**Investigation in using machine learning algorithms to detect if a tweet has a toxic language.**

*Author Saad Iftikhar Project Supervisor Dr Mark Elshaw*

Can machine learning algorithms be used on Twitter to detect if a tweet is toxic?



School of Computing, Engineering and Mathematics,

Coventry University

Computer Science BSc

1st May 2021

**Statement of Originality 6001CEM Declaration of originality**

***I Declare that This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.***

**Statement of copyright**

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialize products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information, please see www.coventry.ac.uk/ipr or contact ipr@coventry.ac.uk.

**Statement of ethical engagement**

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed: Saad Iftikhar Date: 31/12/2021

|  |
| --- |
| First Name:Saad |
| Last Name: Iftikhar |
| Student ID number: 9789180 |
| Ethics Application Number: P130863 |
| 1st Supervisor Name: Mark Elshaw |
| 2nd Supervisor Name |

Table of Content

[INDEX OF TABLES 5](#_Toc101227536)

[Index of figures 6](#_Toc101227537)

[Abstract 1](#_Toc101227538)

[Acknowledgment 2](#_Toc101227539)

[Chapter 1: Introduction 3](#_Toc101227540)

[1.1. Background and Motivation 3](#_Toc101227541)

[1.2. Objectives 3](#_Toc101227542)

[1.3. Structure 3](#_Toc101227543)

[Chapter 2: Literature review 4](#_Toc101227544)

[Introduction 4](#_Toc101227545)

[Sentiment analysis 4](#_Toc101227546)

[Accuracy of Sentiment analysis 5](#_Toc101227547)

[Issues with Sentiment analysis 5](#_Toc101227548)

[Sentiment analysis and Twitter based dataset 6](#_Toc101227549)

[Gaps in Research 7](#_Toc101227550)

[Chapter 3: Research Methodology 8](#_Toc101227551)

[Dataset 8](#_Toc101227552)

[Choose sentiment analysis algorithms 8](#_Toc101227553)

[Naïve Bayes (NB) 8](#_Toc101227554)

[Decision Tree 9](#_Toc101227555)

[Feature representation 9](#_Toc101227556)

[Pre-processing 9](#_Toc101227557)

[Bag of words 9](#_Toc101227558)

[TF-IDF 10](#_Toc101227559)

[Chapter 4: Experiments, Evaluation and Results 11](#_Toc101227560)

[Experimental design 11](#_Toc101227561)

[Evaluation metrics 11](#_Toc101227562)

[Experimental results 11](#_Toc101227563)

[Practical implications 11](#_Toc101227564)

[Chapter 5: Discussion 12](#_Toc101227565)

[Chapter 6: Project Management 13](#_Toc101227566)

[Chapter 7: Conclusion 14](#_Toc101227567)

[Reflection 15](#_Toc101227568)

[References 16](#_Toc101227569)

[Appendix 18](#_Toc101227570)

# INDEX OF TABLES

**No table of figures entries found.**

# Index of figures

**No table of figures entries found.**

# Abstract

# Acknowledgment

I'd like to express my gratitude to my supervisor, Dr Mark Elshaw, a lecturer in Computer Science at Coventry University, for his unwavering support throughout the research. I was able to enhance my work and complete the dissertation assignment to the best of my ability by offering comments and opinions. I also wanted to express my gratitude to Mark for all of his assistance and encouragement throughout my academic career.

# Chapter 1: Introduction

## Background and Motivation

Online hate has been highlighted as a big problem on online social media platforms, defined as abusive language, aggressiveness, cyberbullying, hatefulness, insults, personal assaults, provocation, racism, sexism, threats, or toxicity (Saad & Yang, 2019). As mentioned in the project proposal (Iftikhar, 2021). Abusive language is becoming increasingly common in online discussions. It poses a danger to freedom of expression, degrades the dignity of those targeted, and inhibit healthy and beneficial dialogue. Hate speech is not a clear-cut category; it appears to be part of a continuum of discriminatory discourse and is often shown using indirect linguistic means. Words can have similar vocabulary, but the toxicity of the comments is vastly different. Content moderators can't keep up with the flood of user-generated content quickly enough to keep everyone safe. hate-filled content can have considerable psychological risks to a content moderator.

To offer a healthy discussion environment on the Web, automated identification of conflictual languages is required. Hate speech may now be detected more accurately in textual streams because of recent improvements in Natural Language Processing and Natural Language Comprehension. Machine learning algorithms are thought to be the most effective for detecting conflicting languages (Iftikhar, 2021). In this project, pre-existing tools and algorithms are trained on a dataset and it is evaluated that tool performs best in the detection of abusive language.

Abusive language on an online forum is counter-intuitive and can cause a lot of issues for people of all backgrounds by suppressing their freedom of speech stopping them from expressing themself fully and openly in a positive manner. According to a 2014 Pew Report 4, 73 percent of adult Internet users have witnessed or experienced online harassment, with 40 percent having directly experienced it. Victims of online abuse are frequently from society's most vulnerable groups. Psychological distress, radicalization, and even self-harm and death can result from internet exposure to poisonous and hostile statements (Iftikhar, 2021). The aim of this project was to contribute in the detection of abusive language on any online platform to help reduce this form of oppressive behaviour and help improve these people’s lives (Iftikhar, 2021).

## Objectives

The main objectives of this project are as follows:

## Structure

# Chapter 2: Literature review

## Introduction

Can running sentiment analysis algorithms on tweets be used to detect if they contain toxic language?

Sentiment analysis is a relatively new idea, the research base has not yet expanded to encompass a wide range of applications in detail. Traditionally abusive language would be screened by manual moderation. This project aims to investigate the application of sentiment analysis classification techniques to detect the presence of toxic language in a tweet. This chapter will summarise and assess previous studies in this area. Prior research on sentiment analysis algorithms and how they've been employed in comparable ways to those used in this study.

## Sentiment analysis

The management of sentiments, views, and subjective language is referred to as sentiment analysis. Sentiment analysis analyses several tweets and reviews to offer understanding information on public opinion (Ain et al., 2017). Sentiment analysis is closely connected to (or can be regarded a subset of) computational linguistics, natural language processing, and text mining as a subject of study. It  tries to solve issues long researched in other fields of discourse using new methods supplied by data mining and computational linguistics. It is based on the study of emotional state (psychology) and judgement (appraisal theory). The purpose of sentiment analysis is to determine if the sentiment represented in the document accurately reflects the author's genuine intent (Mejova, 2009, p. *5-6*). Lexicon-based and machine learning approaches are the two main methods for extracting sentiment analysis from text (twitter) (Saad & Yang, 2019; Prakash & Aloysius, 2019). Lexicons are mostly used to query social network APIs for examples of offensive material. Lexicons may soon become out-of-date as users invent new abusive terms to get around censors, and they aren't immune to spelling and typo errors. In other cases, offensive letters may not contain any words or idioms that are typically deemed harsh when used alone (Kiritchenko, Nejadgholi and Fraser, 2021; Salminen et al., 2020). Machine learning approaches are trained using training data with known outputs, allowing them to perform with unknown test data (Saad & Yang, 2019; Iftikhar, 2021).

Refugees, women, a race, or a religion are often targets of hate speech. In online news comments, the media and police are the most often targets of hate. Hatred is more likely to be directed at high-profile social media users (Salminen et al., 2020). The most common targets for hate speech on the internet include race, behaviour, physical appearance, sexual orientation, class, gender, ethnicity, disability, and religion (Silva et al., 2016). Enmity based on race, religion, disability, sexual orientation, or transgender identity is illegal in the UK and is categorised as hate crime ("Hate crime | The Crown Prosecution Service", 2022). A negative effect of online hate speech is user exiting a toxic discussion, silencing or reduced participation in online social media, radicalization, group polarisation where previously held prejudices are enforced, degraded quality ("health") of an online community, offline violence and security threats, and decreased feelings of safety and wellbeing of online users (Salminen et al., 2020). Social media platforms like Facebook and Twitter have regulations against hate speech protecting their users from hate speech (MacAvaney et al., 2019). Twitter uses machine learning to classify toxic tweets in real time ("HateLab", 2022). The majority of computer science research in this topic focuses on automating the detection of online hatred (Salminen et al., 2020).

This research will contribute to the study of detecting hate speech in current research by filling in the gaps in the current research, this research can be studied and improved by future researchers. Making improvements in hate speech detection can potentially help improve tools for hate speech detection such as tools used by Twitter thus aiding in reduction of hate speech online and assisting law enforcers trace the origins of hate speech on a larger scale with better quality screening. So, this research can help researchers, law enforcers in their respective fields and social media platforms such as Twitter. It can help improve the lives of people with different race, behaviour, physical appearance, sexual orientation, class, gender, ethnicity, disability, and religion by helping in a research body creating tools to reduce and trace hate crime.

This research will apply different machine learning algorithms on labelled data to train and test models to classify tweets as toxic or not toxic.

## Accuracy of Sentiment analysis

A near 100% accuracy can be achieved on toxic language binary classification as demonstrated by Dhamija et al. using a combination of Sent2Vec and decision tree model (Iftikhar, 2021). Users' option to dispute automated decisions and seek human review, or even completely opt out of automated decisions. This is a crucial ethical issue regarding abusive language detection and moderation, given the ambiguity of language and the necessity to maintain freedom of speech. Many research papers estimate accuracies in the 80 percent range, which means that 1 in every 5 automated judgments will be "incorrect" (and even "correct" decisions may be debatable owing to the extremely subjective nature of the job) (Kiritchenko et al., 2021).

The challenge of automated sentiment analysis is becoming more and more of a research topic. Although sentiment analysis  is a significant field with several applications, it is apparent that it is not an easy undertaking with numerous obstacles associated to natural language processing. Recent sentiment analysis research is still plagued by theoretical and technological difficulties that limit its overall polarity detection accuracy. When a single approach is tested on a single dataset in a specific domain, the findings demonstrate that it has a reasonably high overall accuracy. When the amount or domain of the data changes, however, the suggested method's trustworthiness is called into doubt (Dang, Moreno-García & De la Prieta, 2020). Not having data features for all social media platforms—limits researchers' ability to use it for cross-platform application tests (Salminen et al., 2020). The total number of characters permitted in a tweet on Twitter is limited. As a result when tweeting, people frequently employ unusual terms and acronyms (Ibrohim et al., 2019). The mentioned is different for different platforms and hence it could be another reason why a model trained on data from a specific platform may not work on data from another platform.

The main objective of algorithmic fairness is to design systems whose outputs are equally accurate for all subsets of the population, even though improvement of algorithmic fairness might come at a cost of lower overall accuracy on a particular test set (Kiritchenko et al., 2021).

## Issues with Sentiment analysis

Word sense disambiguation (WSD) is the computational capacity to determine the meaning of words in context. WSD is an AI-complete issue, which means it is a task whose solution is at least as challenging as the most difficult AI problems.

Because human language is ambiguous, many words can be construed in a variety of ways depending on the context. Take the following sentences as an example:

(a) Bass noises are audible to me.

(b) Grilled bass is a favourite of theirs.

The two uses of the term bass plainly suggest two distinct meanings: low-frequency tones and a species of fish, respectively. Unfortunately, determining the exact meaning that a word takes on in context is not as straightforward as it appears. While most humans are unconcerned by ambiguities in language, computers must take unstructured textual material and convert it into data structures that must be evaluated to function (Navigli, 2009).

The term sick in its traditional connotation of "ill" may also mean "awesome" in a positive slang sense, as in "the band's album is sick." Slang's expressive character demonstrates its social role, as it allows for successful communication and information exchange across groups of people with different social identities (Pei, Sun & Xu, 2019). Pei, Sun and Xu in their experiment on detecting slang using machine learning demonstrates that slang can be detected successfully using machine learning with above 80 % f1 score and 90 % above accuracy.

The difficulties of sarcasm, as well as the value of sarcasm identification in sentiment analysis, have sparked interest in automatic sarcasm detection as a study topic. The line "I love it when my son rolls his eyes at me" should be classified as sarcastic, but "I love it when my son gives me a present" should be classified as non-sarcastic. Because sarcasm may be expressed in a variety of ways, this challenge is hard to solve (Joshi et al., 2017). As in their experiment Joshi et al., demonstrate that above 90 % accuracy and f1 score can be achieved on tweets.

The fact that the same term might have many meanings is a difficulty with keyword matching. Consider the terms "white trash" and "white trash cans," which have the same spelling but very different hatred content. Another example is the term "fruit," which is non-abusive in general but may signify disparaging slang for a gay person when used in a specific context. This topic is known as word-sense disambiguation in Natural Language Processing (NLP), and it is regarded extremely difficult (Salminen et al., 2020). In 2018, it was determined that merely adding positive terms to otherwise offensive postings, such as love, was enough to deceive the Perspective API toxicity detector. System security must be ensured against both simple and complex threats (Kiritchenko et al., 2021).

Although abusive language detection is used to protect individuals, it may also be used to silence disadvantaged voices. Black activists, for example, have claimed that Facebook deletes posts in which they disclose their own racist encounters (Kiritchenko et al., 2021).

In the mentioned problems in the first experiment (Pei, Sun & Xu, 2019) slang was the focal point of the experiment, in the second experiment (Joshi et al., 2017) sarcasm was the focal point of the experiment, a high f1 and accuracy was achieved because of this approach but in this experiment, detecting toxic tweets will be the focal point. Slang and sarcastic words will not specifically be targeted in this experiment. Other mentioned problems will also not be specially targeted in this experiment as the primary goal of this experiment will be detecting toxic tweets.

## Sentiment analysis and Twitter based dataset

"Social media" is the term used to describe microblogs. Twitter, Facebook, YouTube, Instagram, WhatsApp, Snapchat, LinkedIn, and other social media platforms provide real-time communication with little if any content restrictions (Iftikhar, 2021). Social media data may be a valuable resource for tracking public events, analysing citizen mood, and providing early-warning signs to better understand certain elements of a phenomena and make more timely and appropriate judgments (Biffignandi, Bianchi & Salvatore, 2018). In this project labelled tweets will be used with sentiment analysis to determine the toxicity of each tweet. The tweets analysed will be in the English language. Twitter has limited characters permitted in each tweet making use of unusual acronyms common (Ibrohim et al., 2019). Which makes Twitter an excellent platform to check machine learning algorithm’s ability to detect creative, uncommon, and evolving language.

Research in the US determined that 10% of users who are the most active in terms of tweeting are responsible for 80% of all tweets produced by American users. Most Twitter users are modestly engaging; the 10% who tweet the most frequently are primarily women and focus on politics. Users of Twitter are younger, more educated, and richer than the overall public (Wojcik & Hughes, 2019). As it can be seen by the mentioned study the data present on twitter has a extreme bias in the US alone so the results from this study can be implemented on twitter but its application on other social media platforms is not in the scope of this study and hence its application on other platforms is subject to research/tests.

The data from twitter could be in the form of a binary or multiclass classification problem. This data used in this study will be a binary classified. In current studies native bayes algorithm used by Mohiyaddeen & Siddiqui, 2021 has achieved an accuracy of 93 % and even above 99 % accuracy can be achieved using the decision tree algorithm as demonstrated by Dhamija et al., 2021, but the end results depend highly on the methodology of data pre-processing. These exceptional results of 99 % by Dhamija et al., 2021 leave out a key aspect of the dataset which is the degree of class imbalance. The results of 99 % accuracy can be achieved but if the classes are highly imbalanced then the results can have a high bias and cannot effectively test positive or negative tweets correctly depending on the ratio of positive to negative tweets (Iftikhar, 2021).

This project's study is confined to publicly available free algorithms; each will take a different approach to sentiment analysis; thus, it will be examined to see which technique is the most accurate for classifying tweets as toxic or nontoxic.

## Gaps in Research

As stated by Kiritchenko et al., 2021 in current datasets, NLP researchers are looking at two types of label bias: annotator bias and task design prejudice. The subjectivity and ambiguity of abusive behaviour criteria lead to biases. A frequent way to dealing with subjectivity is to label an occurrence by majority vote; nevertheless, this may serve to promote the majority's opinions while silencing minority voices (Iftikhar, 2021).

Despite considerable previous research on detecting of toxic language on Twitter using sentiment analysis the research on hate detection using a balanced dataset is scarce. The studies that were published tend to use imbalanced datasets and or do not focus on balance dataset at all. Without balanced dataset hate evaluation, the generalizability of models built on datasets will have a high bias. Research efforts are needed for developing online hate classifiers that are trained and tested on a balanced dataset.

This research will use a balanced dataset and show the results using a confusion matrix along with the corresponding accuracy scores to evaluate the overall accuracy as well as how many toxic or nontoxic tweets were classified correctly and incorrectly using each algorithm.

# Chapter 3: Research Methodology

## Dataset

When choosing the datasets for this study, the following criteria was used: (1) the dataset must be in English, (2) the dataset must be available at the time of the study, and (3) the dataset should be having a balance between negative and positive tweets.

The dataset has been provided by the Kaggle user Ashwin U Iyer and can be found online at; <https://www.kaggle.com/ashwiniyer176/toxic-tweets-dataset>. There are 3 columns and 54313 rows in this dataset. The tweets present in the dataset are in English, the dataset is available publicly at the time of this study. In the dataset the label 0 denotes non-toxic tweets, and the label 1 identifies toxic tweets. Non-toxic tweets appear 32592 times in the dataset while toxic tweets appear 24153 times in the dataset. This was the most balanced dataset found which was publicly available. The dataset will be balanced in the pre-processing stage of this research making the number of toxic and nontoxic tweets get equall represention.

The data used in this project is a pre-existing Twitter dataset, it is publicly available, and people cannot be identified in the tweets these precautions were considered when choosing the dataset keeping ethical implications of using the dataset in mind. Sensitive information on people such as their name and location are not present in the dataset.

The dataset has sarcastic tweets and tweets containing slang terms which could potentially reduce the accuracy of the model but removing or dealing with these tweets is not the focal point of this study so they will be kept in their original form. The sentence “@user i'm not interested in a #linguistics that doesn't address #race &amp; . racism is about #power. #raciolinguistics bringsÃ¢Â€Â¦” contains the word “not interested” which to a human reader can easily be read and understood but the computer may pick up the word “interested” and determine the results. This problem can easily be solved by replacing the word “interested” with its antonym and removing the word “not” from the sentence. The word could also be replaced by “not-interested” so the computer knows to treat it as one word.

The possible data set concerns will not be addressed using the methods discussed. This will allow examining on how well the algorithms compute the sentiment of complex human language.

## Choose sentiment analysis algorithms

The classification algorithms are discussed in this section. These were picked based on the nature of the challenge and their previous research results. Native Bayes and decision tree algorithms have previously given above 90 % accuracy on similar unbalanced datasets (Iftikhar, 2021) they will be used in this research to compare if the same results can be achieved on a balanced dataset. This research applies auto-sklearn algorithms on the pre-processed data as well. Using auto-sklearn 4 best performing algorithms are chosen and are further evaluated.

### Naïve Bayes (NB)

The Nave Bayes (NB) classifier is used as a baseline in machine learning models. The algorithm is a straightforward probabilistic strategy based on Bayes' theorem, conditional independence, and the total probability theorem. It generates probability by counting the frequencies and combinations of values in a dataset. Even though conditional independence is rarely true in real-world data, the approach works well in supervised classification applications such as text analysis (Salminen et al., 2020). Text categorization difficulties are ubiquitous, it's only natural to include NB in this research.

### Decision Tree

A typical data mining approach for constructing classification systems based on many covariates or developing prediction algorithms for a target variable is decision tree methodology. A population is divided into branch-like segments that form an inverted tree with a root node, internal nodes, and leaf nodes. The method is non-parametric, which means it can handle huge, complex datasets without imposing a complex parametric framework. It's simple to anticipate the outcome for future records using a tree model drawn from prior data. In medical research, too many categories of one categorical variable or severely skewed continuous data are prevalent. Decision tree models can aid in these situations by determining how to compress categorical data into a more manageable number of categories or how to partition severely skewed variables into ranges (Song & Ying 2015).

### Auto-sklearn

Auto-sklearn is a machine learning toolbox that may be used instead of scikit-learn estimators. Auto-sklearn takes care of algorithm selection and hyperparameter adjustment for machine learning users. It takes use of current advances in Bayesian optimization, meta-learning, and ensemble building ("auto-sklearn — AutoSklearn 0.14.7 documentation", 2022). The algorithms auto-sklearn uses are adaboost, bernoulli\_nb, decision\_tree, extra\_trees, gaussian\_nb, gradient\_boosting, k\_nearest\_neighbors, lda, liblinear\_svc, libsvm\_svc, mlp, multinomial\_nb, passive\_aggressive, qda, random\_forest and sgd ("Interpretable models — AutoSklearn 0.14.7 documentation", 2022).

### Adaboost

Adaboost is an iterative technique that employs a variety of classifiers for the same training set before combining them to get the final strongest classifier. The procedure is carried out by altering the distribution of a weight D that is consistently initialised before moving on to the next classifier. Adaboost classifier may be used to filter out non-essential training data characteristics and focus on the most important training data (Feng, 2019).

### Extra Trees

### Gradient Boosting

### k Nearest Neighbours

### Liblinear SVC

### Libsvm SVC

### MLP

### Multinomial NB

### Gaussian NB

### Passive Aggressive

### QDA

### Bernoulli NB

A supervised learning method based on the 'naive' assumption that the trial outcome is unrelated. It considers all tweets to be a collection of tokens. All words in the tweets contribute to the score of each category in Bernoulli NB. It compares the data to a list of terms to determine the appropriate polarity of the tweets. Word occurrence vectors are employed to train and test Bernoulli NB classifiers when data involves text classification. This algorithm's decision rule is as follows:

𝑃(𝑒1𝑖 | 𝑒2) = 𝑃(𝑒2)𝑒1 + (1 − 𝑃(𝑖|𝑒2))(1 − 𝑒1𝑖 )

This is not to be confused with multinomial NB. It immediately penalises any characteristic i that is not acceptable, indicating class e2 (Prema Arokia Mary et al., 2021).

### LDA

LDA is an unsupervised machine learning approach for recognising the latent topic structure of textual texts, as well as document modelling and categorization. In machine learning, LDA is one of the most widely used probabilistic text modelling approaches. Bayes estimation is used in LDA. LDA implies that each document in a corpus is a random mix of latent topics, with each latent topic defined by a word distribution. And while these latent themes may be derived from a group of documents, the percentage of each subject in each document varies (Mohammed & Al-augby, 2020).

### Random forest

### SGD

## Feature representation

Developing powerful text classifiers necessitates feature engineering and extraction (Salminen et al., 2020). Following Salminen et al., 2020 this research will try out a variety of feature kinds with escalating levels of complexity.

### Pre-processing

Due to the noise in language, text processing is a challenging undertaking that must be approached with caution to prevent losing any significant aspects (Fehn Unsvåg & Gambäck, 2018).

This research will follow Fehn Unsvåg & Gambäck, 2018’s example and in the pre-processing stage (i) removal of Twitter specific information (user mentions, emoticons, retweets, URLs, and hashtag symbols; only retaining textual content), (ii) tokenization, (iii) lowercasing, and (iv) stop word removal (with different stop word lists for the datasets) will take place. This research will incorporate an extra step in pre-processing which is (v) lemmatization following the example of Anandarajan et al., 2018. The pipeline's steps normalise the data, resulting in a reduction in the number of dimensions in the text dataset. In the decisions made during the process, there is a balance between preserved knowledge and reduced complexity. Each phase eliminates redundant data from the original text. The analysis will be more successful if the text data is properly pre-processed. Cleaning and preparing data with care and precision makes the analytical process go more smoothly (Anandarajan et al., 2018).

Bag of words and TF-IDF representations were chosen to be used in this research. After discovering that many tweets were marked racist owing to the use of harsh terms, (Kwok and Wang, 2013 as cited in Fehn Unsvåg & Gambäck, 2018) created a lexicon based only on unigram characteristics. However, because this method fails to capture word connections, (Nobata et al., 2016 as cited in Fehn Unsvåg & Gambäck, 2018) used syntactic characteristics in addition to n-grams and distributional semantic derived features. They discovered that integrating all characteristics yielded the greatest results, although character n-grams contributed the most as an individual feature. Character-based techniques were explored in depth by (Mehdad and Tetreault, 2016 as cited in Fehn Unsvåg & Gambäck, 2018), who found them to be superior to token-based approaches and other state-of-the-art methodologies.

### Bag of words

The bag-of-words (BoW) model is a popular text representation approach. In simple terms, this approach transforms a text into a vector as v = [x1, x2,..., xn], where xi signifies the ith word's occurrence. The core terms, which are generally the top n highest-frequency words, are gathered from the datasets. The occurrence feature might take the form of a binary, term frequency, or TF-IDF value (Yan et al., 2020).

### TF-IDF

Instead of counting the words, which would overemphasise common words, TF-IDF weights each word based on its relative frequency. The TF-IDF characteristics tell the model whether a term appears in a comment more frequently than it does in the rest of the text corpus. TF-IDF features have already been proven to be beneficial for detecting online hate (Salminen et al., 2020). The TF-IDF vocabulary is constructed during the model's training and then utilised for the test set, similar to BOW. Both BOW and TF-IDF are basic and well-proven text classification methods (Sahlgren M, et al., 2018 as cited by Salminen et al., 2020).

# Chapter 4: Experiments, Evaluation and Results

## Experimental design

In this research, the performance of multiple models trained using various techniques, alternative feature representations are compared. The models' performance is examined  using two baseline algorithms in addition to the alternative auto-sklearns algorithms. Auto-sklearn  is chosen because it is a simple way for software developers to adopt, it handles hyperparameter optimization and model assessment with a variety of algorithms, and it reduces programming time while providing excellent results.

On the Twitter dataset, the performance of the two best trained classifiers is evaluated. On the Twitter dataset, the instances of the test set performance of the best algorithms are displayed for comparison. Results from previously published research are also included in the paper for completeness. The results are not totally comparable between the source papers and this study because to changes in the dataset and training/test distribution as well as data distribution.

## Evaluation metrics

The test set (about 20% of the entire dataset) is used to evaluate the classifier's performance using two metrics: (a) F1 score and (b) receiver operating characteristic—area under the curve (ROC-AUC). The harmonic mean of accuracy and recall is the F1 score. The area under the ROC curve is a useful statistic to quantify overall model performance since the ROC computes accuracy and recall at all conceivable decision thresholds following the example of Salminen et al., 2020. The formula for calculating the F1 score is given (See Figure 1 F1 Score).

Graphical user interface, text, application, email

Description automatically generated

Avalible online at: https://hcis-journal.springeropen.com/articles/10.1186/s13673-019-0205-6

Figure 1 F1 Score

Receiver Operating Characteristic curve or ROC Curves a graph of the false positive rate (x-axis) vs. the true positive rate (y-axis) for a variety of candidate threshold values ranging from 0.0 to 1.0. It shows the false alarm rate vs the hit rate. The true positive rate is obtained by dividing the total number of true positives and false negatives by the number of true positives. When the actual outcome is positive, it reflects how well the model predicts the positive class (See Figure 2 True positive rate).

Figure 2 True positive rate

**True Positive Rate = True Positives / (True Positives + False Negatives)**

The false positive rate is obtained by dividing the total number of false positives by the total number of false positives and true negatives. It's also known as the false alarm rate since it sums up how frequently a positive class is anticipated while the actual result is negative (See Figure 3 False positive rate). Only the F1 measure and the ROC-AUC will be assessed and compared in this study. In this study, a confusion matrix will be utilised to visually evaluate how each class has been properly and incorrectly categorised (See Figure 4 Confusion matrix).

Figure 3 False positive rate

**True Positive Rate = True Positives / (True Positives + False Negatives)**

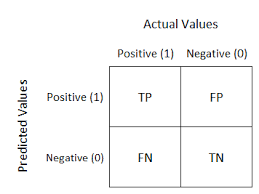


Figure 4 Confusion matrix

## Experimental results

### Decision Tree and Native bayes default models on 1000 features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | N-grams | Representation | Test F1 score | AUC |
| Native Bayes | unigram | BOW | 79 % | 79 % |
| Native Bayes | bigram | BOW | 54 % | 61 % |
| Native Bayes | trigram | BOW | 42 % | 55 % |
|  | | | | |
| Native Bayes | unigram | TFIDF | 79 % | 79 % |
| Native Bayes | bigram | TFIDF | 54 % | 61 % |
| Native Bayes | trigram | TFIDF | 42 % | 55 % |
|  | | | | |
| Decision Tree | unigram | BOW | 66 % | 67 % |
| Decision Tree | bigram | BOW | 45 % | 53 % |
| Decision Tree | trigram | BOW | 39 % | 52 % |
|  | | | | |
| Decision Tree | unigram | TFIDF | 65 % | 66 % |
| Decision Tree | bigram | TFIDF | 46 % | 55 % |
| Decision Tree | trigram | TFIDF | 39 % | 52 % |

Native Bayes outperforms the decision tree model using bag of words or TFIDF representation without any hyperparameter optimization. The best performance is obtained with unigrams distribution on both bag of words and TFIDF model representations using Native Bayes algorithm giving 79 % F1 score and 79 % AUC (See Table 1 Results default DT VS NB) when using 1000 feautres.

Table 1 Results default DT VS NB

Observing the results from the Native Bayes algorithms with the bag of words and TFIDF model representation using a unigram distribution both the representations have the same f1 score but the accuracy, precision and recall is higher on the bag of words representation by 1 % (See Figure 6 default NB BOW CM and Figure 5 default NB TFIDF CM). The model with the bag of word representation has classified 38 % non-toxic tweets and 42 % toxic tweets (See Figure 6 default NB BOW CM) correctly while the model with the TFIDF representation has classified 36 % non-toxic tweets and 44 % toxic tweets correctly (See Figure 5 default NB TFIDF CM).

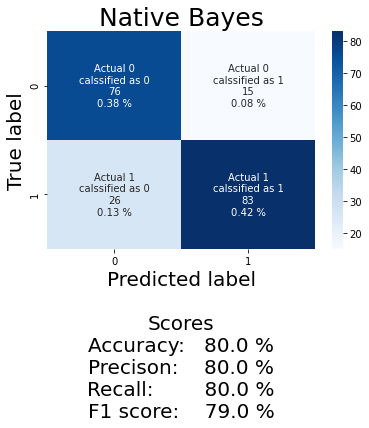
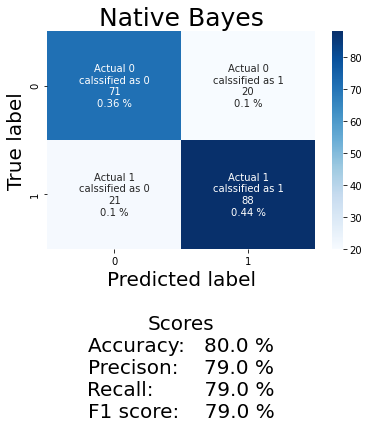


Figure 5 default NB TFIDF CM

Figure 6 default NB BOW CM

As this is a rounded results looking at the results of the AUC in decimal form the model with bag of words representation with an AUC of 0.7983163625365461 (See Figure 8 default NB BOW AUC) performs better than the TFIDF represented model with a AUC of 0.7937796148805323 (See Figure 7 default NB TFIDF AUC) by 0.004536747656013862 % greater AUC.

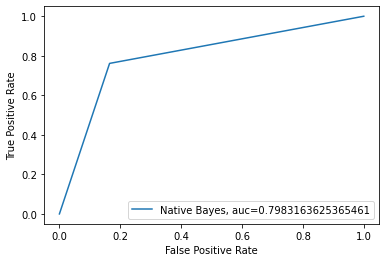
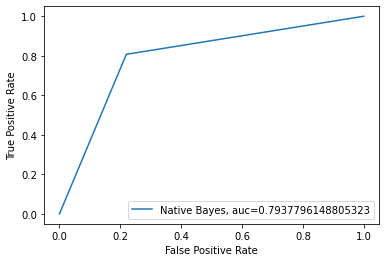


Figure 7 default NB TFIDF AUC

Figure 8 default NB BOW AUC

### Decision Tree and Native bayes default models on 2000 features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | N-grams | Representation | Test F1 score | AUC |
| Native Bayes | unigram | BOW | 84 % | 83 % |
| Native Bayes | bigram | BOW | 66 % | 68 % |
| Native Bayes | trigram | BOW | 56 % | 62 % |
|  | | | | |
| Native Bayes | unigram | TFIDF | 83 % | 83 % |
| Native Bayes | bigram | TFIDF | 66 % | 68 % |
| Native Bayes | trigram | TFIDF | 56 % | 62 % |
|  | | | | |
| Decision Tree | unigram | BOW | 76 % | 75 % |
| Decision Tree | bigram | BOW | 65 % | 66 % |
| Decision Tree | trigram | BOW | 56 % | 62 % |
|  | | | | |
| Decision Tree | unigram | TFIDF | 73 % | 73 % |
| Decision Tree | bigram | TFIDF | 62 % | 64 % |
| Decision Tree | trigram | TFIDF | 49 % | 57 % |

With adding 100 % more features results are observed to determine the linear improvement of both algorithms in 2000. Native Bayes gives the best results with a f1 score of 84 % and AUC of 83 % using a bag of words model representation and a unigram distribution (See Table 2 Results default DT VS NB 2).

Table 2 Results default DT VS NB 2

Comparing results with 1000 features and 2000 features it can be observed that given 1000 features the Native Bayes algorithm has roughly around 10 % improvement with a unigram distribution, has 22 % improving in the bigram distribution and a 33 % improvement in the trigram distribution using f1 score as a performance measure and bag of word model or TFIDF representation (See Table 2 Results default DT VS NB 2 and Table 1 Results default DT VS NB).

Now comparing the results with 1000 features and 2000 features of the decision tree algorithm using bag of word, TFIDF model and using f1 core as a performance measure it can be observed that the unigram distribution has 15 % improvement, the bigram distribution has 44 % and the trigram distribution has 25 % performance improvement (See Table 2 Results default DT VS NB 2 and Table 1 Results default DT VS NB).

An improvement percentage, also known as a percentage increase of a value, is a measure of an activity's or test's progress over time. It is measured using the formula given (See Figure 9 Improvement formula). Currently the decision tree model using bigram distribution has the most over all improvement. If the improvement was linear then given enough features the decision tree model using bigrams would have the best overall performance without hyperparameter optimization.

Figure 9 Improvement formula

Improvement = ( New Value-Original Value)/ Original Value \*100

### Decision Tree and Native bayes default models on 2500 features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | N-grams | Representation | Test F1 score | AUC |
| Native Bayes | unigram | BOW | 79 % | 79 % |
| Native Bayes | bigram | BOW | 68 % | 70 % |
| Native Bayes | trigram | BOW | 55 % | 61 % |
|  | | | | |
| Native Bayes | unigram | TFIDF | 79 % | 79 % |
| Native Bayes | bigram | TFIDF | 68 % | 70 % |
| Native Bayes | trigram | TFIDF | 55 % | 61 % |
|  | | | | |
| Decision Tree | unigram | BOW | 76 % | 75 % |
| Decision Tree | bigram | BOW | 62 % | 64 % |
| Decision Tree | trigram | BOW | 52 % | 59 % |
|  | | | | |
| Decision Tree | unigram | TFIDF | 72 % | 72 % |
| Decision Tree | bigram | TFIDF | 61 % | 64 % |
| Decision Tree | trigram | TFIDF | 50 % | 58 % |

Increasing the previous set of features 50 % it can be observed that all the results have a worsened in all algorithms with all model representations except the decision tree algorithm with the unigram distribution and bag of words representation as its results stay the same with the 2000 feature set and decision tree algorithm with the trigram distribution and TFIDF model representation has a improvement of about roughly 1 % from the 2000 feature set (See Table 2 Results default DT VS NB 2 and Table 3 Results default DT VS NB 3).

Table 3 Results default DT VS NB 3

The previous experiment was done to check the validity of the results of this research and the application of its results on features of increasing size as is the case of tweets. It was observed that even the same models trained with a set of data may perform poorly or better if additional data is added and trained with the same algorithms with all the same conditions.

### Auto-sklearn with max features set to 1000

After applying auto-sklearn when max features set to 1000 (because of hardware of the machine used in this project and time restrictions of completing the project) on bag of word and TFIDF model representations the best performing models were sgd, random\_forest, bernoulli\_nb and lda. bernoulli\_nb had 85 % accuracy, lda had an accuracy of 81 %, sdg had an accuracy of 78 % and random forest had an accuracy of 78 %. All of the mentioned models had this accuracy when they were applied of a unigram distribution. When models were applied on bigram and trigram distributions the accuracy was lower than 63% and 57% respectively which was obtained using lda on both cases.

### Decision Tree and Autos learn optimized models on 1000 features

## Practical implications

# Chapter 5: Discussion

# Chapter 6: Project Management

# Chapter 7: Conclusion

# Reflection

# References

Ain, Q. T., Ali, M., Riaz, A., Noureen, A., Kamran, M., Hayat, B., & Rehman, A. (2017). Sentiment analysis using deep learning techniques: a review. *Int J Adv Comput Sci Appl*, *8*(6), 424.

Anandarajan, M., Hill, C., & Nolan, T. (2018). Text Preprocessing. *Practical Text Analytics*, 45-59. <https://doi.org/10.1007/978-3-319-95663-3_4>

*auto-sklearn — AutoSklearn 0.14.7 documentation*. Automl.github.io. (2022). Retrieved 24 April 2022, from https://automl.github.io/auto-sklearn/master/.

Biffignandi, S., Bianchi, A., & Salvatore, C. (2018, June). Can Big Data provide good quality statistics? A case study on sentiment analysis on Twitter data. In *Proc. Int. Total Surv. Error Workshop Duke Initiat. Surv. Methodol (ITSEW-DISM)*.

Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics*, *9*(3), 483.

Fehn Unsvåg, E., & Gambäck, B. (2018). The Effects of User Features on Twitter Hate Speech Detection. *Proceedings Of The 2Nd Workshop On Abusive Language Online (ALW2)*. <https://doi.org/10.18653/v1/w18-5110>

Feng, X. (2019). Research of Sentiment Analysis Based on Adaboost Algorithm. *2019 International Conference On Machine Learning, Big Data And Business Intelligence (MLBDBI)*. https://doi.org/10.1109/mlbdbi48998.2019.00062

*Hate crime | The Crown Prosecution Service*. Cps.gov.uk. (2022). Retrieved 15 April 2022, from https://www.cps.gov.uk/crime-info/hate-crime.

*HateLab*. Developer.twitter.com. (2022). Retrieved 15 April 2022, from https://developer.twitter.com/en/community/success-stories/hatelab#:~:text=From%20this%20work%2C%20they%20developed,learning%20to%20classify%20toxic%20speech.

Ibrohim, M., Setiadi, M., & Budi, I. (2019). Identification of hate speech and abusive language on indonesian Twitter using the Word2vec, part of speech and emoji features. *Proceedings Of The International Conference On Advanced Information Science And System*. https://doi.org/10.1145/3373477.3373495

Iftikhar, S. (2021). *Investigation in using machine learning algorithms to detect if a tweet has a toxic language.* Coventry: Coventry Univeristy.

*Interpretable models — AutoSklearn 0.14.7 documentation*. Automl.github.io. (2022). Retrieved 24 April 2022, from https://automl.github.io/auto-sklearn/master/examples/40\_advanced/example\_interpretable\_models.html#sphx-glr-examples-40-advanced-example-interpretable-models-py.

Joshi, A., Bhattacharyya, P., & Carman, M. (2017). Automatic Sarcasm Detection. *ACM Computing Surveys*, *50*(5), 1-22. https://doi.org/10.1145/3124420

Kiritchenko, S., Nejadgholi, I., & Fraser, K. (2021). Confronting Abusive Language Online: A Survey from the Ethical and Human Rights Perspective. *Journal Of Artificial Intelligence Research*, *71*, 431-478. https://doi.org/10.1613/jair.1.12590

MacAvaney, S., Yao, H., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PLOS ONE*, *14*(8), e0221152. https://doi.org/10.1371/journal.pone.0221152

Mejova, Y. (2009). Sentiment analysis: An overview. *University of Iowa, Computer Science Department (p. 5-6)*.

Mohammed, S., & Al-augby, S. (2020). LSA &amp; LDA topic modeling classification: comparison study on e-books. *Indonesian Journal Of Electrical Engineering And Computer Science*, *19*(1), 353. https://doi.org/10.11591/ijeecs.v19.i1.pp353-362

Navigli, R. (2009). Word sense disambiguation. *ACM Computing Surveys*, *41*(2), 1-69. https://doi.org/10.1145/1459352.1459355

Pei, Z., Sun, Z., & Xu, Y. (2019, November). Slang detection and identification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)* (pp. 881-889).

Prakash, T. N., & Aloysius, A (2020). A Comparative study of Lexicon based and Machine learning based classifications in Sentiment analysis.

Prema Arokia Mary, G., Hema, M., Maheshprabhu, R., & Nageswara Guptha, M. (2021). Sentimental Analysis of Twitter Data using Machine Learning Algorithms. *2021 International Conference On Forensics, Analytics, Big Data, Security (FABS)*. https://doi.org/10.1109/fabs52071.2021.9702681

Saad, S., & Yang, J. (2019). Twitter Sentiment Analysis Based on Ordinal Regression. *IEEE Access*, *7*, 163677-163685. https://doi.org/10.1109/access.2019.2952127

Salminen, J., Hopf, M., Chowdhury, S., Jung, S., Almerekhi, H., & Jansen, B. (2020). Developing an online hate classifier for multiple social media platforms. *Human-Centric Computing And Information Sciences*, *10*(1). https://doi.org/10.1186/s13673-019-0205-6

Silva, L., Mondal, M., Correa, D., Benevenuto, F., & Weber, I. (2016, March). Analyzing the targets of hate in online social media. In *Tenth international AAAI conference on web and social media*.

Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130.

Wojcik, S., & Hughes, A. (2019). Sizing up Twitter users. *PEW research center*, *24*.

Yan, D., Li, K., Gu, S., & Yang, L. (2020). Network-Based Bag-of-Words Model for Text Classification. *IEEE Access*, *8*, 82641-82652. https://doi.org/10.1109/access.2020.2991074

# Appendix