Detecting Online abuse on Twitter using Machine learning and Lexicon approaches

Junayid K Chowdhury - 001044060

May 11, 2021

A Dissertation project completed to conclude a Degree of BSc H Computer Science at Greenwich University

Abstract

Automating the detection of abuse on Twitter is a difficult task. With a significant volume of data being created by the second. This paper aims to delve into lexicon and machine learning approaches to detecting abuse on Twitter. Utilising a TFIDF SVM model, the popular AFINN lexicon, while also exploring the concept of a supervised learning-based lexicon. It looks to compare these models and mention their success while exploring the differences in fundamentals between lexicon and machine learning approaches. The 3 methods are thoroughly tested using both validation testing and real-world testing with the use of an anonymous survey. This project looks to then highlight the strengths and weaknesses of the models and contributes to future discussion on how to improve these methods to create a real-world application to detect abuse on Twitter.

Acknowledgements

This project has been completed under the guidance of both Mohammad Majid Al-Rifaie and Jing Wang. I would like to thank them both for valuable insight and help in writing this dissertation. I would also like to thank my friends and family for providing much needed moral support during these stressful times.

Contents

1 Introduction

Twitter is a social media site created in 2006, becoming one of the pioneering leaders of the social media boom. The site is commonly used for users to express their thoughts, keeping up with their favourite celebrities or connecting with people who often share similar interests. Over the years Twitter has garnered a significant user base. Twitter has approximately 187 million daily active users. it sees an average of 500 million tweets a day, coming to almost 6000 tweets a second Smith (2021). With large amounts of data being produced in a short time continuously, Twitter has been subject to many journals, covering a broad array of topics. The micro blogging system of twitter opens up for many instances of research into sentiment analysis. One example is the study performed by Liu et al. (2017) who incorporates an automated approach to tweet extraction utilising sentiment analysis. Creating big data used by brands to maximise their advertising procedures.

Twitter's very own CEO Jack Dorsey is on record for stating the following "We suck at dealing with trolls and abuse". Hern (2015) with the exponential growth that social media has seen since its rise around the mid-2000s. A very apparent problem has been chipping away at the growing phenomenon that is the online social web, the use of consistent and prevalent abuse. An aspect as deep-rooted in social media as abuse presents an almost insurmountable problem. With Twitters current methods being very trivial. Twitter has a major reliance on user-based reporting to be able to flag tweets for further checks. One of Twitter's senior directors, Gasca (2019) states only 38% of abusive content is surfaced proactively without reports from users. This reliance on manual and cohesive user interaction is not sustainable for a website with a recorded 353 million monthly users Dean (2021). Abuse on Twitter is a diverse spectrum that contributes to the difficulty of managing it. Abuse ranges from affecting football stars to even the most localised of bullying within communities. Swanson (2021) reports in an interview with Arsenal's Ceo Vinai Venkatesham states "Nothing is off the table " in regards to a social media boycott due to the abuse suffered by players. The works of Chatzakou et al. (2017) highlights the unhealthy evolution of bullying. The addition of social media like Twitter allows for cyberbullying. What was once limited to within school boundaries can now be experienced and occur anytime.It is evident that the wide range and reach of platforms like Twitter and their excess of abuse. Presents a contemporary issue that is completely relevant in the modern era.

This paper explores a much more automated approach to detecting abuse rather than Twitter's current reliance on a manual reporting method. It utilises the likes of NLP and text classification to explore this option further. The use of natural language processing methods allows for the breaking down of tweets. Which can be treated through computational means. The methodologies identified in this paper are a machine learning approach and lexicon approach's. Both have been widely used across sentiment analysis and have proven to have both positives and negatives associated with them. This paper looks to compare the two methods and explore the use of a new lexicon approach which utilises supervised learning.

The first method identified as a viable option was a machine learning approach in a TFIDF-SVM model. Machine learning has been shown to provide consistent results within sentiment analysis and text classification. Within this project, the only goal was to identify whether a tweet was considered abusive. SVM's binary classification was well suited in this abuse classification task. Machine learning's dynamism allows it to be improved with the use of better datasets and different types of data. Machine learning methods are also computationally less exhausting than methods like deep learning, which is much more realistic with the computing power used for this project. Machine learning models are easy to evaluate with the likes of inbuilt metrics like Precision, Recall and F1-Score. The ease of evaluation also allows for a clearer image of the success of its ability to detect abuse.

Lexicon methods utilise a large premade corpus, containing words given a sentiment value. The unseen text is then read with a lexicon tally counting how many words within the lexicon are spotted. Based on the words counted, the sentence is assigned a sentiment score.

This paper uses two instances of a lexicon approach. One lexicon used was the Afinn lexicon developed by Nielsen (2011). Afinn is a widely used lexicon containing 3300+ words. Each word has an associated polarity score and helps to generate a sentiment. Another lexicon method adopted in this method was a look at a potential new approach to Lexicons. A TFIDF Lexicon that can retrieve the highest importance words within a dataset creating the corpus automatically, a form of supervised learning. These words are then given positive or negative value and are used in testing unseen data in a traditional lexicon style. Lexicon methods can prove to be harder to evaluate than methods like Machine learning and deep learning. The lexicon methods were both tested on the same test data examined by the SVM but without the detail of evaluation examined in machine learning. To combat this, additional metrics were adopted through the use of a survey.

An anonymous survey was carried out on a sample of tweets classified by the three methods. These tweets were retrieved by the use of the integrated Twitter API. Participants were asked to highlight which tweets they considered abusive or not, the results of the participants were then cross-validated across the chosen methods to retrieve additional evaluation criteria. These results were then used to analyse the success of the methods.

Following the use of extensive evaluation, the project helps to highlight issues that are affecting the ability of automating the approach of abuse detection. Discussing lexicon faults and and issues with viable datasets. While also giving an overview on how these methods can be adopted and improved to create better abuse detection within social media.

2 Literature Review

This section of the report aims to highlight the work of other academics and their attempts in trying to contribute to the research of twitter abuse detection. Problem domain looks to highlight some of the major issues that have been considered in abuse and the multiple methodologies put forward to attempt to counter them. Giving variation in methods covered. It also looks to explore the use case and effectiveness of machine learning and lexicon methods in fields outside of social media and abuse detection.

2.1 Problem domain

2.1.1 Context

Sentiment analysis at a fundamental level performs to a very high degree when working with basic sentences, recognizing rather easily the difference between a sentence of positive or negative sentiment. However, when we introduce levels of English where the sentence structures lack context, sentiment analysis begins to struggle greatly. the work of Chakrabarty et al. (2019) helps to show the limits of using a traditional " bag of words" method. Instead, they present a deep learning approach where they uniquely incorporate a "contextual attention vector". When training data is used the context of the data is labeled as well. This leads to Chakrabarty et al. (2019) getting an algorithm that had learned to be better at identifying tweets that may lack context and still giving an accurate sentiment score. In a lexicon approach Bross & Ehrig (2010) were able to overcome the issue of context in sentiment analysis (concerning product reviews) by having two factors affect the semantic orientation; an opinion factor and a product aspect. They made the opinion factor affected by the context in relation to the product aspect. By doing this they are able to calculate the sentiment of a statement as negative or positive "low price vs low battery drain" Bross & Ehrig (2010)

Judging context can prove to be an important element to consider when trying to perform sentiment analysis on Twitter. With Twitter's 280 character limit, often tweets are lacking context and have been shortened to accommodate. Incorporating context-awareness can help to address this issue and also increase the accuracy of results

2.1.2 Emoticons

Twitter as a modern-day social media encompasses the use of many emoticons. Naturally, this leads to needing to incorporate emoticons into sentiment analysis.Yamamoto et al. (2014) decided to tackle the issue by using two lexicons in coercion. One standard lexicon but also a lexicon specifically just for emoticons. When using two lexicons in conjunction they found their algorithm would demonstrate drastically different results when compared with just the use of a standard lexicon approach. in their evaluation by using user trials and cross-referencing their own results, they were also able to prove the large extent to which emoticons have on the sentiment of a tweet. They found the sentiment of a tweet would often drastically change when including or excluding an emoticon. A similar approach was also echoed by other researchers like Hogenboom et al. (2013))showing that the use of an emoticon lexicon performs well and achieves desired results. Given the volume of emoticons likely to come across in this topic area, the works of Hogenboom et al. (2013) and Yamamoto et al. (2014). Has given a good insight into how to deal with the potential issue of emoticons in sentiment analysis.

2.1.3 Sarcasm

A complex level of natural language that machines often find difficulty in reading accurately, is the use of sarcasm. Sarcasm is prevalent throughout Twitter due to its informal nature and short sentences. In the works of Bouazizi & Ohtsuki (2015), they devised a method to detect sarcasm in sentiment analysis. After creating a clear definition of sarcasm they set up a feature extraction method specific to their definition of sarcasm, looking for the following in their data sets "Punctuation-related features, syntactic features, sentiment related and pattern related features". They further enriched their algorithm by incorporating a data set including 37,918 tweets, half of which had the sarcasm and half which did not. This lead to the algorithm gaining a lot of training data just specifically for sarcasm. They found their algorithm outperform the likes of "the ngram-based approach " and another method " proposed by Riloff et al. (2013). The method proposed by Riloff et al. (2013) falters, in that its definition of sarcasm is one dimensional, only considering a passage to be sarcastic when assuming " that many sarcastic tweets contain both a positive sentiment and a negative situation in close proximity". Allthough Riloff et al. (2013) are able to successfully analyze some sarcastic tweets, many do not fall under their definition.

The work documented by Bouazizi & Ohtsuki (2015) and Riloff et al. (2013) helps to demonstrate that using sarcasm to enhance sentiment analysis can prove to be a vital step in creating proficient sentiment analysis, especially in the realm of Twitter where sarcasm is used frequently

2.1.4 User-based approach

Chatzakou et al. (2017) cover a methodology specifically for detecting aggression and bullying on Twitter, but with a focus on detecting users. They can work around the issues of context by using a method called "Sessionization" using groups of tweets from users (as opposed to individual tweets) to create their database. This gave them an original approach to the problem of detecting abuse, as they looked to detect abusive users rather than individual tweets. They utilized crowdsourced labeling, to label users in one of three categories " Aggressor, bully, and spammer". All be it a more expensive approach the use of crowdsourced labeling helped them create a specific dataset that would cause low recall for their proposed method. Sanchez & Kumar (2011)follows a similar approach and focuses more on specific users as opposed to tweets. After registering a negative tweet from a user, their activity is monitored, and the activity of the followers as well. By doing this Sanchez & Kumar (2011) begins to create a social graph of potential bullies and their victims. Sanchez & Kumar (2011) and Chatzakou et al. (2017) help demonstrate a completely different approach to the problem, utilizing a more user-based approach to detecting abuse online while still using sentiment analysis

2.1.5 Using Optimal Datasets

Pereira-Kohatsu et al. (2019) worked on creating a system to detect hate speech on Twitter. They emphasise that they hope their system can be used to detect negative trends and abusive content to such an accurate level. That it can also be used to provide valuable information to security and police forces as an attribute for crime forecasting.

Pereira-Kohatsu et al. (2019) model contains two major functions. Hate speech detection and a social network analyser. Their method for hate speech detection includes the use of a raw dataset with 2 million tweets. The dataset was put through a combination of filtering techniques. They utilised multiple dictionaries to create an initial supervised classification of the dataset. 6 dictionaries used, contained hate speech of different categories. With one additional final dictionary being used that contained general insults. These were used to filter tweets that may contain even the slightest bit of abuse. Tweets flagged by the filtering system as abusive was sent to the training data. After this system had been utilised, only 8710 tweets were selected for training data out of the original 2 million. Following this initial filtering, the tweets were then further labelled by 4 experts. This created a rich dataset with 26% of the corpus being labelled as hate containers and 74% being labelled as non-hate containers. With 6000 tweets in the overall dataset.

Pereira-Kohatsu et al. (2019) Paper highlights the importance of a usable dataset. The excessive techniques and processes used to garner a dataset that could be used within abuse detection. After the numerous techniques used including the likes of expensive manual labelling, Pereira-Kohatsu et al. (2019) were still only able to garner a dataset of 6000 tweets. This highlights a problem for

Twitter abuse detection. That the are very few datasets that are created to the high level needed to create optimal machine learning models.

2.2 Related Works

2.2.1 Lexicon

One lexicon that sees wide use in sentiment analysis problems is the AFINN lexicon developed by Finn Arup Nielsen. Nielsen (2011) paper delves into the negatives associated with the other available lexicons in 2011. Nielsen (2011) made his Lexicon approach unique by including popular internet slang and additional rules like sentiment strength. In the paper, He details the creation of the lexicon utilising sites like Urban Dictionary and Wiktionary while also citing many other lexicons creating an amalgamation of multiple sources. Nielsen (2011) Evaluates the performance of his AFINN Lexicon on 1000 tweets labelled by AMT, the results were then compared with ANEW. Bradley & Lang (1999) created ANEW, a project seeking to provide a set of normative emotional ratings for a large portion of the English language. Nielsen compares the results of his AFINN Lexicon against ANEW. Nielsen's results proved that the lexicon was better suited for short forms of text as opposed to ANEW which created better results on longer text. Following Nielsen's research AFINN has become a dominant lexicon especially within the realm of sentiment analysis for shorter bodies of text.

Aung & Myo (2017) Assessed the use of a lexicon-based approach to analysing the sentiment of student feedback to lecturers. Aung & Myo (2017) Created a lexicon corpus prioritising opinion words for teacher evaluation. Their lexicon consists of 745 words which were examined by two teaching evaluation experts to measure and label the sentiment scores for each selected word. Many words would hold different scores based on their polarity, allowing for specific words to hold deeper quantifiable scores, within the context of teacher evaluation.

Granted that quantitative data is easier to process, qualitative data in which students give extensive written feedback is where a Lexicon method would help generate data. Thus Aung & Myo (2017) highlight the importance of being able to analyse sentiment through a lexicon as it allows for the further interpretation of lecturer performance. The use of a Lexicon method creates opinion scores on lecturers that can be used to call attention to lecturer performance.

Their lexicon was then tested on feedback comments from fourth-year students. A range of scores was generated for the 12 teachers selected. These scores were examined and then compared with the widely used AFINN lexiconNielsen (2011) . The AFINN lexicon and proposed lexicon both performed similarly but with slight differences. The difference in results showed that the context-induced teacher evaluation lexicon was more insightful on the student's comments. This is most likely due to the process in which the lexicon was created. Given that the lexicon was manually made with teacher evaluation words and label scores the lexicon is better suited to tackle the test data. This demonstrates that the creation of a specific lexicon within a problem domain allows for better scores than general lexicons. But given the results were very similar it also highlights the AFINN lexicon is good as a baseline to compare lexicon methods.

Overall this paper works well in demonstrating the potential of lexicon-based methods for sentiment analysis. They chose a specific topic and were able to use teaching evaluation experts to create the lexicon, giving their lexicon more validity, as opposed to assigning scores themselves. They test their method on primary source data from genuine student comments. One negative however is, there could have been further evaluation metrics taken to measure the success of their method as opposed to a flat comparison with the Afinn lexicon's scores.

2.2.2 Deep learning

Deep learning is a subset of machine learning that incorporates a different set of mathematical fundamentals to create a ML model. Deep learning algorithms all form their basis on the neural network algorithm and work through layers to teach the model. Severyn & Moschitti (2015) explore the use of a deep convolutional neural network (CNN) to perform Twitter sentiment analysis.

This paper helps to highlight the benefits of deep learning methods for sentiment analysis. Severyn & Moschitti (2015) look to contribute a way to optimise the parameter weights within the CNN, without the use of additional features.Severyn & Moschitti (2015) experiment with the use of a single layer CNN trained on a larger corpus. The single layer is programmed to look for patterns that include word sequences, for example, discriminative sequences of words. Their model is trained on nearly 60 million tweets using different methods for labelling amongst the data. The training of their model is recorded to have taken several days. Once their model was trained they evaluated their proposed method against other forms of their CNN with different parameters. It is seen that their proposed 3 step process to train parameters performs better overall across the 4 datasets tested.

One model that was considered for this project was an LSTM approach to sentiment analysis. Jing & Xu (2019) Performed a survey on an abundance of neural network models. in their evaluation of the LSTM model they also highlighted its value in being able to solve the vanishing gradient issue stating that it performs well. However Jing & Xu (2019) mentions how it is difficult to train the LSTM approach on a large corpus, as it is time-consuming. Given its excessive time consumption, better performance is to be expected. LSTM provides a great counter to vanishing gradient with its long term and short term memory. Yet, due to its computational appetite and slow training times. LSTM would not be the best suited for this current project. The computational power required to be true to the methods best qualities would not be met in this project. Although deep learning is shown to have a greater understanding of reading sentiment through their results. Both papers help to demonstrate the computational complexity of training deep learning models. The dataset used by Severyn & Moschitti (2015) took several days to train and was being fed datasets with millions of tweets. Deep learning although better at providing better sentiment analysis, is too computationally expensive to consider appropriately in this project. Machine learning methods like SVM are better suited to the computer power used in this project.

2.2.3 SVM

Colas & Brazdil (2006)delve into the success of SVM in text classification, and creates comparisons with other machine learning models. They examine the likes of K Nearest Neighbour algorithm and Naïve Bayes. Colas & Brazdil (2006) hope to answer the question on whether the SVM should become the standard solution for text classification. By comparing with "older" methods to understand if they have much to offer over SVM within the realm of classification.

The paper examines the model's ability to classify the popular "20newsgroups" dataset. which is a dataset containing 20,000 emails, with 190 classifications tasks. They were also examined on how they react to a learning setting with an increasing number of documents. The methods' were then analysed on their performance in metrics like recall, precision, F1 Score and computational processing time.

It was initially examined that the 3 classifiers behaved similarly within the classification task. With differences becoming minimal as the total number of documents increased. Yet SVM was seen to hold a significantly longer processing time than the other 2 despite sharing similar results. Colas & Brazdil (2006)explains how SVM is better suited for a problem domain that contains more features, and when this condition is not met SVM performs closely to other ML models. Colas & Brazdil (2006) states that SVM's main strengths are seen when working with non-linear data.

In general, this paper helps to portray the SVM model would be better suited for the likes of abuse detection as it performs much better when multiple features are to be classified. Within the problem domain of Twitter abuse, many categories can be defined. Abusive, Hate speech, racism etc could all be differentiated. Establishing these categories to create distinct features to train the SVM model would be an optimal way to attempt countering abuse. While also optimising the use of an SVM model. This helps portray a valid reason why SVM is a preferred model for Twitter abuse detection

3 Requirements

3.1 Functional Requirements

These are the requirements that must be met to establish that the project has implemented the necessary components to meet its significant goals. Highlighting its overall functionality.

- Create a cleaning function that incorporates techniques to clean data.
- Access training and validation datasets
- Perform text vectorisation utilising TFIDF
- Train an SVM model on training data
- Create TFIDF Lexicon, calling keywords from 3 datasets
- Call TFIDF Lexicon and create a function to generate scores of 0 and 1
- Call AFINN Lexicon from library and create a function to convert polarity scores to 0 and 1 labels
- Fetch tweets through Twitter API and then have all 3 models run on fetched tweets
- Print 3 sets of classification predictions on the tweets fetched
- Create a survey to generate user-based classification of fetched tweets, to allow for further comparison of methods in results

3.2 Non Functional Requirements

Non-functional requirements help determine the success and usability of the program. The following requirements are used to increase the overall practicality of this program

- Reduce computational times
- Include Visualisations to allow for an understanding of validation results with Lexicon methods.
- Adjust hyperparameters to increase results
- Adjust TFIDF Lexicon to increase results

3.3 Comparison of systems

Aung & Myo (2017) performs a similar project in that they create a unique lexicon and look to test against AFINN. They also expand this further by generating new data with the use of a survey. This helps propel some of the evaluation metrics used in this project. One choice made was to adopt a survey to test the effectiveness of lexicon methods, this was because lexicons are harder to evaluate with just the use of a validation dataset. in addition, comparing to a traditional lexicon allows for a broader picture of how the lexicon is performing overall.

3.4 Evaluation Requirements : Testing and Validation

The comparison of the three methodologies requires a uniform evaluation. One of the requirements should be to create a feasible and fair assessment of all three of the methods covered. Although Machine learning methods have some level of evaluation with their inbuilt metrics across a test dataset. Lexicons are much harder to evaluate. Due to this a much more thorough evaluation method must be adopted.

This paper adopts an evaluation method that utilised both a validation dataset and a test dataset. Ripley (2007) expresses the significance of using both validation and testing data in machine learning. Ripley (2007) describes validation as a set of examples that are used solely to maximise the effect of hyperparameter tuning on the classifier. While the final test dataset serves the purpose of fully examining the success of a completely completed classifier. This paper adopts this technique to assess the SVM model but also uses this technique to optimise the lexicon created.

The validation dataset used is the test split for the SVM model. This presented a large corpus of labelled tweets that could be used to measure the accuracy of all 3 models. Both Lexicons and SVMs will be examined on how many tweets they have classified compared with the true values of the test split dataset. It will also be used to examine what further adaptions can be made on the 3 methods to improve their initial results. Adjustments will be made in accordance to how the model's predictions compare with the true values. Once the optimal results have been reached the models will then be moved on to the next phase of testing.

3.5 Understanding Classification Metrics

Machine learning metrics that were specifically analysed for drastic changes were Recall, Precision, F1 Score, Accuracy and the classification matrix.

Accuracy represents the total number of correct predictions made by the model which is then divided by the total number of predictions the model made. For example, if the was 1000 messages within a test set of data and the model correctly predicted 670 of these. This would equate to 67% accuracy (670/1000)

To be able to analyse the remaining metrics further, an understanding of True positive and False negatives etc is required.True positive and true negative represent when labels are guessed correctly completely, with real value and predicted value being the same. While false negative and false positive represent labels that were predicted incorrectly.

Recall assesses the model's ability to correctly find all the appropriate values within the dataset. It is given by this formula:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{1}$$

Precision looks to evaluate a models ability in identifying only the relevant data points. Given by the formula:

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2}$$

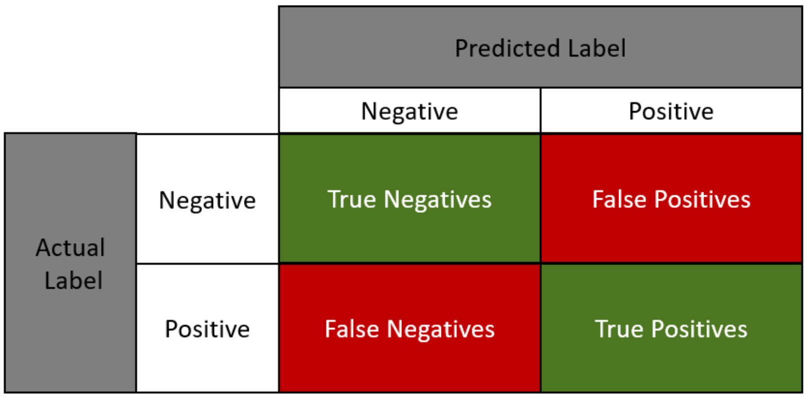
F1 Score provides a combination of both precision and recall. It is the harmonic mean of both values. The harmonic mean is adopted instead of simple mean as it helps to punish and avoid skewing the value by extreme values. F1 score gives an overall evaluation of how the model can perform both precision and recall.

$$F1Score = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The confusion matrix will allow for an overview of the model's ability to correctly predict labels and differentiate between labels that might prove difficult. This is by displaying the true positive and true negative values. True positives and true negatives being considered as good results as they represent completely correct classifications. False positives and false negatives representing miss classifications.

Confusions matrixes detailing high false positives are highlighted as worrying results. This is because a false positive means mislabelling of a negative label. An example of this would be if negatives represented cancer and positives as cancer free, a false positive would mean to classify a patient suffering with cancer as cancer free this is evidently problematic as they would be cleared to continue and would mean further inspection never occurs. Yet false negatives would entail labeling a cancer free patient as having cancer, this is less severe as it would just mean further examination would highlight this miss classification all round limiting consequences.

Within abuse detection the same applies, miss labeling negative tweets as positive would mean the system would allow for abuse to occur never once highlighting an abusive tweet for further inspection. Whereas false negative tweets would be inspected at a later date and allow for a recalling of an retrospective ban or report. False positives are often referred to as a type 1 error with False negatives being referred to as type 2 errors.



		Predicted Label	
		Negative	Positive
Actual Label	Negative	True Negatives	False Positives
	Positive	False Negatives	True Positives

Figure 1:

4 Design

4.1 Importance of Python

Python was the chosen language adopted in this project, due to its relative simplicity and common use in artificial intelligence. While also being one of the most popular languages, this meant the was massive support online with an extensive community. This allowed for easier coding, as many queries and solutions had been posted online. This aided in making actual programming easier, with plenty of aid online. Python incorporates the use of numerous libraries that were fundamental to the success of this project. These include libraries like, NLTK, SKlearn, Pandas etc. All of which see common use within machine learning

Natural language processing (NLP) is a subset of python coding that focuses on developing applications to work on human language. It is employed in a wide range of uses. Language translation, email filters, image captioning and many more etc. NLP is applied through the use of the Natural Language Toolkit (NLTK) library. The NLTK library is open-source developed by Loper & Bird (2002). The library is regularly updated and has a large community, is widely used for natural language processing. it is the most popular library in NLP. NLTK allows for the use of various crucial techniques needed within this project like tokenisation, lemmatisation etc.

Scikit-Learn is a library initially designed by David Cournapeau in 2007 Pedregosa et al. (2011). It builds upon many of the principles created in the SciPy stack which includes the likes of NumPy, Matplot and pandas which were also used in this project. Scikit-Learn allows for the use of machine learning tools that are essential in implementing the likes of SVM and TFIDF feature extraction.

4.2 Choice of IDE

When deciding for the implementation of this project, the Jupyter IDE was chosen. Jupyter notebook is an open-source web application that builds python programs within your web browser. Its main designed use case was to help developers share their work, because of this Jupyter adopts practices that focus on making code easier to understand. The IDE implements a single cell system that makes code segmented and straightforward. These cells can be supported with additional titles and comments to make the code much more comprehensible.

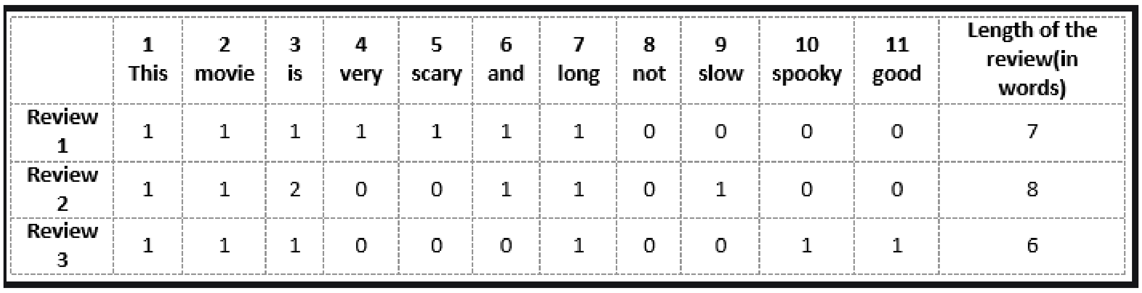
Jupyter notebook sees wide use specifically within data science due to its user interface allowing for the reading of visualisations within code rather than leaving and fetching produced visualisations outside the IDE. The cell system also presented a large benefit in its ability to allow developers to test specific segments of code, irrespective of the rest of the program. This is particularly useful in this project as it allows for the Machine learning model to be trained once before further adjustments are made to the code.

Being able to separate the runtime of the ML model from other areas of code was important. Running the ML model every time small adjustments were made to the code would increase time spent coding unnecessarily due to high run time. With Jupyter, the ML model can be trained once, and then other sections of code can be executed multiple times.

4.3 Text Feature Extraction

Computers by nature are unable to read raw text. let alone the large datasets of tweets that are to be analysed by the machine learning model. To circumvent this, machine learning practices the use of text feature extraction. Text feature extraction, involves the changing of words into numbers through vectorisation. This is important as it allows for the relaying of information to the machine learning model. Without this step, the is no way to communicate to the model the human language within the data selected.

A basic example of text feature extraction is the Count vectorizer method, also known as the Bag of Words model. A look at this allows for the demonstration of the basic feature extraction method. When a small selection of text is analysed, the Count vectorizer will begin by counting the occurrences of all unique words seen. It should be noted after NLTK Techniques like stemming and tokenisation are applied. The will be an exclusion of unimportant words like "the", "is", "an" etc. By avoiding the recording of these words, will allow for a less noisy collection of data. filled with only important words that hold meaning.



	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Figure 2: "A Typical Document Term Matrix" Huilgol (2020)

The occurrences of each unique word found across the entire document is then recorded in a Document Term Matrix (DTM) (See Figure 2). These words are then represented by their placements within individual sentences. Along with the amount of times they are recorded across the entire document as a whole. The unique vector value is then associated with that specific word. This allows for machines to identify words, but in numerical forms as vector values. Once every word has been vectorised, machines can understand every single word within the document. Yet, within text feature extraction, there are many methodologies available. Selecting the best suited for our project is important.

Count vectorizer / Bag of words is amongst the weaker methods to adopt. Semantic variation is non-existent within the BoW model. the semantic meaning of words are not taken in to account during the vectorisation of the document. This would mean words like "great" and "awful" are both given a standard vectorisation with no representation of their sentiment. The exclusion of semantic meaning would be less optimal to increase the model's ability to classify abusive tweets. Due to this, the text feature extraction technique adopted in this project was "Term Frequency - Inverse Document Frequency" (TFIDF). TFIDF is much more aware of the semantic meaning of words, in its vectorisation of the dataset.

4.3.1 Importance of TFIDF

TFIDF approaches vectorisation different from BoW. it looks to identify the relevance of a word within a document by its frequency of use across the document. The main aim is to vectorise while maintaining the sentiment of words. It does this by utilising two important variables. Term Frequency and Inverse Document frequency. Term Frequency (TF) is a value produced by following the formula.

$$TF(I,J) = \frac{\text{Term I frequency in document J}}{\text{Total words in document J}} \tag{4}$$

Term I in frequency measures how often term I (I can represent any word) appears in the document. This is then divided by the total words in the entire document as a whole. This value is significant in identifying terms that appear many times within a document. By itself the value is not useful for thorough feature analysis. Term frequency alone would highlight very common words within a document with high priority. This would mean words like "is" and "the" would counted as more important over words like red, dog, good etc. Inverse document frequency (IDF) is the 2nd value used. IDF follows the formula:

$$IDF = log_2(\frac{\text{Total documents}}{\text{Total documents with Term I}}) \tag{5}$$

This measures how often the term I is used across the documents within the larger dataset. This value allocates a value to a word based on its rarity across the document. Words close to 0 are considered as common words, words with a higher value are rarer.

The two are combined to create TFIDF. TF highlights words of common use while IDF filters out words that have little importance within the document. Words with a high TFIDF score represent relevance within the document.

$$TF * IDF \tag{6}$$

In practice, Terms like "bad" may generate a TF-IDF score of 0.6, it is a relatively commonly used word to label something with a negative sentiment. Yet words like "Disgusting" would score higher. This is because it is used less often within the entire document. Increasing its IDF signifying its of higher relevance. As a result, "Disgusting" would score a higher TFIDF than "bad". Its rarer use emphasises that it most likely holds higher importance in the realm of negative sentiment. TFIDF's ability to retrieve sentiment within a document, give machine learning models an extra layer of information and sentiment analysis used in its training. Thus TFIDF is the chosen feature extraction adopted for this projects Machine learning model.

4.4 Lexicon Design

The lexicon methodology is widely used in sentiment analysis. Lexicon by definition is simply just a vocabulary. Lexicons consist of a large corpus of words and assigning them values of polarity and general sentiment. A program is then created in which, a document is read word by word. each word examined is then checked to see if it exists in the used lexicon. When words are seen to appear in the lexicon, their respective lexicon score is accounted for, once the entire document has been examined alongside the lexicon, a score will be tallied that should be an indication of its overall sentiment.

Lexicon methods overall provide consistent results and generate a good evaluation of the sentiment of a document. They can be domain-specific and provide a robust approach. They can be updated regularly to increase their effectiveness over time. Yet lexicons can suffer from context and sarcasm heavily. The use case of words within sentiment can be overlooked, meaning "negative" words may be used in a neutral or positive tone, but would still be highlighted as negative sentiment. This also applies vice versa.

4.4.1 TFIDF Lexicon Concept

TFIDF is often used on a document with the intent to recover a certain amount of words that hold high relevance within the document. This practice, referred to as keyword extraction is a significant use case of TFIDF. This paper explores the use of this keyword recovery, to create a potential lexicon to be utilised in abuse detection as well. The idea presented in this paper is to use TFIDF keyword recovery on multiple hate speech datasets. This is to recover individual words that are seen to emphasise general abuse. With each dataset targeting a different approach to abuse. The words are then accumulated to a larger list that is then used as a Lexicon to detect abusive tweets. This concept looks to combine lexicon with a supervised learning approach, if this concept were to work it would greatly aid in the creation of lexicons across any use case.

4.4.2 Lexicon Dataset Selection

This section looks to shed light on the active design choice taken when considering selecting the datasets used in TFIDF Lexicons application.

The use of 3 datasets in TFIDF Lexicon, is to ensure a wide range of abuse is covered within its built-in lexicon. The 3 datasets are specifically chosen to attempt to cover a different class/curriculum of abusive data. Dataset 1, composed of a set of annotated comments from Fox news' website was retrieved by Gao & Huang (2017). It is accessible through their GitHub page going into detail on their machine learning model. In their project, they look to focus on context-aware machine learning models. Their dataset is specifically chosen to be context-rich to train their model. Due to this their labelling of data is chosen to retain as much contextual information about the comments chosen. The use of this dataset is to help the TFIDF key feature recovery, target terms from the context rich classification of comments, in doing so help increase the lexicons robustness to context.

The 2nd dataset is found on Kaggle and referred to as the Sentiment140 dataset. This set of data originally contains 1.6 million tweets, with a 50/50 split on pure sentiment. This paper is produced by the works of Go et al. (2009), their main method of classification is based on the use of emoticons in the tweets they recovered through the use of an API. Go et al. (2009) gave general positive and negative sentiment to the tweets indifferent to the existence of abusive content. But instead focusing on the emoticons recorded in the tweet dataset Using this dataset gives the Lexicon a better understanding of general sentiment. It serves two significant purposes. One is to make the lexicon more aware of words that are used in conjunction with emoticons to specify sentiment. This makes it much more aware of emoticon heavy tweets. Additionally, this dataset is best suited out of the three in highlighting non-abusive terms and words. This dataset contains tweets of positive sentiment, something which is scarce in the other two datasets. Which Mainly highlight neutral and abusive data. This additional data for positive sentiment will help in the lexicons ability to differentiate from abusive tweets. Feeding the lexicon more positive data will help the classification of future tweets. The positive sentiment expressed in these tweets will give the lexicon a better understanding of non abusive data. Helping its general ability to differentiate better This dataset contained 1.6 million tweets. which was extremely large causing excessive computing time. To counter this, the dataset was sampled of 40,000 tweets. 20,000 tweets of positive and negative sentiment were randomly separated into a CSV file, which was then chosen to feed to the Lexicon.

The 3rd dataset used is available on Kaggle. It contains tweets that have been split into subsections, and then distinguished by manual labelling. The manual labelling, was addressed by users of Crowd Flower. A website used to source large groups of people to perform tasks not yet achievable through AI. The subsections used are: "hate speech" "offensive language" "neither" "class" and "count". Once several tweets were selected, Crowd Flower users were asked to categorise the tweets. The "count" variable of these tweets represented the total score across the 3. The dataset originally comprised of 24,783 values was sampled to just 900 tweets to reduce computational time. Yet, TFIDF is still able to generate the most impactful keywords in the entire document by following the sampling method. The sampled dataset was selected based on the highest "count" values and the highest "Neither" values. "Count" within this dataset represents the total points gathered for a tweet, which fell under hate speech and offensive language. The highest "count" values would hold the most negative sentiment within the dataset. The highest 450 "count" scoring tweets were allocated to the sampled dataset. "Neither" in this dataset meant data which was not considered either abusive or containing offensive language. A high "neither" score meant tweets that were not abusive in nature. The highest 450 "neither" tweets were also allocated to the sampled dataset, with a non abusive score. By selecting the extremes of both ends, the tweets highlighted were abuse and non-abusive. While also being the highest scoring in the dataset for both groups. TFIDF would still provide rich keywords from this smaller sample.

4.4.3 TFIDF Lexicon Flowchart

This Flowchart helps represent how the final TFIDF Lexicon should be created

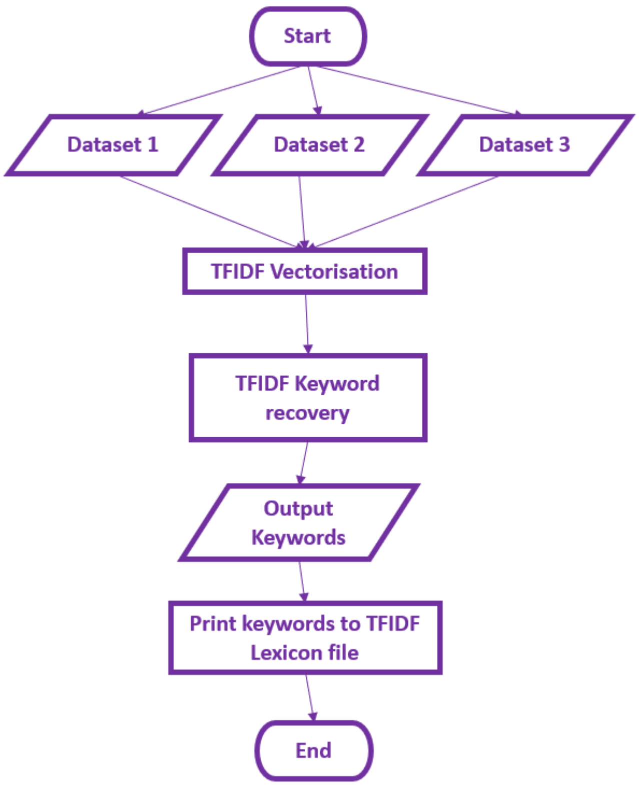


Figure 3: Creating TFIDF Lexicon

4.4.4 AFINN Lexicon

To measure the success of the TFIDF lexicon proposal, an additional lexicon method was adopted to be able to compare the success of the TFIDF model with an "industry standard" lexicon. The lexicon chosen for this comparison was the AFINN lexicon. AFINN lexicon is widely used in other lexicon projects and utilises a general sentiment analysis outlook as opposed to one specific for hate speech. The TFIDF lexicon is used with a score of 1 to 0 strictly due to the datasets used, while the AFINN lexicon has a wider range for recorded polarity. The lexicon AFINN lexicon scores were restricted to 0 and 1 to generate comparable results with the TFIDF. Using the AFINN also allowed for a recording of overall sentiment score on tweets fetched by the API, this is through its inbuilt polarity scoring.

4.5 SVM Design

4.5.1 What is SVM

Support vector machine (SVM) is a well-respected machine learning method that has seen numerous adaptions in a wide range of applications. it is a powerful machine learning tool that utilises supervised learning to create a model that can perform both classification and regression tasks. How this project looks to tackle abuse, is through the classification of abusive content. Highlighting abuse as 1 and non-abusive as 0. SVM's binary classification of data is well suited for this task. This makes this a classification task of two features.

Pupale (2019) Defines hyper-planes as "hyper-plane in an n-dimensional Euclidean space is a flat, n-1 dimensional subset of that space that divides the space into two disconnected parts." . Hyper-planes represent decision boundaries used in classifying data, they allow for a spread of data to be objectively separated. Within our project, there are two distinct features, whether a tweet is abusive or not. This means that the hyper-plane created is of one dimension.

Within the spread of data vectors, multiple hyper-planes can be created. Many of these hyper-planes would be relatively successful in creating a clear separation of the data classes. However, finding the most optimal hyper-plane is the goal of SVM. SVM takes a step further than Logistic regression in this aspect as it is not only looking for an optimal hyper-plane like Logistic regression. But it is also looking for the best marginal classifier in addition. This extra step allows for an additional measure in attempting to split the data.

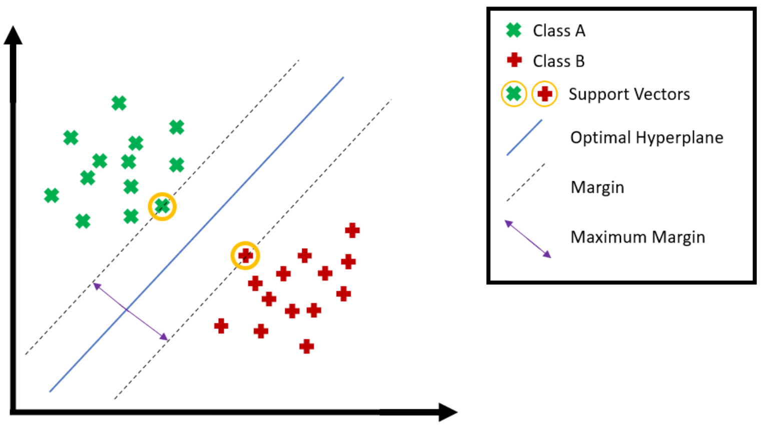


Figure 4: General SVM Graph

Support vector machine aims to create separation between vectors of different classifications. With the use of a hyper-plane and marginal classifiers. Marginal classifiers play an important role, they act similar to an additional cushion for classification, finding an optimal hyper-plane also means finding an optimal margin in SVM. By creating both hyper-plane and margin. The SVM model can successfully classify data (represented as vectors) based on where the data falls, using the decision boundary as a guide on classification. When unseen data is introduced to the SVM, it can classify new data based on which side of the hyper-plane it falls in.

Support vectors are the specific vectors that are used by the model to find the correct marginal classifier. They also help select the most optimal hyper-plane. How these support vectors are determined is the nature of SVM's mathematical intuition.

4.6 SVM Further Maths Intuition

Hyper-planes within a 2 dimensional set of data are represented by a singular line. The formula for a hyper-plane is $y = w^t x + b$ (w = normal to the line, b = bias) , additionally both margins will hold formulas of $y = w^t x + b = + - 1$. These formulae create the hypothesis formula:

$$h(x_i) = \begin{Bmatrix} +1 & \text{if } w \cdot x + b \geq 0 \\ -1 & \text{if } w \cdot x + b < 0 \end{Bmatrix} \tag{7}$$

This forms the basis of the hypothesis function used in SVM. This formula dictates any values above or below the hyper-plane will be classed as +1 or -1 respectively. Creating the instance of classification for data. These two equations are then used to find the distance within the margin.

The next step is to identify the distance between both classifiers, by taking the x values of the support vectors of distinct classification. SVM creates distance by finding the difference between both support vectors. By using the hypothesis formula and substituting $X =_2$ and X_1 and subtracting. This creates the formula $w^t(x_2 - x_1) = 2$. This formula is then further simplified to create the formula: $\frac{w^t}{\|w\|}(x_2 - x_1) = \frac{2}{\|w\|}$ In this formula, the final sum grants the SVM its' optimisation function. SVM aims to maximise this value $\frac{2}{\|w\|}$. This must be maximised pertaining that the constraint condition is met. This creates the equation

$y_i(w \cdot x_i - b) \geq 1$

Finding the w and b value that grants the most maximal $\frac{2}{\|w\|}$ while remaining within the conditions set in $y_i(w \cdot x_i - b) \geq 1$. The SVM problem has been solved. The support vector machines are the two values that fulfil this intuition the best out of the data presented.

4.6.1 Why SVM?

The SVM model is seen to be used in multiple use cases, it can be utilised in both classification and regression tasks, this makes it commonly used within the community. Increasing the overall potential to access support and examples online. The mathematical intuition of SVM also make it a robust model which is beneficial when dealing with less effective data. SVM also provides future proof capabilities as it is able to adapt to non linear data through the use of its unique kernel trick, this can be useful when datasets containing much larger features are taken in to consideration.

4.6.2 TFIDF SVM Synergy

When data is vectorised by text feature extraction methods, they are given a vector value as stated previously. The vector values are all unique by nature. But data points, of the same label, will naturally assimilate in similar areas. TFIDF will have given the vector values higher semantic variation amongst their respected classes. This means that more abusive data will have a higher negative score and place further along within the measured one dimension. Making it easier to classify as it is further away from the hyper-plane. This helps to demonstrate the benefit of adopting TFIDF within the proposed SVM model.(See Figure 5).

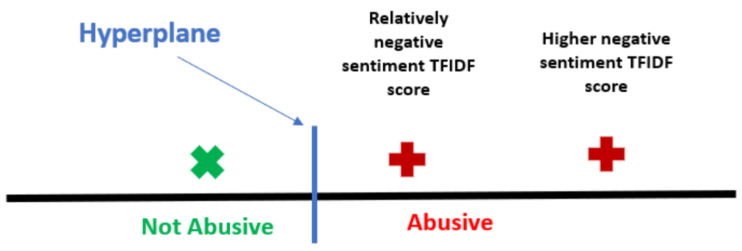

Hyperplane
Relatively negative sentiment TFIDF score
Higher negative sentiment TFIDF score
Not Abusive
Abusive

Figure 5: TFIDF aiding classification

4.7 SVM Dataset Decision

The dataset sourced for the SVM is found on Kaggle. The dataset was produced as a part of a competition to produce a viable toxic comment classification dataset. This dataset was originally formed with multiple classes but was transformed for binary classification of abusive content. Due to its original multi-class nature, the dataset is composed of online comments that were highlighted with threats, insults and identity-based hate and other forms of abuse. Giving it a broad range of toxicity which is better suited for the ”general abuse” path followed in this project.

Being comment based allows for a set of data that is in the first person and similar to the format that would be found in Twitter, making it suitable for this project. In its entirety, the dataset contains roughly 225000 unique values. With 21000 comments of abusive sentiment. This is then split into a training and test set of CSV files. The training and test dataset are split with 160,000 and 60,000 unique values respectively. Initially, the training dataset contains 160,000 comments with around 15,000 being toxic comments. This low ratio of abusive comments resulted in low recall as seen in the initial prototype results:

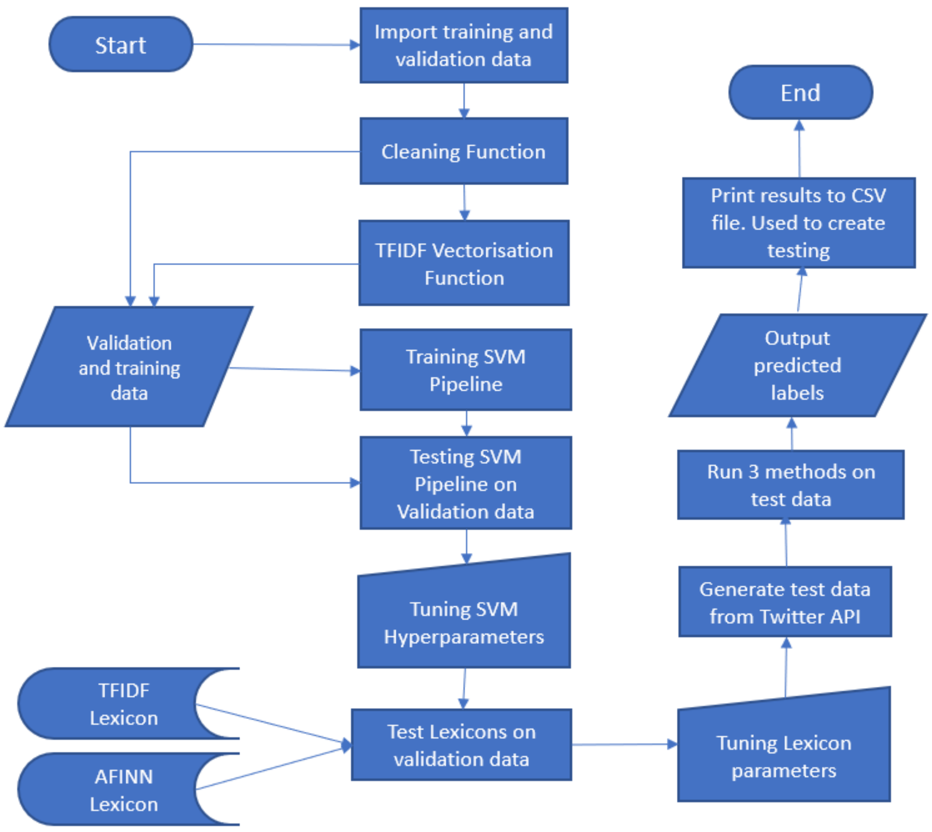
SVM Prototype	Initial Results
Accuracy Score	95.63
Precision recall	84.97
Recall Score	46.07
F1-Score	59.75

The low recall was very telling of the datasets disproportionate amount of negative comments. With low recall, it was evident the 15000 negative comments were not large enough to correctly train the model for abusive comments. To counter this the datasets were rearranged to include much less neutral comments. Both datasets was scaled down to 65,000 unique values across both. Training and testing were now 43,000 and 22000 comments respectively, with a 33 percent share of abusive comments amongst both sets. This adaption helped to increase the SVM's ability to read negative comments a lot better, resulting in higher recall scores within the final product. This use of specific sampling puts the model at risk of missing out on vital information within the deleted data. However this rearrangement proved vital in increasing the overall success of the SVM model. While also reducing excessive computational times examined in initial prototype.

The splitting of the dataset into separate training and test files was a crucial step. This test set of data was to be used as validation testing, which was defined in the requirements section.

4.7.1 Total System Flow chart

This flow chart helps to portray how the final product should function.



Start
Import training and validation data
Cleaning Function
TFIDF Vectorisation Function
Validation and training data
Training SVM Pipeline
Testing SVM Pipeline on Validation data
Tuning SVM Hyperparameters
TFIDF Lexicon
AFINN Lexicon
Test Lexicons on validation data
Tuning Lexicon parameters
Generate test data from Twitter API
Run 3 methods on test data
Output predicted labels
Print results to CSV file. Used to create testing
End

Figure 6: Final Design Flowchart

5 Implementation

5.1 Data Pre-Processing

The pre-processing of raw data is an important tool that is required for the mining of data and any later patterns. It allows for the transformation of raw data text, into a comprehensible arrangement for NLP techniques . Real-world text often contains many discrepancies that make it difficult to read: misspellings, inconsistencies, lack of grammar etc. Furthermore the is an extra layer of preparation required to process tweets. The removal of hyperlinks and Twitter keywords like " Retweet ". This general pre-processing of data in text classification is referred to as cleaning.

Cleaning text requires the use of NLTK libraries mentioned earlier. The use of these libraries allows access to necessary techniques like lemmatisation, tokenisation and using stop words. Tokenisation is a technique used to break larger streams and passages of text into singular tokens to help create a much more coherent list of words instead. Lemmatization looks to shorten original words however unlike stemming, lemmatization looks beyond just word reduction. It makes sure to take a morphological analysis approach to words, attempting to keep words' original meaning. Making lemmatization a more favourable path than stemming. The pre-processing of data also includes the use of NLTK's stop word corpus with around 300 words. These are words that NLTK have highlighted as frequently occurring words that don't provide additional information to the text. These are words like "a" and "the". Removing these words will help reduce unnecessary computational time as well as increase NLP's overall processing of relevant text.

These techniques were used to create a cleaning function which was used through out the program. It was used to clean all tweets examined in all datasets aswell as the Twitter API. The cleaned data would go on to be processed by both lexicon and machine learning methods. Data pre-processing helped improve the overall results of all classification techniques used. This is an important aspect of the methodology adopted in this project.

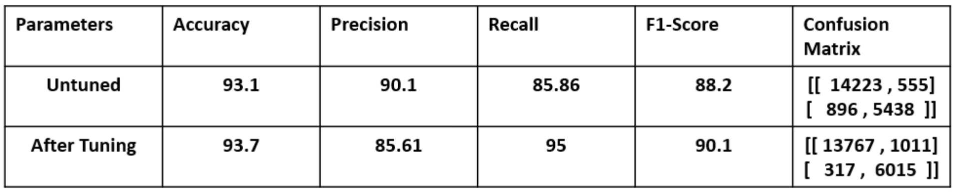
5.2 SVM Application

After training data has been successfully vectorised through TFIDF aswell as pre-processed by the cleaning function. The processed set of data is ready to be used by the SVM model. Implementing an SVM classifier incorporates using the Sci-kit learn library and utilising their SVM pipeline. The pipeline is initialised without tuned hyperparameters, with Sci-Kit learn incorporating general parameters for broad use. The classifier is introduced to the newly vectorised training data, the model will also be fed the validation data. After training the model, it is tested on the validation data. Generating machine learning metrics for analysis. These results are then examined and used to determine how hyperparameters can increase these values.

5.2.1 Hyperparameter Tuning

Hyperparameter tuning is the process of adjusting the pipeline defined in the SVM to attempt to create better results. It is often used in machine learning to adjust parameters that contain many elements of machine learning models that cannot be directly learned and rather human intuition is needed. By acknowledging the data and features, then selecting parameter values that will be best suited. The test split dataset was used as validation to evaluate the success of the adjustments made.

After examining the results of the base SVM Pipeline the following parameters were chosen to be changed. The kernel parameter selected was linear this is due to the nature of this project containing only a few features. The data can thus be separated linearly and doesn't require the use of a polynomial kernel used in non-linear data. Class weight was changed to "balanced" this automatically adjusted weights to create a more optimal model. C represented the regularisation parameter. This parameter dictates how strict the model is in allowing for misclassifications it was seen that setting this value to 2 provided the best results. Random_state serves just as a value to



Parameters	Accuracy	Precision	Recall	F1-Score	Confusion Matrix
Untuned	93.1	90.1	85.86	88.2	[[14223 , 555] [896 , 5438]]
After Tuning	93.7	85.61	95	90.1	[[13767 , 1011] [317 , 6015]]

Figure 7: Hyperparameter Tuning results

randomise the data before processing. This final set of parameters was chosen to be used in the concluding classifier.

After hyperparameter tuning, it is evident the was an increase in overall performance as seen in the figure above. The confusion matrix difference helps to portray that the model has become better at correctly predicting abusive tweets, this is inferred by the increase of the true negative value. This in conjunction with the increase of F1 Score shows overall Precision and recall values have improved. Making the final model improved from its original raw parameters.

5.3 Application of TFIDF Lexicon

The three datasets are loaded in and processed to all share the same format with strictly tweet and label. An empty array is made for "positive" and "negative" words. These are the arrays to be allocated the highest-scoring TFIDF words across the three documents. The data is then cleaned with a cleaning function containing tokenisation and stop words, the same function used for the SVM. This is to ensure a fairer comparison between the two methods.

A secondary function is used to perform TFIDF extraction. The function looks to select a X amount of words with the highest-scoring TFIDF value per dataset. (X representing the *number* of words extracted). Amongst both abusive and non abusive classified data, the highest TFIDF scoring words are selected. These are the terms added to the lexicon. The top X words of abusive labels are added to the negative list and the top X words of non-abusive labels added to the positive list. Deciding X as the number of words to extract was one of the values that were monitored in accordance with validation testing.

Once this function is executed across all three of the datasets, the positive and negative list will consist of the top words from the 3 datasets. The positive and negative lists are combined while keeping their classification value of 1 and 0. The lexicon is then finalised, it contains the positive and negative words extracted by TFIDF. Their lexicon sentiment value is assigned based on their classification, with negative words being used as +1 and positive words being used as -1.

When tweets are classified by the lexicon, the tweet is tokenised, then measured against the lexicon. The score is then created by counting the lexicon values registered in the tweet. The value is then localised to 0 and 1. 0 being a tweet classified as non-abusive and 1 being classified as abusive.

5.3.1 Lexicon Validation Testing

X was one of the adjustable values that were altered and monitored to see how it affected the lexicon's validation testing results. Depending on how many X words are extracted influences the semantic dilution of the lexicon. With a lower X value meaning only words of higher TFIDF are in the lexicon. Whereas a higher X value may dilute the overall lexicon as it will incorporate lower-scoring TFIDF words that might not hold as much relevance in their respective classifications. This value will also affect the total size of the lexicon.

The validation dataset used was the test split of the original SVM dataset, this was a portion of unseen data that has set labels. Containing 6332 labels of abuse and 14778 labels of non-abusive data. The lexicon is executed on this set of data and seen how similar its classifications are to the real values. This method of testing is a rough estimate of the lexicons success and doesn't provide as much depth as machine learning metrics. Yet it gives a good estimate on how adjustments made, contribute to the lexicon's overall classification.

X (Number of terms)	Abusive Labels	Non-Abusive Labels
250	7923	13167
200	6565	14545
150	5972	15138
100	5666	15444

Originally X the number of words extracted was set to 250, making the lexicon composed of 750 words of abusive and 750 words of non-abusive terms. The lexicon produced classifications of 7923 abusive and 13187 non-abusive against the validation data. This showed that the lexicon worked in classifying abusive and non-abusive tweets to some extent, but, it was grossly overestimating the true value of abusive labels.

The x value was then adjusted multiple times. This was to see how it affected the lexicons classification of the validation data. The results then showed an X value of 200, provided 6565 abusive and 14545 non-abusive labels, this displays a much closer approximation to the real values. This helps determine that the adjusting of the x value has increased the accuracy of the lexicon in detecting abuse.

This change particularly showed it became much better at detecting non-abusive tweets as that classification increased and abusive classification decreased. The decreasing of the lexicon's size most likely cut out terms that didn't provide much abusive sentiment. Cutting out these terms would have allowed the lexicon to retain purer sentiment/abuse classifying terms. The final X value used before final evaluation was 200.

However despite using the validation dataset. It is evident that this form of testing for validation does not give much insight in to the actual success of the lexicon's ability to differentiate between labels. It could easily be most of the labels are miss classifications despite the being such a close resemblance to the actual values of labels. This highlights one of the major issues of the lexicon method, it is much more difficult than ML models to gain an initial measure of success.

5.4 Twitter API

After importing Twitter library "tweepy" and entering in respective API keys. The API looks to fetch tweets in two methods, one randomly with the use of a "timeline" variable and a second method through a user-chosen query. Users can select the number of tweets generated additionally. This customizability is important to allow for a more flexible selection of tweets. Allowing the program to generate tweets of a broader range to creating a testing range to in turn test the flexibility of the models.

Once these tweets are fetched they have placed into a separate data file. This data file is then cleaned with data pre-processing and then evaluated by the 3 methods generating a score. These scores are then printed to a CSV file for further examination and use in the test survey.

6 Survey Testing

6.1 Experiments

After completing hyperparameter tuning and lexicon adjustment through validation testing. Both models were now complete to progress on to final testing. Final testing was to take place using a survey to collect human processed data labels on new tweets. This is an important aspect in garnering an overall evaluation of the methods. This is particularly helpful, in providing a measure of success for the two lexicon methods.

The Twitter API allows for many ways in which to fetch tweets. The main method in gathering data for this survey was to use the Tweepy librarys' query-based search. To generate the data the query variable would be chosen from a selection of random words. This allowed for the retrieval of random tweets. The survey examination of random tweets provides valuable information in one dimension. This is because it shows the models' success rate in detecting abuse within normal tweets.

But to further test the models, a selection of more abusive tweets were also examined. These tweets were generated using stronger abuse themed query searches. This ventured into how good were the individual models at demonstrating disparity in abusive tweets. This set of classifications would aim to highlight which models were more successful in their definition of abuse. These selections of tweets would also be used in displaying results to portray the success of the three models. The two sets of survey tweets would also be combined to give an overall evaluation of the models' success.

When generating tweets to display to participants. There were a couple of criteria, followed to allow for a fairer testing environment. Tweets that were fetched incorporating the use of images and videos. Were not used in the test sample as this was unfair to the models. The models did not initially include these features in their training. Thus making their classifications on tweets with media, obsolete. As the raw text retrieved from these tweets were most likely to be heavily out of context. Only tweets with strictly text were chosen for the survey test sample.

Participants were simply asked to rate tweets they felt to be abusive as 1 and tweets they felt non-abusive as 0. A definition of abuse was not prescribed as this survey was also to explore the different levels of sensitivity seen in evaluating the models. Naturally, participants were not able to see the labels made by either models or other participants. when deciding on their classifications. This was to maintain credibility and refrain from labelling bias. Participants were selected without disclosing any personal information. This was to stand within ethical guidelines without the use of a proper ethical form. No personal data would be risked.

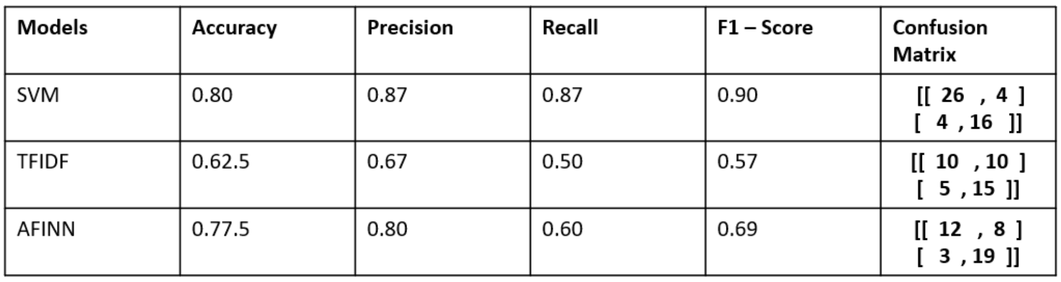
6.2 Data processing

After the survey is completed, the 5 participants have read the selection of 40 tweets, and have created their classifications. To culminate their data fairly, an average was taken across their results. The average chosen was the mode value as it was the most suitable when dealing with this type of data and the low sample of participants. This produced 40 "true" labels that can be used to create metrics for the 3 methods compared. Using the true labels and comparing them with the original predictions, a confusion matrix was made, and overall accuracy was also calculated. This created additional data for the lexicons that were not accessible before, simply through the validation data. While also giving an insight into how accurate the SVM was outside of its own original dataset.

6.3 Results and Discussion

6.3.1 Results

The results extracted from the survey help to portray the reality of the 3 methods. While also allowing for additional analysis on the lexicon. Once converting the results into the classification metrics here were the results recovered.



Models	Accuracy	Precision	Recall	F1 – Score	Confusion Matrix
SVM	0.80	0.87	0.87	0.90	[[26 , 4] [4 , 16]]
TFIDF	0.62.5	0.67	0.50	0.57	[[10 , 10] [5 , 15]]
AFINN	0.77.5	0.80	0.60	0.69	[[12 , 8] [3 , 19]]

Figure 8: Survey results

The disparity within the accuracy results helps to create an early understanding of the methods' ability to detect abuse. SVM is seen to have the highest accuracy with 80% score with AFINN following behind with 77.5%. While TFIDF Lexicon shows an accuracy of 63%. Accuracy alone would portray the image that the SVM and the AFINN are both performing well with TFIDF being the clear worst. However, when exploring the other metrics further it is seen that the AFINN lexicon is not nearly as successful as the SVM model.

The confusion matrix results help to illustrate further issues seen in the lexicon models. Their overall classifications are often littered with misclassifications. The confusions matrix for both lexicons shows a high "False Negative" value which is a major cause for concern as it interprets that they are classifying non-abusive tweets as abusive, and doing so regularly. In a real-world application, this would be extremely severe as it could lead to bannings and reportings of users wrongfully which would prove unsuccessful. Whereas the SVM is seen to have a low number of both false negative and false positive

The F1 Scores examined further emphasise the difference in misclassifications, highlighting TFIDF Lexicon to be the worst with Afinn 2nd. Both are seen to score considerably lower than the likes of the adopted machine learning model.

By interpreting these results it is clear that the lexicon methods have performed worse, and the SVM model has performed considerably better in all aspects.

6.3.2 Discussion

One of the reasons why the SVM Model has performed considerably better than the lexicons is down to the SVM Model being better suited to deal with the issues of context. With the sarcastic nature of Twitter, it can be seen some tweets may contain abusive words/ content but not be used in an abusive manner. The lexicons may have highlighted words as "abusive" and without the understanding of context, "abusive" words will heavily enforce the lexicon to classify tweets as abusive without understanding the rest of the sentence. This is evident by their low F1 Score meaning a significant number of non-abusive tweets, were classified as abusive.

However, to say the SVM method is conclusive in solving all abuse detection would be genuinely preposterous. The SVM method showed an inaccuracy of 20% which would be considered high in a real-world application. Although the process of data sampling that was adopted earlier on within methodology has made the SVM better suited at detecting abuse. It can be seen that the cutting down of overall data has restricted its ability in learning to differentiate non-abusive tweets. With some concerning classifications having been examined in the data surveyed.

The most significant reason why this SVM model still has issues is most likely due to the limited availability of a well-defined dataset. The dataset used was initially considerably imbalanced, with a 15 % split of abusive to non-abusive data. Furthermore, it was already of relatively small size containing just 160,000 tweets. The sampling method used on this dataset aided in clearing some of the faults, however, is not a concrete replacement for the importance of a well-made dataset. Sampling also reduced the general volume of data drastically, reducing the model's overall learning.

Essentially an ideal dataset would include considerably larger amounts of data to maximise exposure of the SVM model to Twitter data. While feeding it a balanced set of abusive and non-abusive data. In addition to this, an ideal dataset would also include a broader categorisation of abuse. From the survey, it was evident that users will have different levels of tolerance on what they would consider abuse, to solve this a dataset would need to integrate manual labelling. Manual labelling would allow for a broader and thorough definition of abusive content. Reducing the blurry line of what is considered abuse. Adopting such a dataset would likely solve many of the issues seen with the SVM model within the survey testing.

The mathematical fundamental of SVM provides a strong base to create detection, but without a significantly profound dataset, any instance of a machine learning model will struggle to provide a real-world use case against abuse detection.

With the AFINN lexicon, the results displayed are understandable as AFINN aims to detect negative or positive sentiment, making it much more general. This means it is not suited to differentiate between Abusive content and general negative sentiment. It has served as a good indicator of the issues of the TFIDF lexicon. It helps to portray that the TFIDF lexicon is failing at more than just standard lexicon faults like understanding of the context. Highlighting there are greater issues to understand for the proposed lexicon.

The TFIDF lexicon has performed poorly overall due to the datasets used and the fundamental of TFIDF extraction. The lexicon is seen to recover numerous words that can be argued as not being related to abuse. This is most likely because the datasets used were made to suit differing ranges of abuse, to help machine learning models differentiate better between non-abusive and abusive. This creates an issue in TFIDF extraction as it clouds the extraction method with many words of lower importance that are still relevant in distinction, but only at a deeper fundamental level applicable only in machine learning. Yet when used in a lexicon, these words do not hold a high enough semantic meaning to justify being used in a score metric designed to differentiate solely off of words alone.

The issue of dataset and semantic keyword purity was originally considered in the TFIDF lexicon concept. Hence the paper adopted three datasets for the lexicon. In order to extract a wide range of abusive terms for the lexicon hoping to minimise the issue of noisy data being integrated into the lexicon. Yet in retrospect with further testing data now acquired it can be seen that an extra fundamental that should have been considered was the implementation of polarity scores. Utilising polarity scores would have greatly helped overcome the issue of noisy data, by providing semantic variation. Attributing polarity scores is difficult to perform successfully using a supervised learning approach.

Ultimately attempting to create a lexicon with supervised learning proves difficult, with too much responsibility being placed on the success of TFIDF extraction. Furthermore, if this TFIDF lexicon concept was to be explored further, 3 major factors would need to be adopted. Creating a dataset specifically for the model and the assigning of polarity scores. Additionally in future works adopting text feature extraction methods like word2vec can also help to further the creation of a supervised learning lexicon. Word2vec is much better than TFIDF in retrieving semantic meaning.

Overall the survey results combined with the validation testing show that the program can detect abusive content and retrieve tweets through the API. The 3 models have shown different levels of success

7 Conclusion

Detecting abuse on Twitter is an astronomical task, being continuously explored by large cooperations and academics alike. With the likes of Twitter being pressured through social media blackouts and online outrage. Due to their poor solutions to this issue. This project helps highlight some of the difficulties that plague developing a solution. While comparing key automated methods put forward to try to tackle abuse detection.

This paper demonstrates a comparison of abuse detection methods. Using machine learning and lexicon methods the paper can demonstrate and portray a fundamental understanding of these two types of methods. Discussing their benefits and drawbacks while showing examples of their success through an anonymous survey.

The proposed TFIDF-SVM model is thoroughly tested and shown to be successful to a good extent but highlights the importance of issues like dataset availability and suffers from extensive run time. While the experimental lexicon created through a supervised learning approach showed much less success. Highlighting issues with the lexicon concept, relying too much on manual development, examples such as evaluating their performance. By using two different types of testing the project can discuss the extent of the success of both methods, evaluating overall performance. While providing insight into how to improve both methods in future work

The TFIDF lexicon worked to some extent and was able to detect tweets with abusive content. But ultimately would often cause miss classifications, a second look at the dictionary values showed a lot of words of non-abusive nature. While the AFINN Lexicon was relatively accurate in detecting abuse it also classified many negative sentiment tweets as abusive. Overall the only real benefit exhibited by these lexicon approaches were the much quicker execution times. Lexicon's inability to read sarcasm and context is not suited well for Twitter, where "abusive/negative" words are often used in a sarcastic nature.

Adopting a machine learning model would be best suited to tackle the issue of abuse, with enough training a model can be taught how to differentiate between abusive and non-abusive labels regardless of problems like sarcasm and context. This could be proven further by adopting better datasets that integrate multiple types of abuse and containing A higher volume of training data. Doing so would help solidify the notion that machine learning is better suited than lexicon methods to tackle abuse detection. Models like SVM can become robust solutions to censoring Twitter.

References

Aung, K. Z. & Myo, N. N. (2017), Sentiment analysis of students' comment using lexicon based approach, *in* '2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)', pp. 149–154.

Bouazizi, M. & Ohtsuki, T. (2015), Opinion mining in twitter how to make use of sarcasm to enhance sentiment analysis, *in* 'Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015', pp. 1594–1597.

Bradley, M. M. & Lang, P. J. (1999), Affective norms for english words (anew): Instruction manual and affective ratings, Technical report, Technical report C-1, the center for research in psychophysiology . . . .

Bross, J. & Ehrig, H. (2010), Generating a context-aware sentiment lexicon for aspect-based product review mining, *in* '2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology', Vol. 1, IEEE, pp. 435–439.

Chakrabarty, T., Gupta, K. & Muresan, S. (2019), Pay "attention" to your context when classifying abusive language, *in* 'Proceedings of the Third Workshop on Abusive Language Online', pp. 70–79.

Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G. & Vakali, A. (2017), Mean birds: Detecting aggression and bullying on twitter, *in* 'Proceedings of the 2017 ACM on web science conference', pp. 13–22.

Colas, F. & Brazdil, P. (2006), Comparison of svm and some older classification algorithms in text classification tasks, *in* 'IFIP International Conference on Artificial Intelligence in Theory and Practice', Springer, pp. 169–178.

Dean, B. (2021), 'How many people use twitter in 2021? [new twitter stats]'.
URL: *https://backlinko.com/twitter-users*

Gao, L. & Huang, R. (2017), 'Detecting online hate speech using context aware models', *arXiv preprint arXiv:1710.07395* .

Gasca, D. (2019), 'A healthier twitter: Progress and more to do'.
URL: $https://blog.twitter.com/en_us/topics/company/2019/health-update.html$

Go, A., Bhayani, R. & Huang, L. (2009), 'Twitter sentiment classification using distant supervision', *CS224N project report, Stanford* **1**(12), 2009.

Hern, A. (2015), 'Twitter ceo: We suck at dealing with trolls and abuse', *The Guardian* **5**.

Hogenboom, A., Bal, D., Frasincar, F., Bal, M., de Jong, F. & Kaymak, U. (2013), Exploiting emoticons in sentiment analysis, *in* 'Proceedings of the 28th annual ACM symposium on applied computing', pp. 703–710.

Huilgol, P. (2020), 'Bow model and tf-idf for creating feature from text'.
URL: *https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/*

Jing, K. & Xu, J. (2019), 'A survey on neural network language models', *arXiv preprint arXiv:1906.03591* .

Liu, X., Burns, A. C. & Hou, Y. (2017), 'An investigation of brand-related user-generated content on twitter', *Journal of Advertising* **46**(2), 236–247.

Loper, E. & Bird, S. (2002), 'Nltk: The natural language toolkit', *arXiv preprint cs/0205028* .

Nielsen, F. Å. (2011), 'A new anew: Evaluation of a word list for sentiment analysis in microblogs', *arXiv preprint arXiv:1103.2903* .

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Pereira-Kohatsu, J. C., Quijano-Sánchez, L., Liberatore, F. & Camacho-Collados, M. (2019), 'Detecting and monitoring hate speech in twitter', *Sensors* **19**(21), 4654.

Pupale, R. (2019), 'Support vector machines(svm) - an overview'.
URL: *https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989: :text=do%20this%20job.-,HYPERPLANE,space%20into%20two%20disconnected%20parts.*

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N. & Huang, R. (2013), Sarcasm as contrast between a positive sentiment and negative situation, *in* 'Proceedings of the 2013 conference on empirical methods in natural language processing', pp. 704–714.

Ripley, B. D. (2007), *Pattern recognition and neural networks*, Cambridge university press.

Sanchez, H. & Kumar, S. (2011), 'Twitter bullying detection', *ser. NSDI* **12**(2011), 15.

Severyn, A. & Moschitti, A. (2015), Twitter sentiment analysis with deep convolutional neural networks, *in* 'Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval', pp. 959–962.

Smith, K. (2021), '60 incredible and interesting twitter stats and statistics'.
URL: *https://bit.ly/3wS28OZ*

Swanson, B. (2021), 'Vinai venkatesham: Arsenal ceo says 'nothing off table' over social media boycott'.
URL: *https://www.skysports.com/football/news/11670/12262836/vinai-venkatesham-arsenal-ceo-says-nothing-off-table-over-social-media-boycott*

Yamamoto, Y., Kumamoto, T. & Nadamoto, A. (2014), Role of emoticons for multidimensional sentiment analysis of twitter, *in* 'Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services', pp. 107–115.