set held back to see how well the model performed on unseen data. The data was split 70:30 into training and testing sets, stratified by the outcome variable in order to ensure a similar representation of each outcome class in the two sets. To further improve this training and validation process, 10-fold cross-validation was used. The training data was further split into 10 folds, trained on 9 and tested on the 10th, and then the process was repeated untill all folds had been used as a test fold. The final parameter estimates were the average of the estimates made in all 10 steps of this process, and the performance of the model was finally tested on the 30% of the data that had been held out for this purpose.

In machine learning there is a trade-off between bias and variance. If a model perfectly fits the training data it will have low bias but high variance, and it is likely to be modelling random noise, so it will not generalise to new data. Therefore some bias can be deliberately introduced to

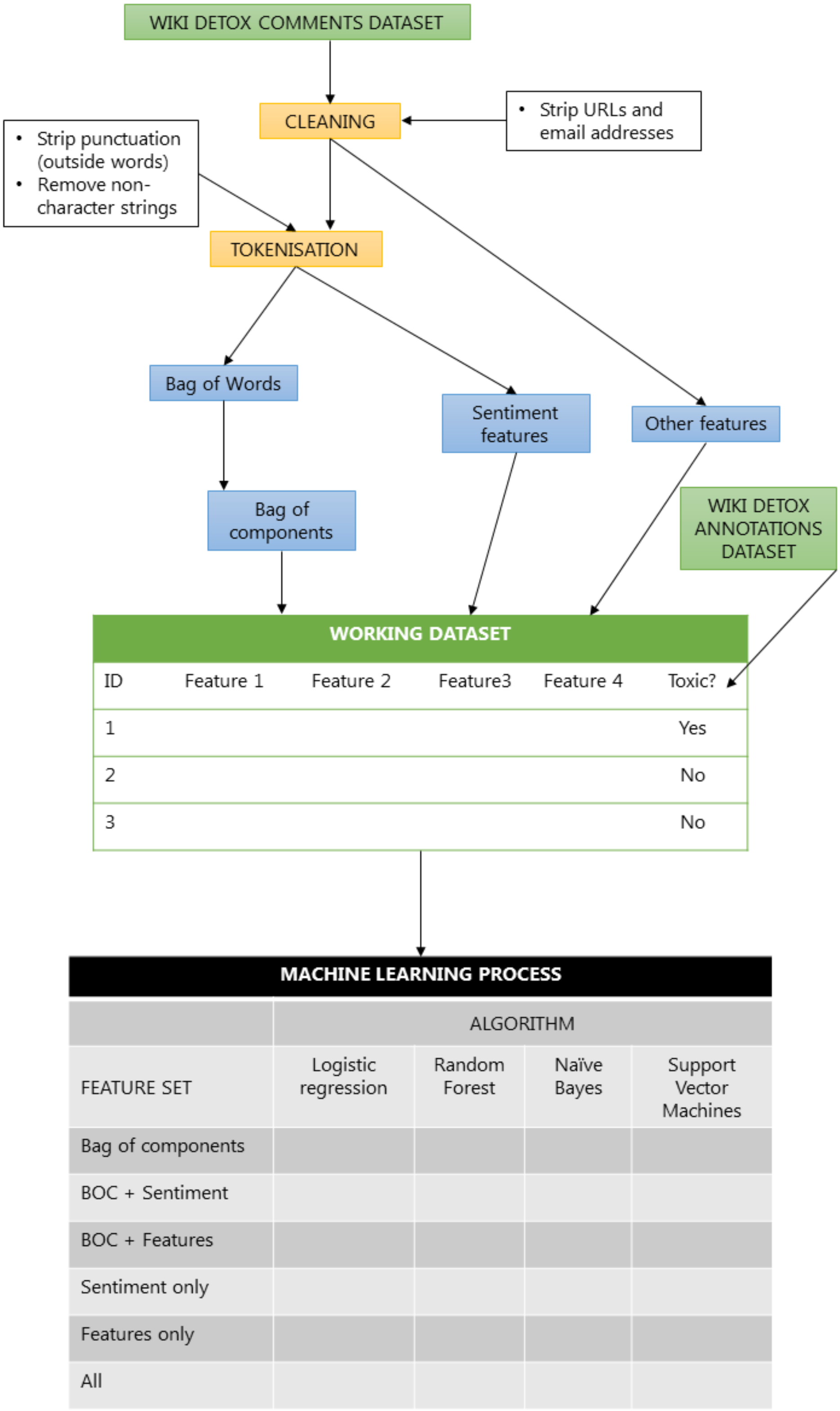


ID	Comment	Logged in?
19869719	Yes, I did take note of your question at the RD and thanks for thinking of me and calling it to my attention. I think that you're pretty far ahead of me as I haven't yet bought any book at all. Good luck with Xcode, it looks to be quite powerful.	Yes

Word	Sentiments (NRC lexicon)	Sentiment score (AFINN lexicon)
ahead		
attention		
book		
bought		
calling		
learning		
luck	anticipation, joy, surprise	+3
mac		
note		
powerful	anger, anticipation, disgust, fear, joy, trust	+2
pretty	anticipation, joy, trust	+1
question		
thinking		
xcode		

Feature	Value
Post length	51
Mean word length	3.7
Number of words in capitals	1
Any repeated letters	No
Any repeated punctuation	No
Mean sentiment score	(3+2+1) ÷ 3 = 2
Trust words	present
Disgust words	present
Anticipation words	present
Anonymous comment	No
PC1	0.814
...	...
PC50	0.504

Figure 3.2: Example transformation of a comment into a feature set


WIKI DETOX COMMENTS DATASET
CLEANING
Strip URLs and email addresses
Strip punctuation (outside words)
Remove non-character strings
TOKENISATION
Bag of Words
Sentiment features
Other features
Bag of components
WIKI DETOX ANNOTATIONS DATASET

WORKING DATASET					
ID	Feature 1	Feature 2	Feature3	Feature 4	Toxic?
1					Yes
2					No
3					No

MACHINE LEARNING PROCESS				
	ALGORITHM			
FEATURE SET	Logistic regression	Random Forest	Naïve Bayes	Support Vector Machines
Bag of components				
BOC + Sentiment				
BOC + Features				
Sentiment only				
Features only				
All				

Figure 3.3: Overview of the data cleaning and modelling process

prevent overfitting to the training data. However we do not want the model to be so biased that it is not sufficiently sensitive to the underlying relationships, and therefore makes poor predictions.

In order to find this compromise, machine learning models have settings called hyperparameters that can be 'tuned' to find the optimal trade off between bias and variance. Hyperparameters cannot be directly estimated; the tuning process involves moving through a search space. In this case, a grid search was employed, estimating each model at a range of hyperparameter settings and choosing the most optimal in terms of the highest kappa value produced. In the logistic model, the aim is to find the optimal level of regularisation that penalises unimportant coefficients to avoid overfitting, but does not penalise so much that the model cannot explain anything. In random forest models, the aim is to allow enough random variables at split points so that the trees can be less correlated, but not so much that the trees lose their ability to classify. In SVMs, the relevant tuning parameter is the extent to which some cases are allowed to be on the 'wrong' side of the hyperplane. Naive Bayes is slightly different, as unlike the other algorithms where the aim is to balance model fit with predictive ability by limiting the number of predictors or the effect they have, it uses all the available evidence.

3.3.2 Implementation

Each of the four algorithms outlined above were implemented for six different feature sets; each type of feature separately (bag of components, sentiment features, and other features), followed by bag of components with the sentiment and other features, separately and then together.

All model training and evaluation was carried out in R using the `caret` package, which contains functions for performing the key stages of the machine learning pipeline. The advantage of using this package is that it can perform key tasks, such as splitting the dataset, centering and scaling the predictors, and resampling, in an automated way that is standardised across the different algorithms and feature sets. The same cross-validation folds were used across all models, so that the differences between them were less likely to be a sampling artefact. The package also has the option to perform upsampling, which tries to correct for a class imbalance by imputing additional data points in the smaller class [62]. This was beneficial in this case given the relatively small prevalence of the toxic class.

The `caret` provides a wrapper for the packages that implement the algorithms themselves, outputting results in a standardised way for ease of comparison. The penalised logistic regression model was implemented using the `glmnet` R package, which allows tuning of the the balance between lasso and ridge, and the size of the penalty (i.e. from zero to complete shrinkage). The random forest model was implemented using the `ranger` package in R, which has been optimised for high dimensional data and is therefore well suited to the task at hand. Splits are chosen on the basis of maximising the purity of the resulting nodes, as measured by the Gini index, and the number of random variables at each split can be tuned. Naïve Byes was implemented with the `naivebayes` package in R. The package allows for a form of tuning in the form of adding Laplace smoothing; adding a very small value to each probability to allow for combinations of feature and class that do not occur in the data. Finally, SVM was implemented using the `kernlab` package in R, which has the option to tune the cost (the extent to which cases are

Table 3.1: A confusion matrix with four possible outcomes of the classification process

		Actual	
		Not toxic	Toxic
Predicted	Not toxic	True negative (TN)	False negative (FN)
	Toxic	False positive (FP)	True positive (TP)

allowed to be on the wrong side of the hyperplane).

3.3.3 Evaluation

The resulting model was used to predict outcomes on the testing dataset, and from this, measures of the success of the model were calculated. A confusion matrix compares the outcomes that a model predicts for the test set cases against their actual outcomes. In the case of the binary outcome here, this resulted in a 2x2 table (Figure 3.1), with four possible outcomes:

- true positives (TP): toxic comments correctly predicted to be toxic
- false positives (FP): not toxic comments predicted to be toxic
- true negatives (TN): not toxic comments predicted to be not toxic
- false negatives (FN): toxic comments predicted to be not toxic

From these four numbers, a number of measures of success can be constructed. The model **Accuracy** is the percentage of correctly classified test set cases: $\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}$. However, in situations such as this, where the toxic category accounts for less than 15% of the cases, a reasonable seeming accuracy of over 85% could be achieved simply by guessing non-toxic every time. What is interesting, therefore, is the extent to which the model takes the information it is given, and uses this to improve on what it could do if it had not been given the information. An adjusted measure of accuracy called the **Kappa statistic**, in evaluating how successful a model is, takes into account the probability that a correct prediction was reached by random guess. It compares the extent to which the model's actual predictions compare with the true values against the extent to which you would expect them to do so if they were chosen at random, and takes a value between zero (no agreement between the predictions and the true values) and 1 (perfect agreement). There is no fixed threshold for what constitutes an acceptable or good kappa value, but the conventional cut-off points [53] are:

- Poor agreement = less than 0.2
- Fair agreement = 0.2 to 0.4
- Moderate agreement = 0.4 to 0.6
- Good agreement = 0.6 to 0.8
- Very good agreement = 0.8 to 1

Beyond the overall accuracy of a model, it may be interesting to know if the model is overly cautious or overly zealous in identifying toxic comments. **Sensitivity** (also known as the true positive rate) is the proportion of toxic test cases that are classified as toxic: $\frac{\text{TP}}{\text{TP}+\text{FN}}$. The inverse of this is the false negative rate, the proportion of toxic cases classified as not toxic. **Specificity** (also known as true negative rate) is the proportion of non-toxic test cases that are classified as not toxic: $\frac{\text{TN}}{\text{TN}+\text{FP}}$. The inverse of this is the false positive rate, the proportion of not toxic cases classified as toxic. The relative importance of these is context dependent. In this case, if flagging a post as toxic creates a nontrivial amount of effort (the post has to be followed up by a person), then a high false positive rate is not very efficient. However, a high false negative rate means that potentially toxic posts may be missed.

The extent to which a model gives false positives or negatives depends on the threshold of predicted probability at which a test case is assigned an outcome. If this threshold is set such that a predicted probability must be 100% before a case is assigned a positive outcome, then there will be very few false positives, but probably many false negatives, while if this threshold is very low there will be many more false positives. This trade-off can be plotted in the form of a ROC curve, which plots the true positive rate against the false positive rate at every threshold level. If the resulting 'curve' takes the form of a diagonal 45 degree line then there is no difference between the model's ability to predict the outcome and doing so by random. The further away the curve is from the diagonal, the better the model, and this is why the area under the curve (AUC) is also used as a measure of model quality.

Related to sensitivity and specificity are the concepts of **precision** and **recall**. The former can be thought of as a measure of trustworthiness; it tells us, when the model predicts a positive result, how likely is it to be correct: $\frac{\text{TP}}{\text{TP+FP}}$. The latter (the same as sensitivity) tells us to what extent the model is picking up positive results: $\frac{\text{TP}}{\text{TP+FN}}$. As with sensitivity and specificity, there is a trade-off: between reliably identifying positives but only identifying a small proportion of them, and identifying a high proportion of positives but where many positive predictions turn out to be false. A good model should do both well, and therefore these two measures are averaged to produce the F1 score, with a high score indicating a good model.

All of the above metrics were calculated for all models, in order to compare their ability to predict whether a post is toxic or not. In order to ensure comparability across models, the same training dataset and the same folds for cross-validation were used when training the models.

3.4 Ethics

The project makes use of data that has been contributed by many thousands of individuals through their participation in editor discussions on Wikipedia. As the project uses data from a third party platform (Wikipedia), contributed by humans (Wikipedia editors), the associated legal and ethical issues need to be addressed.

Some analyses of user-contributed online content can violate the terms of service of the platforms from which the data has been taken; this can especially be the case with data that has been 'scraped' from sources such as Twitter or discussion boards. However, this is a dataset that has been collected and distributed for analysis by the platform itself, so there is no issue in this respect. The key ethical issue here is around how 'public' this data is, and what constitutes legitimate reuse of data. The Wikipedia users have released their comments, and any information they supply about themselves in their Wikipedia profile, into the public domain. However it has been supplied for a specific purpose and audience, not intended for further scrutiny, and they may have been more inhibited in their expression had they known their words would be analysed [65]. It is not possible to ask the authors of this data for permission to analyse it, but it is legal under the new General Data Protection Regulations (GDPR) to repurpose this type of data for scientific research, provided the published analysis does not breach confidentiality or do harm to those who have supplied it.

The key risk of harm when repurposing data is unwanted disclosure that results in a negative outcome for the person, such as embarrassment, attacks, or danger. Although the dataset does not link comments to their authors, the data is not completely anonymised; for example, usernames mentioned within a comment have not been redacted. However only a handful of comments are reproduced in full here, and care has been taken not to directly identify in this analysis any author of toxic comments, in case they experienced backlash from this. As this is the only presentation of data at an individual level, there is no possibility of cross-identification, where a number of pieces of information about an individual can be linked together to infer another, undisclosed piece of information. The data used here is broken down in to constituent features, and any link between an author and their words is lost very quickly. Therefore it is difficult to argue that any contributor has come to any harm as a result of this analysis.

A further ethical consideration relates to the way this work might be deployed in real moderation situations. A principle of the GDPR is that no decision affecting a person should be based on an algorithm alone. Therefore the ethical usage of such work relies on it being deployed as part of a system in which no sanction is administered without human intervention. For example an alert could be triggered when the predicted probability of toxicity is above a certain level, for passing to the next stage of moderation. As Binns et al. [34] observe, what constitutes toxic is a subjective matter governed by the norms of a platform, and continuously contested among its users and moderators. Therefore it is important that the role of the classifier is an assistant, and it is not seen as some neutral arbiter, or a way to avoid difficult conversations or decisions, or sidestep the problem of defining what is acceptable or unacceptable behaviour.

4 Results

4.1 Description of the data

The Wikipedia Detox toxicity dataset contains 159,686 comments, of which 64,700 are taken from article talk pages and 94,986 from user talk pages. This analysis used the latter, more toxic, set for training the machine learning models (13% of the user talk comments are toxic compared with 4.6% of the article talk pages). Comments from the article talk pages were used afterwards as a way to test how well the classifiers might generalise to a different dataset.

The most common 20 words in the toxic and non-toxic posts were counted, after removing common 'stop words' and the most offensive swear words (Figure 4.1).[1] The vocabulary in the toxic posts was still fairly offensive, while the words in the non-toxic posts were much more neutral. The top 20 words in the toxic comments featured both 'wikipedia' and 'wiki', suggesting that the decision not to stem words was appropriate, as differences in the level of formality of writing could be indicative of differences between toxic and non-toxic posts.

To better distinguish between between the language used in the two types of comment, it is beneficial to look at the relative frequency of words, to see which words are relatively frequent in one type of comment but not the other. In Figure 4.2, words near the diagonal line appear with a similar relative frequency in both types, while words below the line are relatively more frequent in toxic posts. Each dot represents a word, and the words on the plot are examples of words that appear at that position. Although the words were concentrated around the diagonal, the edges of the cloud suggest a vocabulary that might distinguish the two types of post. For example, a post containing the word 'tutorial' was much more likely to be non-toxic, while a post containing the word 'poop' was much more likely to be toxic. The correlation between the word frequencies was found to be 0.23, which is fairly low, lending support to the idea of distinct vocabularies.

The average sentiment score in toxic posts was lower than non-toxic posts (Figure 4.3), suggesting that their overall tone is more negative. The average non-toxic post was found to be essentially neutral in tone, with an average score of 0.37, while toxic posts were on average negative, with a score of -1.62. Picking out the words relating to the 8 emotions in the NRC lexicon and looking at the prevalence of (three or more of) each in the different types of post, non-toxic posts were much more likely to contain words pertaining to trust, while toxic posts were more likely to contain words pertaining to disgust (Figure 4.4). For some emotions, such as fear or sadness, there was little difference between the two types of post.

Looking at the other features considered in this analysis, toxic posts were found to be more likely to be anonymous, and more likely to contain features such as words in capital letters, words with repeated letters, and repeated punctuation (Figure 4.5). The difference in the presence of all-capital words was small; in fact the key difference turned out to be in the *number* of all-capital words, with a median of 20 in toxic posts and 3 in non-toxic. The mean word length was about the same in toxic and non-toxic posts (4.3 and 4.6 respectively) but the mean post length was longer for non-toxic posts, at 67.5 words compared with 50.7 for toxic posts.

[1] The offensive words were removed for the purposes of the descriptive analysis only, and not for the final modelling process.

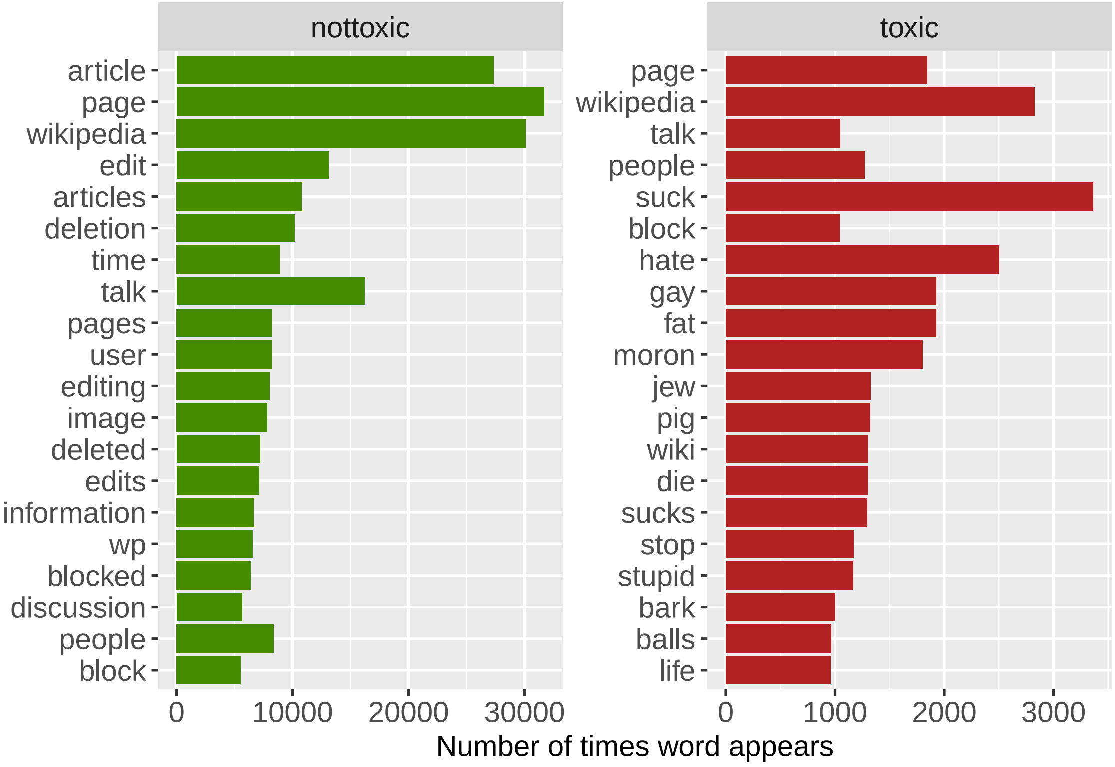


Figure 4.1: Top 20 words appearing in each type of post

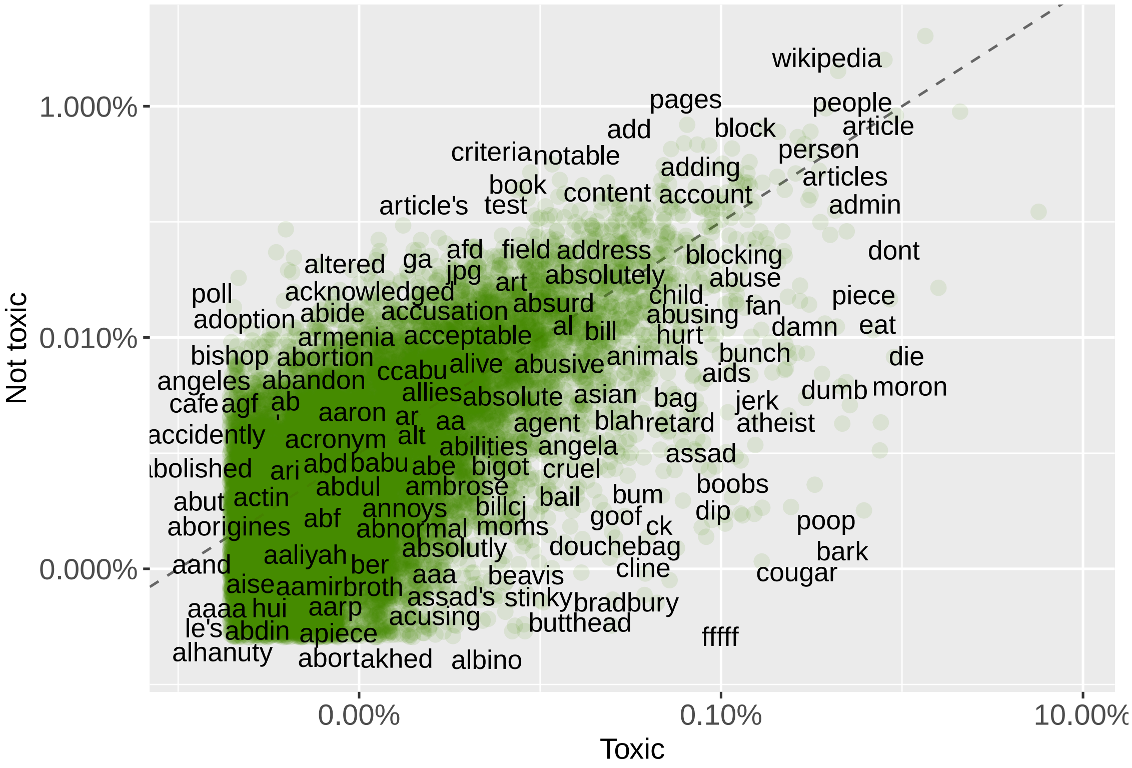


Figure 4.2: Relative frequencies of words in toxic and non-toxic posts

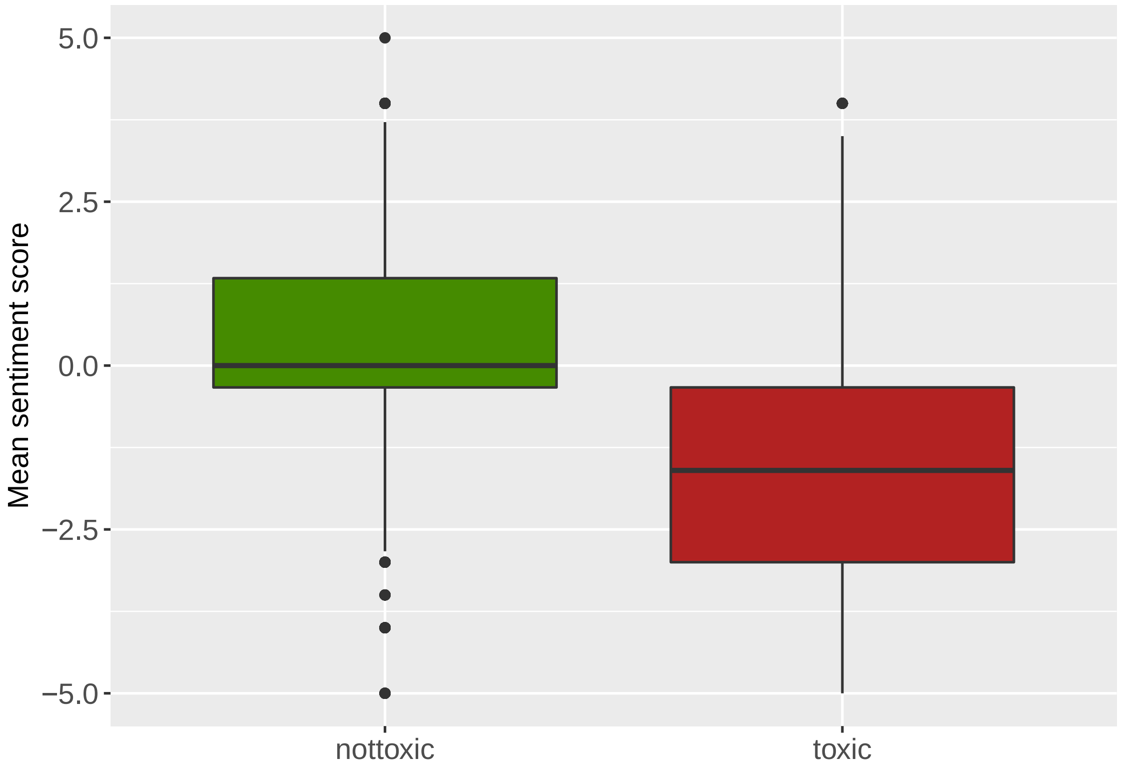


Figure 4.3: Boxplots showing the mean and distribution of sentiment scores in toxic and non-toxic posts

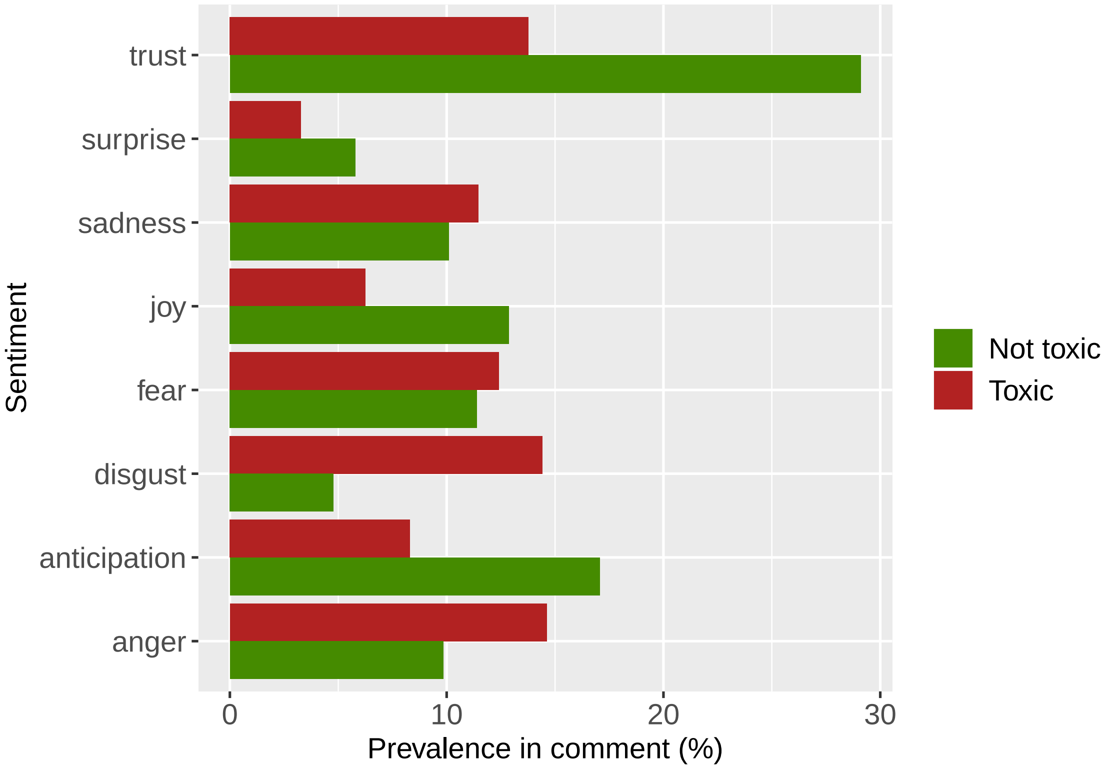


Figure 4.4: Prevalence of 8 types of sentiment in toxic and non-toxic posts

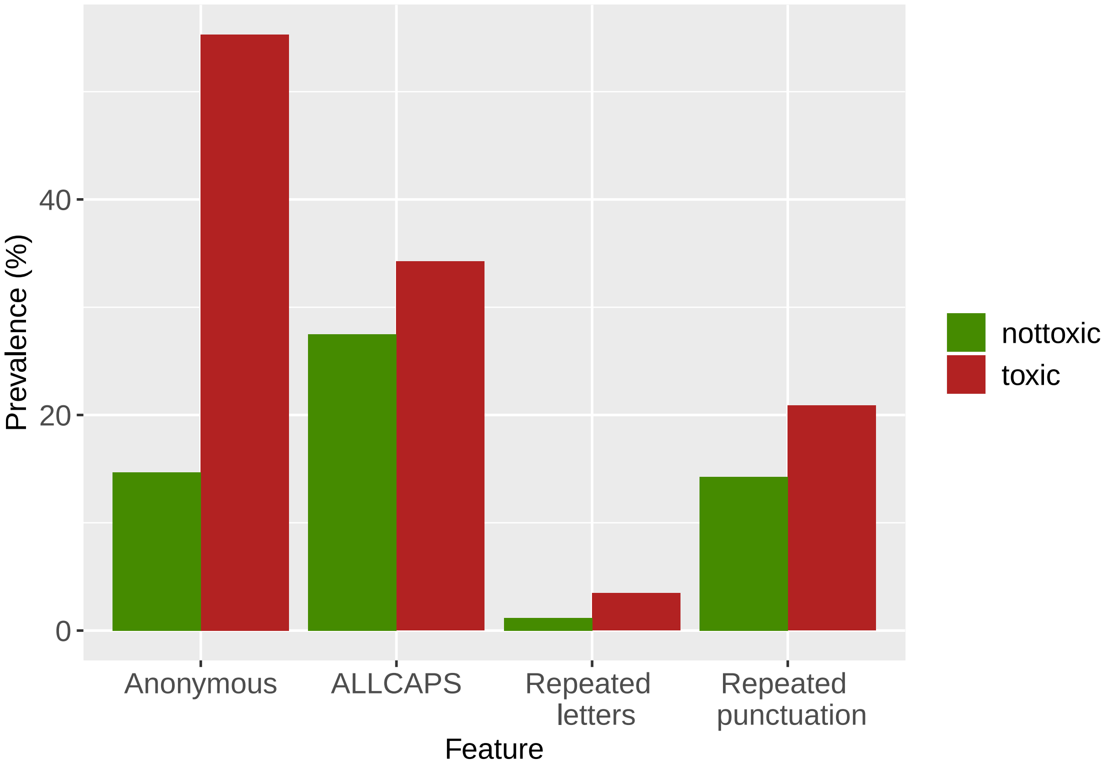


Figure 4.5: Prevalence of features in toxic and non-toxic posts

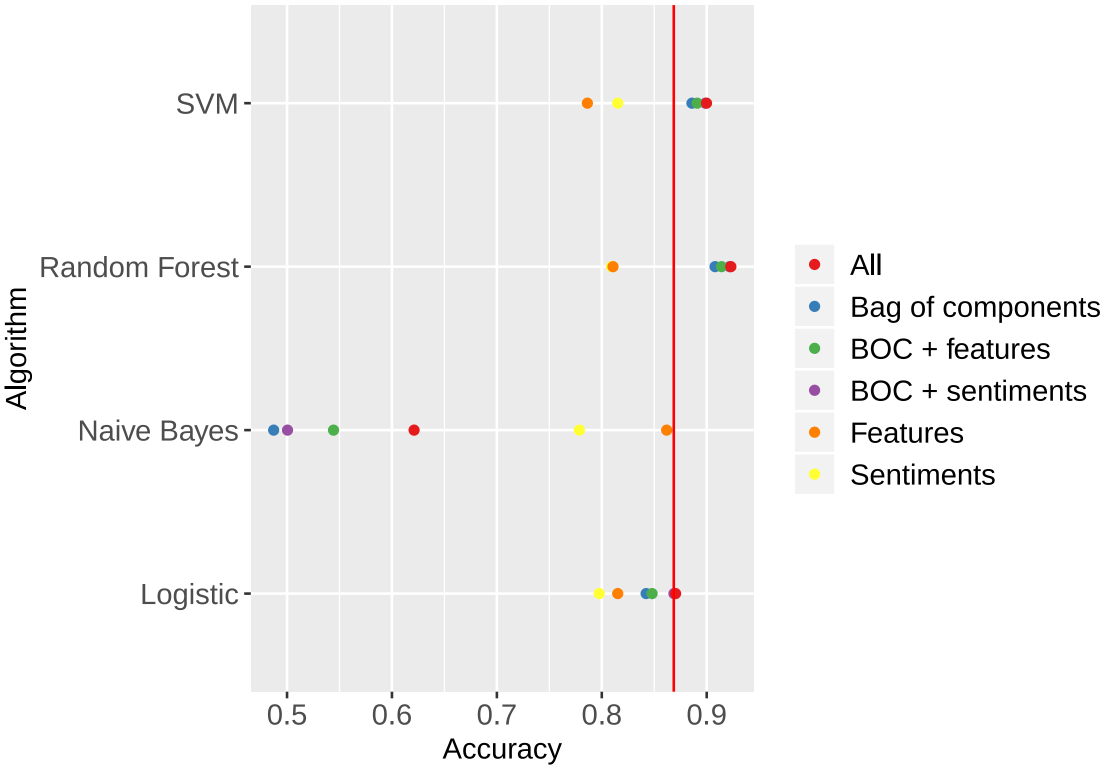


Figure 4.6: Accuracy obtained by each algorithm for each feature set

4.2 Modelling results

Classifiers were trained using four different algorithms and six different feature combinations; thus, 24 models in total. Figure 4.6 shows the classification accuracy obtained by each algorithm, for each feature set. The red line indicates what the accuracy would be without the additional information provided by the feature set (i.e. by guessing the most prevalent category), and it is difficult to improve on this as the dataset is unbalanced. Most of the random forest and SVM models are better, as is the full model for logistic regression. For the random forest, SVM and logistic regression algorithms, a bag of components alone performs almost as well as a model with additional features, but the best model is the full model. In contrast, none of the naive Bayes models perform better than the no information level, and bag of components does not appear to be a particularly useful feature set when using this algorithm.

As the version of the model containing all features was found to be the best for most algorithms, the remainder of the chapter will present results pertaining to this full model only. A table showing all of the evaluation metrics outlined in the previous chapter for all models is shown in Table 6.1 in the Appendix. The values presented here are also for what was found to be the optimal hyperparameter settings, as found by the grid search detailed in the previous chapter; details of different results obtained under different hyperparameter settings are given in Figure 6.1 to Figure 6.4 in the Appendix.

As 10-fold cross-validation was undertaken in the model training process, an accuracy value was

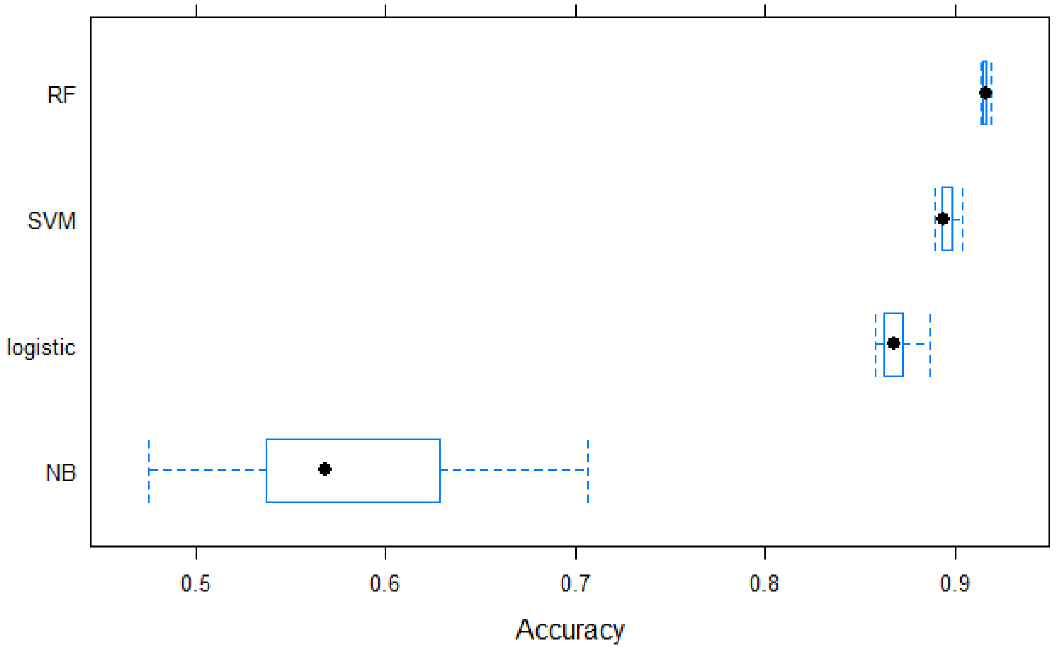


Figure 4.7: Boxplot of classification accuracy obtained in each resample (all features)

obtained at each resample, and the distribution of the obtained values can be seen in Figure 4.7. This allows us to compare not only the accuracy of each model, but also their consistency and reliability. The random forest model is the best performer, with a higher accuracy in every resample than any other model. It is also the most consistent of the algorithms, resulting in a similar classification accuracy every time. Accuracy values are distributed across a relatively small range for the SVM and logistic classifiers, but the range of values obtained by Naive Bayes is wide, suggesting that any good accuracy obtained from this algorithm should be viewed with caution, as it may not be representative of its performance more broadly.

The performance of the models relative to simply taking a random guess can be seen in their respective kappa values, which indicate moderate to good agreement in most cases. Figure 4.8 shows the kappa values obtained in the full models across each of the 10 resamples. Most values are at the higher end of the moderate agreement category. On this metric, random forest is the better performer, although there is some overlap with the logistic model.

On accuracy and kappa alone, random forest appears to perform the best. However this is not a full assessment of a classifier's usefulness. A classifier guessing non-toxic would be over 85% accurate, and have a high specificity (all the non-toxic cases would be correctly be classified as non-toxic), but have a sensitivity or recall of zero (it would not identify any of the toxic posts). A useful classifier will identify some toxic posts, even if it returns some false positives in doing so. Looking at the area under the ROC curve as a metric to understand this trade-off (Figure 4.9), naïve Bayes performs relatively poorly here. The logistic and random forest models have a

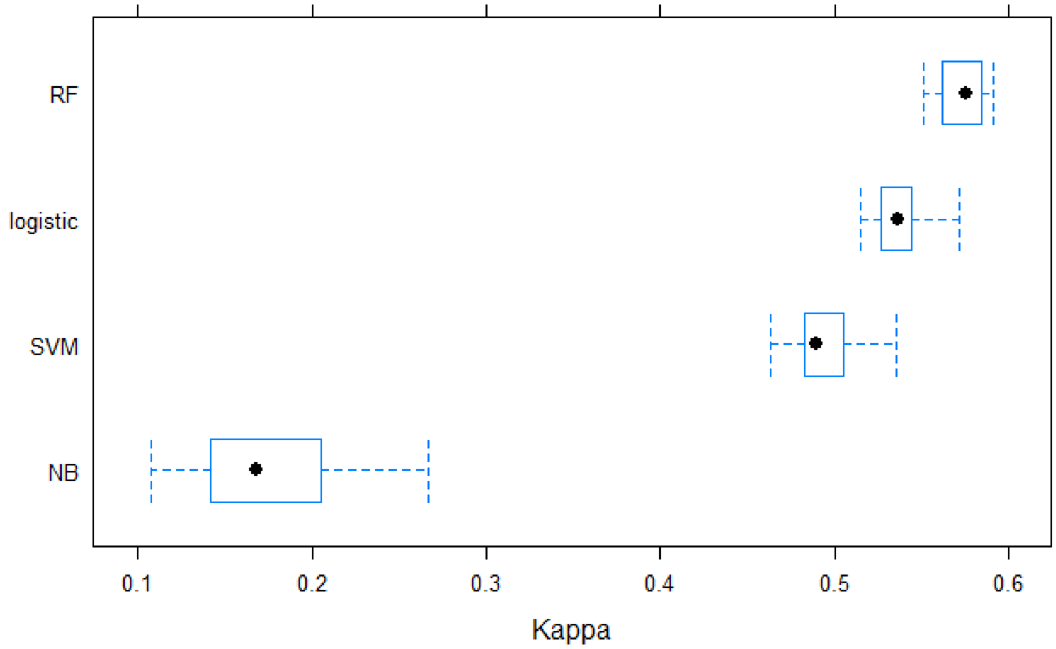


Figure 4.8: Boxplot of kappa values obtained in each resample (all features)

similar area under the curve, with SVM sitting somewhere between these and the naive Bayes model.

As outlined in the previous chapter, good models balance precision and recall. Figure 4.10 shows the precision, recall and F1 scores for the full feature set for each algorithm. The three best performing algorithms achieve very similar F1 scores, but in different ways; logistic regression has higher precision but lower recall, but they are equal in the SVM model.

The different algorithms resulted in similar performance metrics, but looking at the confusion matrices they produce (Table 4.1) shows that this conceals some notable differences between what each model is good at. Random forest achieves a good F1 score because it is the best at not misidentifying a non-toxic post as toxic. However, it is not as good as the logistic model at identifying toxic posts as toxic. If there is a cost to flagging a post as toxic – for example if this is some trigger for human intervention – it would potentially be quite wasteful to use a classifier like the logistic regression model, which produces false positive 10% of the time, compared to random forest, which produces a false positive just 2% of the time. However it depends how this cost compares to the cost of missing a toxic post, if every exposure to toxic posts is resulting in a 'cost' in the form of fewer site users. The random forest and SVM classifiers both miss around 2 in 5 toxic comments, while the logistic classifier misses 1 in 5.

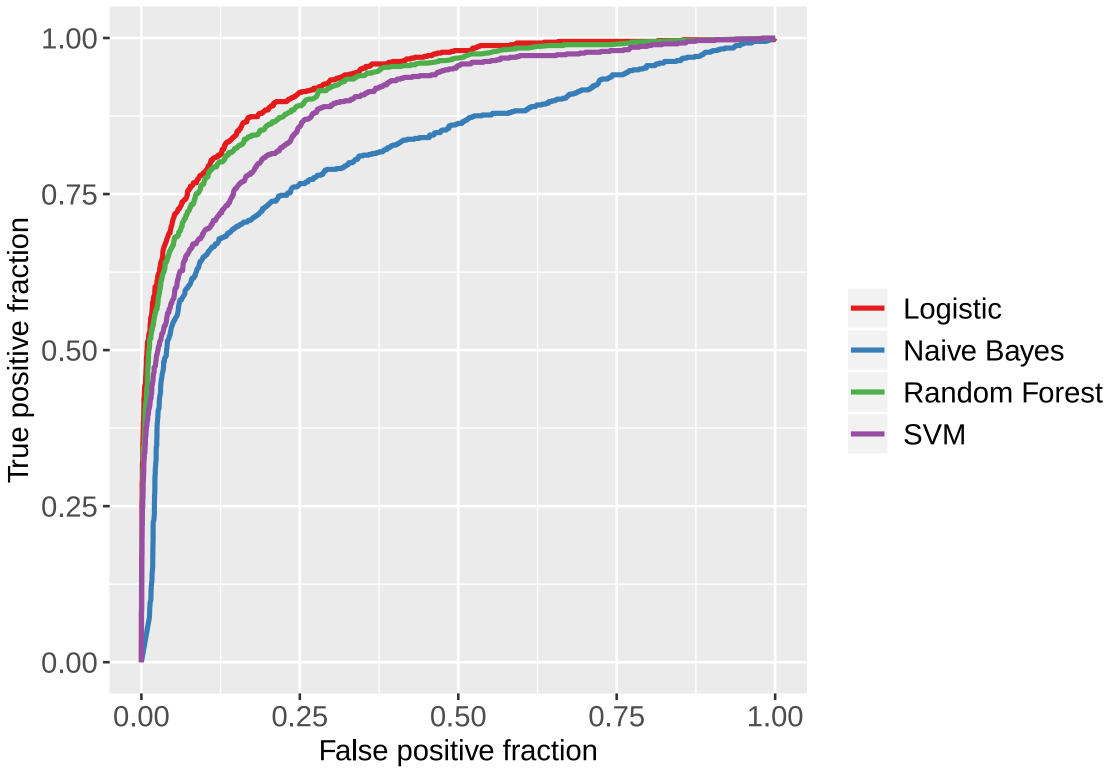


Figure 4.9: ROC curves for each algorithm (full feature set)

Table 4.1: Confusion matrix for each algorithm (full feature set)

	Actual	
Predicted	Not toxic	Toxic
Logistic		
Not toxic	4429	142
Toxic	594	604
Random forest		
Not toxic	4908	331
Toxic	115	415
Naive Bayes		
Not toxic	2905	120
Toxic	2118	626
SVM		
Not toxic	4780	312
Toxic	243	434

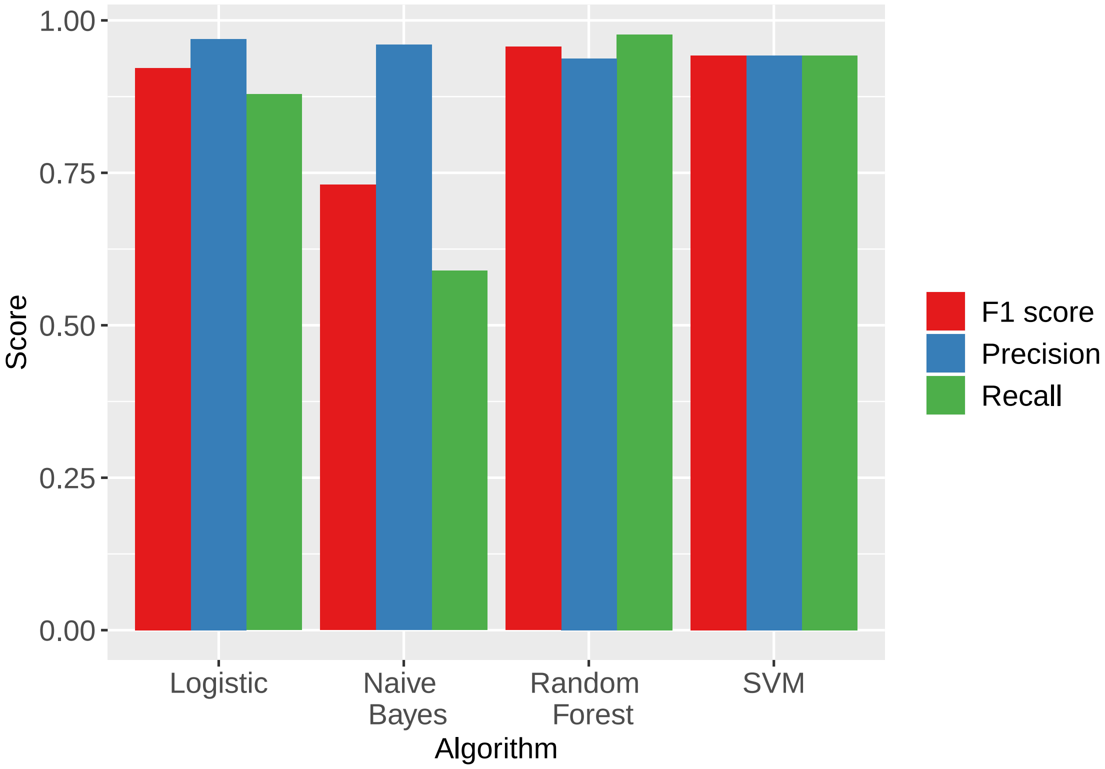


Figure 4.10: Precision, recall and F1 scores for each algorithm (full feature set)

4.3 Failure cases

As a further evaluation measure, it might be useful to look for examples that have confused the models, to help understand their limitations in detecting what is toxic and what is not

The logistic model was the least likely to miss a toxic post. However, here is an example of a comment to which this model assigned a low probability of being toxic (12%), which in fact turned out to be toxic:

> *Yeah, right. You're so "busy" you're answering questions on other people's talk pages which DON'T CONCERN YOU! lol Have you even seen Mark Walters play? ROFL!*

The use of uppercase words in the post has given the comment some probability of toxicity, but it is probably the sarcasm that has fooled the classifier. It has a number of ostensibly positive words, which give it a mean sentiment score of 2.5 because its content has been interpreted at face value, when in fact it has a negative tone.

This example was only given a 13% chance of being toxic, because it makes two hostile accusations (that the addressee is either a fascist or has a financial interest in the page they are editing), but couched in such indirect terms that it is unlikely to be detected:

> *Do you have a Hitler complex? On the 2 Unlimited page, you keep maliciously deleting a section that has everything to do with 2 Unlimited. The fan club and the author's book discusses 2 Unlimited. Are you working for the group or do you have some sort of Hitler complex? The subject matter is directly related to 2 Unlimited. Now stop deleting it.*

This toxic example was given an 18% chance of being toxic by the logistic classifier, and this low probability may be because the comment, despite containing abusive language, contains a lot of the kind of netural or technical sounding vocabulary that a non-toxic comment might contain:

> *However a removal of content is considered, a large chronology spanning a great length is poor way to put together an article. Please don't be an arsehole when writing your edit summaries. A stock test message in response to a good-faith edit is also rather obnoxious. I found one typo in the revision, which hardly warrants the summary "badly spelled". I was trying to be productive during my insomnia. Fuck you and go to hell.*

However, whether a politely expressed comment is toxic or not is not just difficult for a classifier to pick up, but it is also a bit of a grey area from an annotation perspective. For example, the following comment was classified as non-toxic (13% probability of being toxic), but annotated as toxic, when it is relatively benign compared to the previous example:

> *I expect your further cooperation in improving Barelvi Page Article which is disliked by some section. Many People regularly tries to put Negative Info about this movement. If u have time may i suggest u something from neutral point of view regarding this Article? Non constructive edits have lead this situation there. I tried a bit aggressively to tell others that people are editing it according their agenda.*

And in fact, looking at the annotations this comment received, only six of the ten annotators rated it as toxic; one fewer, and it would have been classified as non-toxic and the classifier would have been right. The mean toxicity score for toxic cases the logistic classifier missed was 0.72, compared to a mean score of 0.86 for the cases it caught, indicating that it will struggle more the less consensus there is.

The random forest model was not as good at identifying toxic posts, but was found to have high precision; when it says something is toxic, it is very likely that it is. However, an example of a comment that the random forest model assigned a high probability to (95.9%), but actually turned out not to be toxic, is:

> *Omg i love Dch!!! Who doesn't???!!! I love Pac Sun!!!*

There is not much information to go on in this short post, so the excessive use of punctuation is perhaps what the classifier has based its prediction on. However it seems in this case the puncutation has not been deployed in a sarcastic or aggressive way as it has elsewhere.

However, many of the most egregious misclassifications made by the random forest classifier are of posts that should perhaps have been classified as toxic. For example the following comments, given a probability of 99% and 96% respectively by the random forest classifier of being toxic, were not rated as such by the annotators:

> *Go stick an icepick through your sku11 and do everyone a favour*

> *well then don't undo other people's edits when you clearly don't know what you're talking about! you made a right c** of that one mate!*

This, alongside examples above of questionably toxic comments classified as such by human annotators, underlines the inherent limitation that subjectivity in labelling will place on the success of any classifier.

4.4 Testing the classifiers in a new domain

The models presented above were trained on comments taken from the user talk pages of Wikipedia. As a test of how the classifiers might perform on data from a different domain, the comments from the article pages were used as a fresh dataset on which to predict toxicity. Outcomes were predicted for each algorithm, using the model trained on all the features.

The classifiers do not appear to generalise very well. The performance metrics obtained are shown in Table 4.2. In some cases these look deceptively good; for example the random forest algorithm gives a classification accuracy of 95%. However, given the severe classification imbalance (less than 5% of the article comments are toxic), it is not that impressive. And a closer look at the other metrics for this model reveals some issues, namely that the random classifier has very high sensitivity but very low specificity; it is barely detecting any toxic cases at all.

The failings of these algorithms are further apparent if we look directly at the relevant confusion matrices (Table 4.3). The random forest algorithm fails to detect almost all of the toxic posts; it is

Table 4.2: Performance metrics using article comments as a test set (full feature set)

	Logistic	Random Forest	Naive Bayes	SVM
Accuracy	0.81	0.95	0.49	0.95
Kappa	0.17	0.06	0.03	0.08
Sensitivity	0.82	0.99	0.48	0.99
Specificity	0.62	0.04	0.67	0.06
ROC	0.80	0.78	0.62	0.71
Precision	0.98	0.96	0.97	0.96
Recall	0.82	0.99	0.48	0.99
F1	0.89	0.97	0.64	0.97

very accurate because this data is very unbalanced (less than 5% toxic), so its tendency to classify things as not toxic results in high accuracy, but it is not a useful classifier. The SVM algorithm is only slightly less conservative. The logistic model also has a high false positive rate - 18% of what it predicts as toxic turns out not to be - but is the best at identifying the toxic posts, although it is still not very good at this. The naïve Bayes algorithm has a slightly higher true positive rate than the logistic algorithm, but a very high false positive rate; well over half of what it predicts as toxic is not.

It would seem that a classifier trained on one set of words does not necessarily work well on a different vocabulary, and even the effect of the sentiment and other features does not necessarily work the same way. The new dataset contained conversations from the same website, and there is even likely to be some overlap in authorship, but these two slightly different types of discussions are not similar enough for a classifier developed on one to work on the other. This suggests that there is still a considerable amount of work to do before these classifiers could usefully be deployed in a situation other than that on which they have been trained.

4.5 Experiments in defining toxicity

This classification exercise rests upon the construction of a binary indicator of whether a post is toxic or not. So far, a post has been defined as toxic if more than half of those annotating it consider it to be. However, a comment does not need to be considered toxic by a majority of people in order for it to be alienating; it just needs to be perceived as such by one person to potentially discourage that person from further participation. Much of the online discourse that is considered unproblematic by the majority of those already participating in these spheres may in fact be considered toxic by outsiders, and this may be especially true for those whose voices are under-represented.

Three experiments were conducted to explore these issues, using the same algorithms and feature sets as the work so far, and based again on the more toxic user page comments, but using different definitions of a toxic post. By way of comparing the relative success of these models, the area under the ROC curve values obtained from the experiments are presented alongside the AUC from the Final model (i.e. that presented above) in Figure 4.11.

Table 4.3: Confusion matrices using article data as a test set (full feature set)

	Actual	
Predicted	Not toxic	Toxic
Logistic		
Not toxic	7475	162
Toxic	1678	272
Random forest		
Not toxic	9097	419
Toxic	56	15
Naive Bayes		
Not toxic	4388	141
Toxic	4765	293
SVM		
Not toxic	9026	405
Toxic	127	29

The first experiment invenstigated how well the classifier picks up comments that are not toxic by consensus, but are not considered completely unproblematic. For this experiment (Low threshold), a comment was classified as toxic if it was considered as such by two or more annotators. All classifiers performed poorly using this definition, with a maximum AUC of 0.85 using the logistic classifier; well short of the 0.93 achieved by this algorithm in the Final model.

In the second experiment, only annotations provided by women were used. Data is available on the gender of each annotator, so all comments with at least four annotations by a female annotator were considered, and comments were classified as toxic using the same threshold of majority agreement used as the Final model. This resulted in a better model than the Low threshold experiment, achieving an AUC of 0.91, although each algorithm performed slightly worse than in the final model. This is not entirely surprising, as Binns et al. [34] had previously noted that in this dataset there is less inter-annotator agreement between female annotators than male annotators in what they classified as toxic.

These results suggest that a lack of consensus affects the ability of the model to classify correctly. And indeed, in the third experiment (High threshold), when the threshold was raised to a strong consensus (at least 8 out of 10 considered the comment toxic), the model performed better. It exceeded even the Final model in performance, achieving a maximum AUC of 0.96 using the logistic regression algorithm.

The examples presented in Secion 4.3 suggested that it was often in cases of lower consensus that the classifier failed to make correct predictions. These experiments confirm this insight, and highlight a key difficulty with this type of work. The problem under investigation is the way in which toxic online content can discourage participation, and people may be alienated by material that the majority of those already participating do not perceive as toxic. However, it is in this grey area that it is hardest to obtain reliable results from a machine learning classifier,

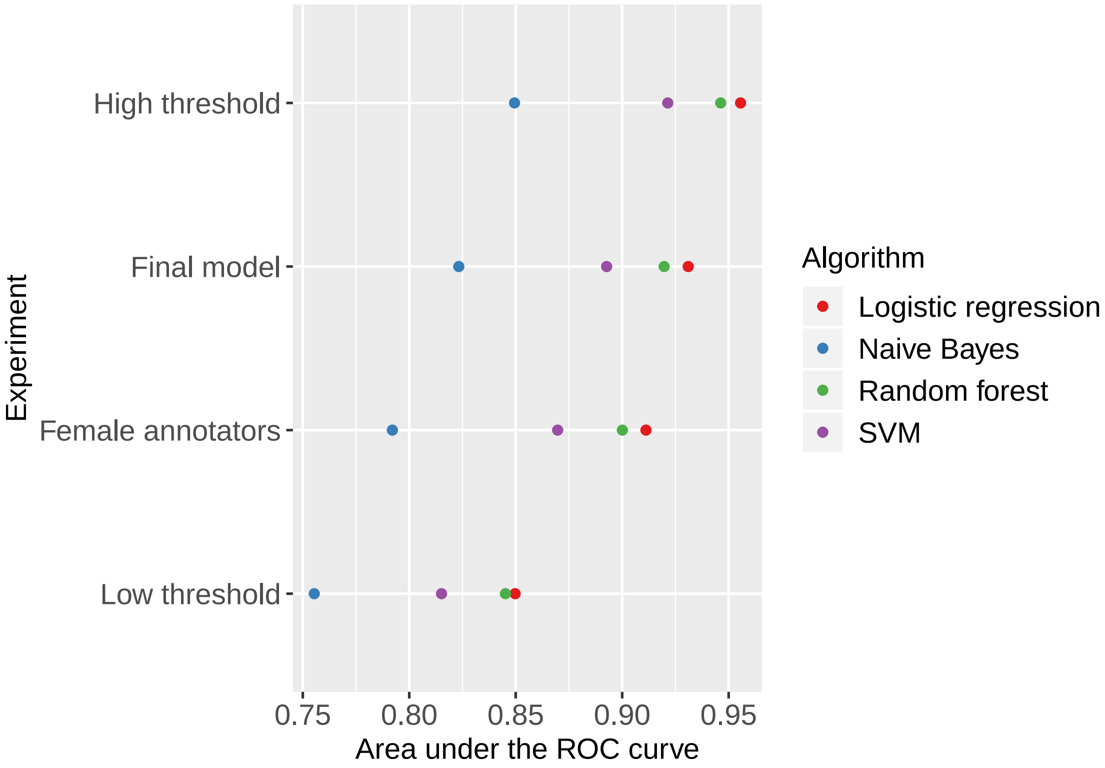


Figure 4.11: Area under the ROC curves obtained by each algorithm for each definition of toxic

undermining the ability of this type of work to make an effective contribution to a moderation system that aspires to reduce the number of ‘discouraged’ users.

5 Conclusion and evaluation

5.1 Summary

This research was motivated by a concern that a toxic atmosphere in online discourse is discouraging participation in the tech sector, particularly among those from under-represented groups. This concern has increasingly been voiced by platforms such as Wikipedia and Stack Overflow, concerned that the voluntary labour on which they rely may be alienated by this culture. However, the problem is still not widely understood in terms of what this toxicity looks like, and how a non-human system could recognise it, in order to identify toxicity in a large body of data, and flag instances to pass to the next, human, stage of moderation.

In order to explore the question of what makes a comment toxic, this project carried out a supervised learning exercise on a dataset containing comments that had been labelled as toxic or not. The definition of toxic revolved around whether the annotator felt it was a comment that would make them want to leave a discussion. Although it contained some adjectives that might describe such a comment, it did not contain a strict set of criteria for classifying a comment as toxic or not (the definition is shown in Figure 3.1 in Section 3.1). What is being classified is the reaction of the beholder of the comment, not the intent of its author, which makes the analysis particularly relevant to the question of how toxicity might be discouraging participation.

The contribution of this work has been to develop a classifier that can achieve fairly accurate results in predicting toxicity on the type of data it has been trained on. It has demonstrated the benefit of drawing upon metadata about a comment as well as the words themselves; that it is beneficial to also look for features of the way people express themselves, and the presence and strength of sentiment. However, despite the overall success of the classifier on unseen data held back from the original dataset, the work also exemplified the difficulty of transferring a classifier trained on one data source onto another.

5.2 Evaluation

This section summarises how the research followed the steps undertaken in the research, as set out in Section 1.2, and how it met each objective, and some of the particular strengths or pitfalls encountered in doing so.

Task 1: Define and understand the problem by looking at previous approaches

Chapter 2 presented literature that has used supervised machine learning to attempt to identify a range of unpleasant online conduct, from outright aggression to more subtle phenomena such as sarcasm. This generated a range of possible data sources, features and algorithms to inform the methodology of this work, and indicated a direction in which success was likely to be found in formulaing a modelling strategy to address the problem.

Task 2: Obtain suitable data

The decision was made given the time constraints of this project to use a pre-labelled dataset. There are several of these that have been deposited by their creators for further analysis, which have rarely been explored to their full potential. This research uses the Wikipedia detox dataset, which contains comments annotated for toxicity; this encompasses a range of potential specific behaviours, but is defined in terms of its effect in making people want to disengage. There is unavoidably a degree of subjectivity in classifying such a reaction, and it is clear from some of the examples presented in the previous chapter that not everything classed as not toxic would necessarily be perceived as such by everyone. This is an inherent problem in this type of work; when there is no objective ground truth, there will be a limit to the potential effectiveness of any machine learning endeavour, no matter how well designed or resourced.

Task 3: Cleaning and feature extraction

The process of turning comments into feature sets was one of finding a balance between the many possibilities available in doing so, and what was practical given the prevailing time and computing constraints. The analysis started with a basic 'bag of words' feature set, although rather than the number of times a word occurred, it calculated the TFIDF of each word. It then reduced the size of the word vectors by projecting the bag of words into a smaller feature space using principal component analysis, as a compromise between a simple bag of words and more complex word embedding methods.

The feature set also included metadata about how commenters wrote, such as the length of their comments, and whether they displayed signs of hostility or aggression such as writing in all caps, or using repeated punctuation marks. It also incorporated information about the overall tone of the comment (positive or negative), and whether it contained words pertaining to emotions that were particularly associated with one type of comment, such as trust or disgust.

Task 4: Model training and evaluation

Model training and hyperparameter search was conducted using 10-fold cross validation to avoid overfitting to the training set, and the benefit of this was reflected in the final models' good performance on unseen data. Four different machine learning algorithms were used; logistic regression as a quick baseline, and three others that had seen some success in the literature (random forest, naïve Bayes and SVM). Multiple algorithms were used because it is not always easy to know in advance which algorithms will perform best on a given dataset, so the analysis tried several that previous literature suggested might be appropriate.

Ultimately naïve Bayes, which has seen success with text classification in other studies, did not perform terribly well, but the other three performed quite well, comparing favourably to other literature in this area. The random forest model performed the best in terms of classification accuracy, and it also had the highest kappa value, indicating that it improved the most over a model that simply guessed. However, its higher success relative to the other models is due to a certain degree of conservatism; it is very good at not misclassifying a non-toxic post, but not as good as the logistic model at identifying a toxic post. Therefore, if a best model had to be chosen, it would depend on whether it is considered worse to miss a potentially toxic post or waste time on flagged posts that turn out not to be toxic. This would be a judgement to be made based on the prevailing values and resources of the specific context in which moderation is taking place.

Analysis of features that undermined the models' effectiveness suggested that sarcasm and indirect insults were not well detected, resulting in misclassification of posts as non-toxic when a human would recognise them as being toxic. However many of the problems arose in cases where the features suggested a comment should be toxic, but a majority of annotators had not classified it as such, or vice versa. The models performed best when consensus was high, which is problematic in this context, where a minority may be discouraged by what they perceive as toxic, but the majority already participating do not.

Task 5: Assist those in moderating online discussion by using the trained classifier to estimate the probability that a comment is toxic

What the model did not do very well is generalise, even to data taken from a different type of discussion on the same website. The models were trained and evaluated on data taken from user discussion pages on Wikipedia, but then subsequently tested on discussions from article discussion pages. The poorer performance of the classifiers on this data suggest that they have some way to go before they could usefully help someone moderate an online platform.

5.3 Future work

A number of steps could be taken to improve the analysis. One would be to make more context specific or theoretically informed choices in the data processing and feature selection stages. The default list of stop words supplied by the `tidytext` package might be throwing out important information, or keeping redundant information. For example, the default list includes the word ‘actually’, which in many cases is a filler word, but in this case might in fact be a common word with which a negative response is prefixed. Conversely, the list does not include commonly found words in this particular corpus, such as Wikipedia, which is common in both toxic and non-toxic comments, and thus unlikely to be contributing much information.

It could well be the case that single words have limited explanatory power, and larger phrases should be considered. However, even just the addition of bigrams would increase the computational load substantially. One compromise might be to include the presence of specific phrases typically associated with toxicity, informed by a scan of previous relevant literature in this area. Another avenue to explore, given more time and computational resources, might be the use of more complex word embeddings. The principal components analysis used here was a better way of making use of the available information than simply selecting a subset of words, and it does to some extent start to recognise the relationships between words. However, more complex word embeddings could make use of the semantic and contextual information that might identify, for example:

- whether a person is directing an insult at themselves or another person
- the proximity of intensifiers or negations that change the meaning of a word
- disambiguating, for example, whether a person is calling someone a Nazi, or having a discussion about World War II

Looking at the failure cases indicated that sarcasm was a particular subtlety that not well picked up by the classifiers. Future work could perhaps draw more on the existing literature on sarcasm to incorporate specific features that have been found to be associated with it, such as incongruities (e.g. the presence of a positive word followed by a negative word).

The analysis used a generic sentiment lexicon, but as others have found, sentiment analysis does not always cross domains very well, as words may mean different things, particularly with respect to specific technical language [66][67]. Perhaps some adjustments to generic lexicons, or the use of something more specific, could improve the ability of the sentiment analysis to more accurately gauge the tone of a post. One possibility might be a sentiment lexicon developed by Calefato et al. [68], using Stack Exchange data to try and improve sentiment detection in this kind of post.

6 References

[1] J. Silge and D. Robinson, *Text Mining with R*. O'Reilly, 2018.

[2] E. Hvitfeldt, "Using PCA for word embedding in R." 2018 [Online]. Available: https://www.hvitfeldt.me/2018/05/using-pca-for-word-embedding-in-r/. [Accessed: 1AD–8AD]

[3] J. Silge, "Understanding PCA using Stack Overflow data." 2018 [Online]. Available: https://juliasilge.com/blog/stack-overflow-pca/. [Accessed: 1AD–8AD]

[4] Tech Nation, "Report 2018 - Connection and collaboration: powering UK tech and driving the economy." 2018 [Online]. Available: https://technation.io/insights/report-2018/. [Accessed: 01-Aug-2018]

[5] Harvey Nash, "2016 Women in Technology: Career aspirations, strategies and support." 2016 [Online]. Available: https://www.harveynash.com/usa/news-and-insights/2016 Harvey Nash Women in Technology Survey.pdf. [Accessed: 01-Aug-2018]

[6] L. Mundy, "Why Is Silicon Valley So Awful to Women?" Apr-2017 [Online]. Available: https://www.theatlantic.com/magazine/archive/2017/04/why-is-silicon-valley-so-awful-to-women/517788/. [Accessed: 22-Jun-2018]

[7] J. Megarry, "Online incivility or sexual harassment? Conceptualising women's experiences in the digital age," *Women's Studies International Forum*, vol. 47, pp. 46–55, 2014 [Online]. Available: http://dx.doi.org/10.1016/j.wsif.2014.07.012

[8] S. Sobieraj, "Bitch, slut, skank, cunt: patterned resistance to women's visibility in digital publics," *Information Communication and Society*, vol. 4462, pp. 1–15, 2017.

[9] R. West and B. Thakore, "Racial Exclusion in the Online World," *Future Internet*, vol. 5, no. 2, pp. 251–267, 2013 [Online]. Available: http://www.mdpi.com/1999-5903/5/2/251

[10] S. I. Brokensha and M. S. Conradie, "(In)civility and online deliberation: readers' reactions to race-related news stories," *Safundi*, vol. 18, no. 4, pp. 327–348, 2017 [Online]. Available: http://doi.org/10.1080/17533171.2017.1335000

[11] A. Kanjere, "Defending race privilege on the Internet: how whiteness uses innocence discourse online," *Information, Communication & Society*, 2018 [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/1369118X.2018.1477972

[12] Stack Overflow, "Stack Overflow Developer Survey Results 2018." 2018 [Online]. Available: https://insights.stackoverflow.com/survey/2018/. [Accessed: 01-Aug-2018]

[13] B. Vasilescu, A. Capiluppi, and A. Serebrenik, "Gender, Representation and Online Participation: A Quantitative Study of StackOverflow," in *2012 international conference on social informatics*, 2012, vol. 26, pp. 332–338 [Online]. Available: http://ieeexplore.ieee.org/document/6542459/

[14] S. Eckert and L. Steiner, "(Re)triggering Backlash: Responses to news about Wikipedia's gender gap," *Journal of Communication Inquiry*, vol. 37, no. 4, pp. 284–303, 2013.

[15] J. Hanlon, "Stack Overflow Isn't Very Welcoming. It's Time for That to Change." 2018 [Online]. Available: https://stackoverflow.blog/2018/04/26/stack-overflow-isnt-very-welcoming -its-time-for-that-to-change/. [Accessed: 01-Aug-2018]

[16] J. Silge and J. Punyon, "Welcome Wagon: Classifying Comments on Stack Overflow."

2018 [Online]. Available: https://stackoverflow.blog/2018/07/10/welcome-wagon-classifying-comments-on-stack-overflow/?cb=1

[17] E. Wulczyn, N. Thain, and L. Dixon, “Ex Machina: Personal Attacks Seen at Scale,” in *Proceedings of the 26th international conference on world wide web - www ’17*, 2017, pp. 1391–1399 [Online]. Available: http://dl.acm.org/citation.cfm?doid=3038912.3052591

[18] Stack Overflow Meta, “How do you know Stack Overflow feels unwelcoming?” 2018 [Online]. Available: https://meta.stackoverflow.com/questions/366692/how-do-you-know-stack-overflow-feels-unwelcoming. [Accessed: 01-Aug-2018]

[19] Stack Overflow Meta, “What examples are there for Not Being Very Welcoming?” 2018 [Online]. Available: https://meta.stackoverflow.com/questions/366867/what-examples-are-there-for-not-being-very-welcoming. [Accessed: 01-Aug-2018]

[20] Stack Overflow Meta, “When is Stack Overflow going to stop demonizing the quality-concerned users who have made the site a success?” 2018 [Online]. Available: https://meta.stackoverflow.com/questions/366858/when-is-stack-overflow-going-to-stop-demonizing-the- quality-concerned-users-who. [Accessed: 01-Aug-2018]

[21] A. Braithwaite, “It’s About Ethics in Games Journalism? Gamergaters and Geek Masculinity,” *Social Media and Society*, vol. 2, no. 4, 2016.

[22] A. Massanari, “#Gamergate and The Fappening: How Reddit’s algorithm, governance, and culture support toxic technocultures,” *New Media and Society*, vol. 19, no. 3, pp. 329–346, 2017.

[23] M. Salter, “From geek masculinity to Gamergate: the technological rationality of online abuse,” *Crime, Media, Culture*, p. 174165901769089, 2017 [Online]. Available: http://journals.sagepub.com/doi/10.1177/1741659017690893

[24] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Springer, 2017.

[25] A. Kao and S. R. Poteet, “Overview,” in *Natural language processing and text mining*, A. Kao and S. R. Poteet, Eds. London: Springer London, 2007 [Online]. Available: http://link.springer.com/10.1007/978-1-84628-754-1

[26] B. Liu, “Sentiment Analysis and Opinion Mining,” *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, May 2012 [Online]. Available: http://www.morganclaypool.com/doi/abs/10.2200/S00416ED1V01Y201204HLT016

[27] T. Kwartler, *Text mining in practice with R*. Wiley, 2017.

[28] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali, “Mean Birds,” in *Proceedings of the 2017 {ACM} on web science conference - websci ’17*, 2017, pp. 13–22 [Online]. Available: http://arxiv.org/abs/1702.06877 http://dl.acm.org/citation.cfm?doid=3091478.3091487

[29] C. Liebrecht, F. Kunneman, and A. van den Bosch, “The perfect solution for detecting sarcasm in tweets #not,” *Proceedings of the 4th Workshop on Computational Approaches to Subjec-*

tivity, Sentiment and Social Media Analysis, 2014.

[30] S. O. Sood, E. F. Churchill, and J. Antin, "Automatic identification of personal insults on social news sites," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 2, pp. 270–285, Feb. 2012 [Online]. Available: http://doi.wiley.com/10.1002/asi.21690

[31] V. Kolhatkar and M. Taboada, "Constructive Language in News Comments," *Proceedings of the First Workshop on Abusive Language Online*, no. 2016, pp. 11–17, 2017 [Online]. Available: http://aclweb.org/anthology/W17-3002

[32] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Discovering value from community activity on focused question answering sites," in *Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining - kdd '12*, 2012, p. 850 [Online]. Available: http://dl.acm.org/citation.cfm?doid=2339530.2339665

[33] C. Van Hee *et al.*, "Detection and Fine-Grained Classification of Cyberbullying Events," in *Proceedings of recent advances in natural language processing*, 2015, pp. 672–680.

[34] R. Binns, M. Veale, M. Van Kleek, and N. Shadbolt, "Like Trainer, Like Bot? Inheritance of Bias in Algorithmic Content Moderation," in *9th international conference, socinfo 2017, oxford, uk, september 13-15, 2017, proceedings, part ii*, 2017, pp. 405–415 [Online]. Available: http://link.springer.com/10.1007/978-3-319-67256-4

[35] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," in *Proceedings of the naacl student research workshop*, 2016, pp. 88–93 [Online]. Available: http://aclweb.org/anthology/N16-2013

[36] C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, and C. Potts, "A Computational Approach to Politeness with Application to Social Factors," in *Proceedings of the 51st annual meeting of the association for computational linguistics*, 2013 [Online]. Available: http://arxiv.org/abs/1306.6078

[37] K. Sahay, H. S. Khaira, P. Kukreja, and N. Shukla, "Detecting Cyberbullying and Aggression in Social Commentary using NLP and Machine Learning," *International Journal of Engineering Technology Science and Research*, vol. 5, no. 1, pp. 1428–1435, 2018.

[38] M. Dadvar, F. M. G. de Jong, R. J. F. Ordelman, and R. B. Trieschnigg, "Improved cyberbullying detection using gender information," in *12th dutch-belgian information retrieval workshop (dir 2012)*, 2012, pp. 23–25 [Online]. Available: http://eprints.eemcs.utwente.nl/21608/

[39] S. Zimmerman, C. Fox, and U. Kruschwitz, "Improving Hate Speech Detection with Deep Learning Ensembles," in *LREC 2018, eleventh international conference on language resources and evaluation*, 2018, pp. 2546–2553.

[40] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing Context Incongruity for Sarcasm Detection," in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (short papers)*, 2015, vol. 51, pp. 757–762 [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0306457314000880

[41] R. González-Ibáñez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: a closer look," in *Proceedings of the 49th annual meeting of the association for computational linguistics:*

Human language technologies: Short papers-volume 2, 2011, pp. 581–586 [Online]. Available: http://www.aclweb.org/anthology/P/P11/P11-2102.pdf

[42] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks," in *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*, 2016, pp. 1601–1612 [Online]. Available: http://arxiv.org/abs/1610.08815

[43] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm Detection on Twitter," in *Proceedings of the eighth acm international conference on web search and data mining - wsdm '15*, 2015, pp. 97–106 [Online]. Available: http://dl.acm.org/citation.cfm?id=2685316 http://dl.acm.org/citation.cfm?doid=2684822.2685316

[44] S. M. Weiss, N. Indurkhya, and T. Zhang, *Fundamentals of Predictive Text Mining*. London: Springer London, 2010 [Online]. Available: http://link.springer.com/10.1007/978-1-84996-226-1

[45] K. Buschmeier, P. Cimiano, and R. Klinger, "An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews," in *Proceedings ofthe 5th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 2014, pp. 42–49.

[46] R. Justo, T. Corcoran, S. M. Lukin, M. Walker, and M. I. Torres, "Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web," *Knowledge-Based Systems*, vol. 69, no. 1, pp. 124–133, 2014 [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2014.05.021

[47] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive Language Detection in Online User Content," in *Proceedings of the 25th international conference on world wide web - www '16*, 2016, pp. 145–153 [Online]. Available: http://dl.acm.org/citation.cfm?doid=2872427.2883062

[48] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate Speech Detection with Comment Embeddings," in *Proceedings of the 24th international conference on world wide web - www '15 companion*, 2015, vol. 186, pp. 29–30 [Online]. Available: http://dl.acm.org/citation.cfm?doid=2740908.2742760

[49] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD '04 proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining*, 2004, pp. 168–177.

[50] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010 [Online]. Available: http://doi.wiley.com/10.1002/asi.21416

[51] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Antisocial Behavior in Online Discussion Communities," in *Proceedings of the ninth international aaai conference on web and social media antisocial*, 2015, pp. 61–70 [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10469

[52] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proceedings of the second workshop on language in social media*, 2012, pp. 19–26 [Online]. Available:

http://dl.acm.org/citation.cfm?id=2390374.2390377

[53] B. Lantz, *Machine learning with R*. Packt Publishing, 2015.

[54] J. Blackburn and H. Kwak, "Stfu Noob! Predicting Crowdsourced Decisions on Toxic Behavior in Online Games," in *Proceedings of the 23rd international conference on world wide web - www '14*, 2014, pp. 877–888 [Online]. Available: http://dl.acm.org/citation.cfm?doid=2566486.2567987

[55] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data mining, inference and prediction*. Springer, 2017.

[56] J. Patterson and A. Gibson, *Deep Learning*. O'Reilly, 2017.

[57] J. Fox, C. Cruz, and J. Y. Lee, "Perpetuating online sexism offline: Anonymity, interactivity, and the effects of sexist hashtags on social media," *Computers in Human Behavior*, vol. 52, pp. 436–442, Nov. 2015 [Online]. Available: http://dx.doi.org/10.1016/j.chb.2015.06.024

[58] J. Groshek and C. Cutino, "Meaner on Mobile: Incivility and Impoliteness in Communicating Contentious Politics on Sociotechnical Networks," *Social Media and Society*, vol. 2, no. 4, 2016.

[59] M. J. Moore, T. Nakano, A. Enomoto, and T. Suda, "Anonymity and roles associated with aggressive posts in an online forum," *Computers in Human Behavior*, vol. 28, no. 3, pp. 861–867, 2012 [Online]. Available: http://dx.doi.org/10.1016/j.chb.2011.12.005

[60] L. Rösner and N. C. Krämer, "Verbal Venting in the Social Web: Effects of Anonymity and Group Norms on Aggressive Language Use in Online Comments," *Social Media and Society*, vol. 2, no. 3, 2016.

[61] I. Rowe, "Civility 2.0: a comparative analysis of incivility in online political discussion," *Information Communication and Society*, vol. 18, no. 2, pp. 121–138, 2015 [Online]. Available: http://dx.doi.org/10.1080/1369118X.2014.940365

[62] M. Kuhn and K. Johnson, *Applied predictive modelling*. Springer, 2013.

[63] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," in *Proceedings of the eswc2011 workshop on 'making sense of microposts': Big things come in small packages*, 2011, pp. 93–98.

[64] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, Aug. 2013 [Online]. Available: http://doi.wiley.com/10.1111/j.1467-8640.2012.00460.x

[65] M. L. Williams, "Data journalism and the ethics of publishing Twitter data." 2018.

[66] N. Imtiaz, J. Middleton, P. Girouard, and E. Murphy-Hill, "Sentiment and Politeness Analysis Tools on Developer Discussions Are Unreliable, but so Are People," *IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering*, vol. 18, no. 2, 2018 [Online]. Available: https://doi.org/10.1145/3194932.3194938

[67] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, "On negative results when using sentiment analysis tools for software engineering research," *Empirical Software Engineering*, vol. 22,

no. 5, pp. 2543–2584, 2017.

[68] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, “Sentiment Polarity Detection for Software Development,” *Empirical Software Engineering*, no. 2017, pp. 1–31, 2017.

Appendix

6.1 Full table of results

Table 6.1: All evaluation metrics for all feature sets and algorithms

	Accuracy	Kappa	Sensitivity	Specificity	Area under ROC	Precision	Recall	F1
Logistic								
Bag of components	0.84	0.47	0.85	0.78	0.90	0.96	0.85	0.90
Sentiments	0.80	0.37	0.81	0.71	0.84	0.95	0.81	0.87
Features	0.82	0.34	0.85	0.58	0.76	0.93	0.85	0.89
BOC + sentiments	0.87	0.54	0.88	0.80	0.92	0.97	0.88	0.92
BOC + features	0.85	0.49	0.86	0.79	0.91	0.97	0.86	0.91
All	0.87	0.55	0.88	0.81	0.93	0.97	0.88	0.92
Random forest								
Bag of components	0.91	0.54	0.97	0.51	0.89	0.93	0.97	0.95
Sentiments	0.81	0.38	0.83	0.70	0.84	0.95	0.83	0.88
Features	0.81	0.35	0.84	0.61	0.78	0.93	0.84	0.89
BOC + sentiments	0.92	0.61	0.97	0.57	0.91	0.94	0.97	0.96
BOC + features	0.91	0.56	0.97	0.51	0.90	0.93	0.97	0.95
All	0.92	0.61	0.98	0.56	0.92	0.94	0.98	0.96
Naive Bayes								
Bag of components	0.49	0.11	0.43	0.86	0.80	0.95	0.43	0.59
Sentiments	0.78	0.34	0.79	0.72	0.81	0.95	0.79	0.86
Features	0.86	0.20	0.96	0.19	0.75	0.89	0.96	0.92
BOC + sentiments	0.50	0.13	0.44	0.88	0.82	0.96	0.44	0.61
BOC + features	0.54	0.14	0.50	0.83	0.80	0.95	0.50	0.66
All	0.62	0.20	0.59	0.84	0.82	0.96	0.59	0.73
SVM								
Bag of components	0.89	0.51	0.93	0.61	0.87	0.94	0.93	0.93
Sentiments	0.82	0.39	0.83	0.70	0.83	0.95	0.83	0.89
Features	0.79	0.31	0.81	0.62	0.76	0.94	0.81	0.87
BOC + sentiments	0.90	0.55	0.94	0.61	0.89	0.94	0.94	0.94
BOC + features	0.89	0.52	0.94	0.59	0.88	0.94	0.94	0.94
All	0.90	0.56	0.94	0.61	0.89	0.94	0.94	0.94

6.2 Hyperparameter tuning

Hyperparameter combinations were tested using 10 fold-cross validation, repeated 10 times. Kappa was the metric used to select the best model, which is shown at different hyperparameter settings in these plots. The plots here show the tuning process for the full model (with all features), but this process was carried out for each combination of features to ensure the optimal model for that combination of features.

The regularised logistic regression models had two tuning parameters. Alpha (the balance between L1 and L2 penalisation) was tested at four values (0,0.1,0.5, and1), and lambda (the extent to which regularisation weights were applied) was varied over a sequence of 10 different values between 0.000001 and 0.01. Figure 6.1 suggests that the optimal values of alpha was in fact zero, with lambda making no difference.

The tuning parameter on the random forest models was the number of random features used to make the trees. The number of permutations tested depended on size of feature set, but comprised a number of values between the a small number (2) and a high number (almost all the features) – in the case of the final model, 9 values between 2 and 59. Figure 6.2 suggests a medium number of random features was best in this case.

For the Naïve Bayes model, various degrees of Laplace smoothing were tried: 0, 0.5 and 1. Figure 6.3 suggests equally optimal performance at levels of 0 or 1.

The SVM model was estimated over a range of 10 cost values between 0.05 and 1. Figure 6.4 suggests that there was an initial advantage to increasing cost, but this fell off rapidly beyond the optimal value.

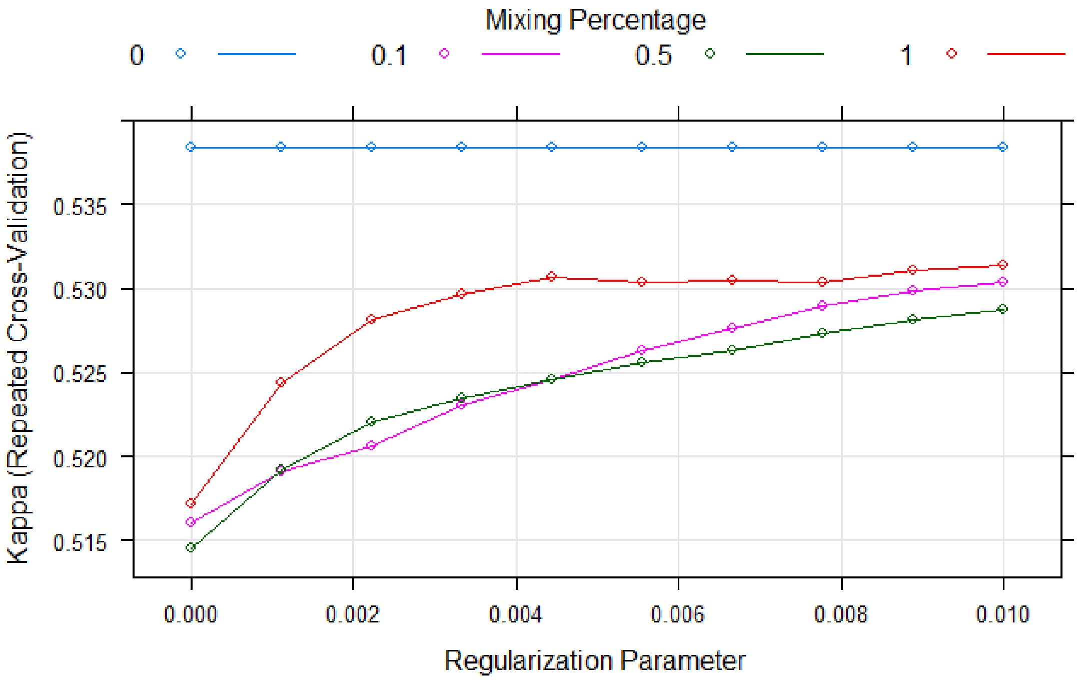


Figure 6.1: Outcome of hyperparameter tuning in regularised logistic regression model

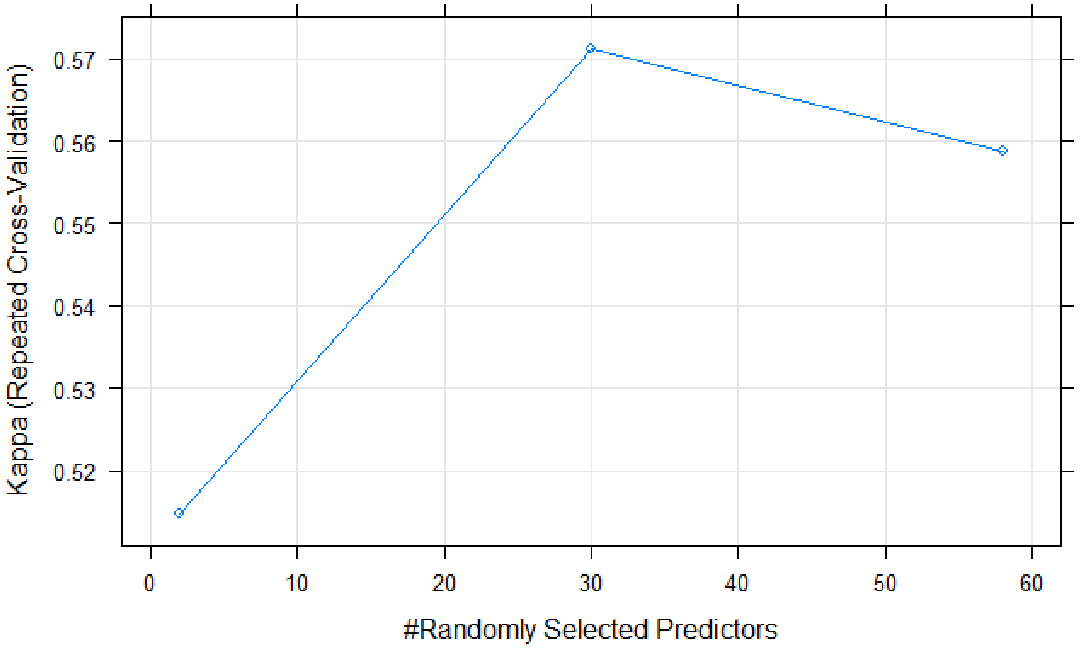


Figure 6.2: Outcome of hyperparameter tuning in random forest model

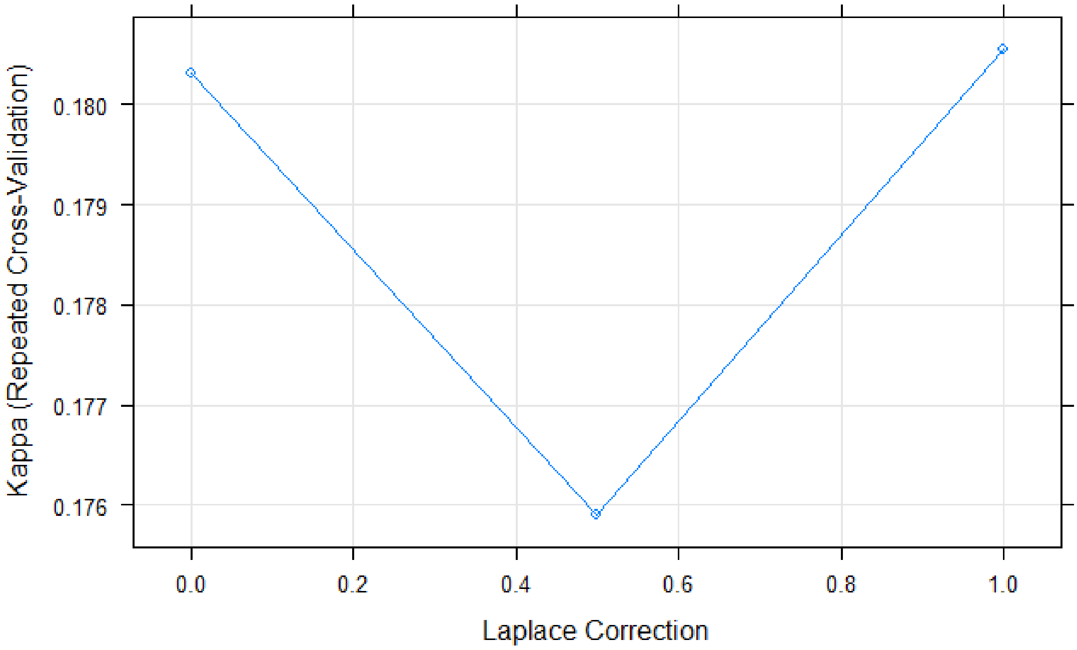


Figure 6.3: Outcome of hyperparameter tuning in naive Bayes model

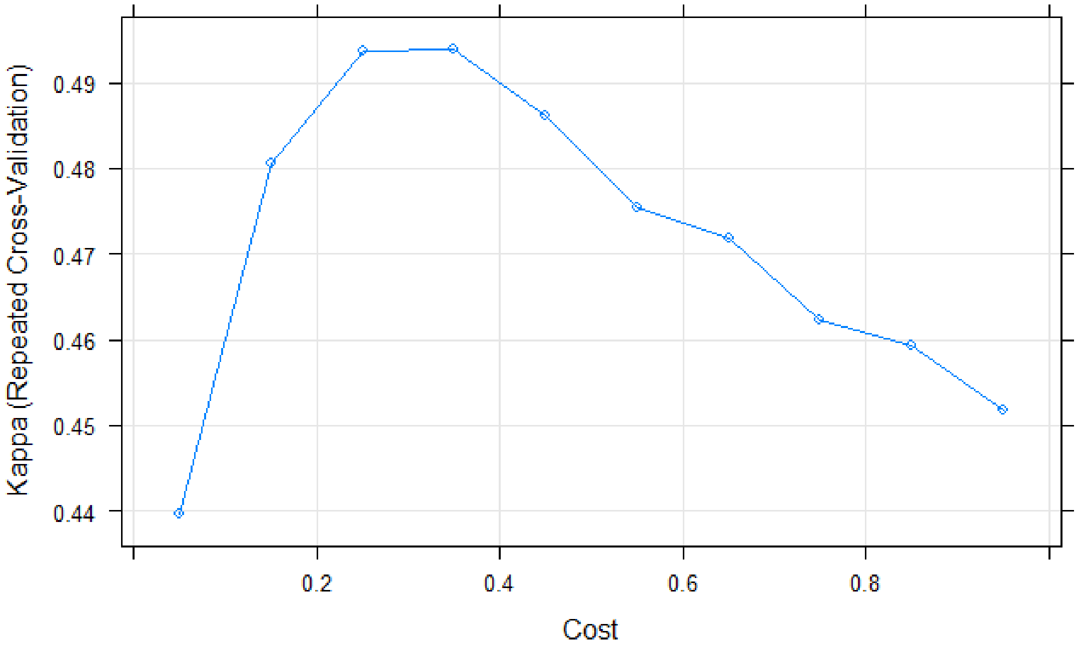


Figure 6.4: Outcome of hyperparameter tuning in SVM model