**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 work and has not been copied in part or from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, the 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 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 |
| Supervisor Name: Mark Elshaw |
|  |

Table of Content

[INDEX OF TABLES 6](#_Toc106633716)

[Index of figures 7](#_Toc106633717)

[List of acronyms used in this project. 1](#_Toc106633718)

[Abstract 2](#_Toc106633719)

[Acknowledgements 3](#_Toc106633720)

[Chapter 1: Introduction 4](#_Toc106633721)

[I. Background and Motivation 4](#_Toc106633722)

[Objectives 5](#_Toc106633723)

[II. Structure 5](#_Toc106633724)

[Chapter 2: Literature review 1](#_Toc106633725)

[I. Introduction 1](#_Toc106633726)

[II. Sentiment analysis 1](#_Toc106633727)

[III. Accuracy of Sentiment analysis 2](#_Toc106633728)

[IV. Issues with Sentiment analysis 2](#_Toc106633729)

[V. Sentiment analysis and Twitter-based dataset 3](#_Toc106633730)

[VI. Gaps in Research 5](#_Toc106633731)

[Chapter 3: Research Methodology 6](#_Toc106633732)

[I. Dataset 6](#_Toc106633733)

[II. Choose sentiment analysis algorithms 6](#_Toc106633734)

[Naïve Bayes (NB) or Gaussian NB 6](#_Toc106633735)

[Decision Tree 7](#_Toc106633736)

[Auto-sklearn 7](#_Toc106633737)

[Adaboost 7](#_Toc106633738)

[Extra Trees 7](#_Toc106633739)

[Gradient Boosting 8](#_Toc106633740)

[k Nearest Neighbours 8](#_Toc106633741)

[Liblinear SVC 8](#_Toc106633742)

[Libsvm SVC 8](#_Toc106633743)

[MLP 9](#_Toc106633744)

[Multinomial NB 9](#_Toc106633745)

[Passive Aggressive 9](#_Toc106633746)

[QDA 9](#_Toc106633747)

[Bernoulli NB 9](#_Toc106633748)

[LDA 9](#_Toc106633749)

[Random forest 9](#_Toc106633750)

[SGD 9](#_Toc106633751)

[III. Pre-processing and feature representation 10](#_Toc106633752)

[Pre-processing 10](#_Toc106633753)

[Bag of words 10](#_Toc106633754)

[TF-IDF 10](#_Toc106633755)

[Chapter 4: Experiments, Evaluation and Results 11](#_Toc106633756)

[I. Experimental design 11](#_Toc106633757)

[II. Evaluation metrics 11](#_Toc106633758)

[III. Experimental results 12](#_Toc106633759)

[Decision Tree and Naive Bayes default models on 1000 samples 13](#_Toc106633760)

[Decision Tree and Naive Bayes default models on 2000 samples 14](#_Toc106633761)

[Decision Tree and Naive Bayes default models on 2500 samples 15](#_Toc106633762)

[Auto-sklearn with max samples set to 1000 16](#_Toc106633763)

[Bernoulli NB default models on 1000 samples 16](#_Toc106633764)

[Bernoulli NB hyperparameter optimization models on 1000 samples 17](#_Toc106633765)

[SDG default models on 1000 samples 17](#_Toc106633766)

[SDG hyperparameter optimization models on 1000 samples 18](#_Toc106633767)

[LDA default models on 1000 samples 19](#_Toc106633768)

[LDA hyperparameter optimization models on 1000 samples 19](#_Toc106633769)

[Cross-validation 20](#_Toc106633770)

[Removing least frequent words removed by the percentage of their occurrence in the data frame 21](#_Toc106633771)

[Removing the least frequent words by the number of their occurrence in the data frame 28](#_Toc106633772)

[Removing the most frequent words by the percentage of their occurrence in the data frame 32](#_Toc106633773)

[Removing words frequent or by their occurrence in a sentence cross-evaluation 37](#_Toc106633774)

[Replacing words starting with not with their synonym or not\_ 38](#_Toc106633775)

[Cross Evaluation between the best results of this study and previous studies 39](#_Toc106633776)

[IV. Practical implications 43](#_Toc106633777)

[V. Limitations and future research 43](#_Toc106633778)

[Chapter 5: Discussion 45](#_Toc106633779)

[Chapter 6: Conclusion 46](#_Toc106633780)

[Chapter 7: Reflection 47](#_Toc106633781)

[I. Project Management 47](#_Toc106633782)

[Gantt charts 47](#_Toc106633783)

[Agile 47](#_Toc106633784)

[Scrum 48](#_Toc106633785)

[Kanban 49](#_Toc106633786)

[Waterfall 50](#_Toc106633787)

[Choosing a project management approach 50](#_Toc106633788)

[Problems with generic work management methodologies 50](#_Toc106633789)

[My project management approach 52](#_Toc106633790)

[Project selection 52](#_Toc106633791)

[Finding a supervisor 52](#_Toc106633792)

[Creating a Gantt chart 52](#_Toc106633793)

[Asking Dr Elshaw for supervision 52](#_Toc106633794)

[Research on the Selected Project 53](#_Toc106633795)

[Write the project proposal 53](#_Toc106633796)

[Fill in the Ethical application 53](#_Toc106633797)

[Completing pre-processing 53](#_Toc106633798)

[Model application 53](#_Toc106633799)

[Project completion 53](#_Toc106633800)

[II. Legal and social, ethical and practical concerns 57](#_Toc106633801)

[Social and ethical concerns 57](#_Toc106633802)

[Legal concerns 58](#_Toc106633803)

[Practical concerns 60](#_Toc106633804)

[Legal and social, ethical, and practical concerns concerning this project 62](#_Toc106633805)

[References 63](#_Toc106633806)

# INDEX OF TABLES

[Table 1 List of acronyms 1](#_Toc106633807)

[Table 2 Hate speech detection classes and their instances 4](#_Toc106633808)

[Table 3 Results default DT VS NB 13](#_Toc106633809)

[Table 4 Results default DT VS NB 2 14](#_Toc106633810)

[Table 5 Results default DT VS NB 3 15](#_Toc106633811)

[Table 6 results auto-sklearn 16](#_Toc106633812)

[Table 7 Cross-Validation results 21](#_Toc106633813)

[Table 8 bag of words and TFIDF Bernoulli NB 27](#_Toc106633814)

[Table 9 bag of words and TFIDF Bernoulli NB 2 32](#_Toc106633815)

[Table 10 bag of words and TFIDF Bernoulli NB 3 36](#_Toc106633816)

[Table 11 bag of words and TFIDF Bernoulli NB 4 38](#_Toc106633817)

[Table 12 A comparison between Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 research and this research 41](#_Toc106633818)

[Table 13 Comparison between the current study with previous studies 42](#_Toc106633819)

[Table 14 Collaboration with Dr Elshaw 56](#_Toc106633820)

[Table 15 Risk analysis 56](#_Toc106633821)

# Index of figures

[Figure 1 Confusion Matrix (Features: Bigram (TFIDF), Classifier: SVM) 4](#_Toc106633822)

[Figure 2 Decision Tree 7](#_Toc106633823)

[Figure 3 F1 Score 11](#_Toc106633824)

[Figure 4 True positive rate 12](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633825)

[Figure 5 False positive rate 12](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633826)

[Figure 6 Confusion matrix 12](#_Toc106633827)

[Figure 7 default NB TFIDF CM 13](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633828)

[Figure 8 default NB BOW CM 13](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633829)

[Figure 9 default NB TFIDF AUC 14](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633830)

[Figure 10 default NB BOW AUC 14](#_Toc106633831)

[Figure 11 Improvement formula 15](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633832)

[Figure 12 bag of words Bernoulli NB 17](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633833)

[Figure 13 TFIDF Bernoulli NB 17](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633834)

[Figure 14 TFIDF SDG 18](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633835)

[Figure 15 bag of words SDG 18](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633836)

[Figure 16 bag of words SDG 2 18](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633837)

[Figure 17 TFIDF SDG 2 18](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633838)

[Figure 18 bag of words LDA 19](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633839)

[Figure 19 TFIDF LDA 19](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633840)

[Figure 20 TFIDF LDA 2 20](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633841)

[Figure 21 bag of words LDA 2 20](#_Toc106633842)

[Figure 22 bag of words and TFIDF Bernoulli NB 22](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633843)

[Figure 23 bag of words and TFIDF Bernoulli NB 2 23](#_Toc106633844)

[Figure 24 bag of words and TFIDF Bernoulli NB 3 24](#_Toc106633845)

[Figure 25 bag of words and TFIDF Bernoulli NB 4 24](#_Toc106633846)

[Figure 26 bag of words and TFIDF Bernoulli NB 4 25](#_Toc106633847)

[Figure 27 bag of words and TFIDF Bernoulli NB 5 25](#_Toc106633848)

[Figure 28 bag of words and TFIDF Bernoulli NB 6 26](file:///C:\Users\User\Desktop\6001%20CEM%20-%20Dissertation.docx#_Toc106633849)

[Figure 29 bag of words and TFIDF Bernoulli NB 7 27](#_Toc106633850)

[Figure 30 bag of words and TFIDF Bernoulli NB 8 28](#_Toc106633851)

[Figure 31 bag of words and TFIDF Bernoulli NB 9 29](#_Toc106633852)

[Figure 32 bag of words and TFIDF Bernoulli NB 10 29](#_Toc106633853)

[Figure 33 bag of words and TFIDF Bernoulli NB 11 30](#_Toc106633854)

[Figure 34 bag of words and TFIDF Bernoulli NB 12 30](#_Toc106633855)

[Figure 35 bag of words and TFIDF Bernoulli NB 13 31](#_Toc106633856)

[Figure 36 bag of words and TFIDF Bernoulli NB 14 32](#_Toc106633857)

[Figure 37 bag of words and TFIDF Bernoulli NB 15 33](#_Toc106633858)

[Figure 38 bag of words and TFIDF Bernoulli NB 16 34](#_Toc106633859)

[Figure 39 bag of words and TFIDF Bernoulli NB 17 34](#_Toc106633860)

[Figure 40 bag of words and TFIDF Bernoulli NB 18 35](#_Toc106633861)

[Figure 41 bag of words and TFIDF Bernoulli NB 19 36](#_Toc106633862)

[Figure 42 bag of words and TFIDF Bernoulli NB 20 37](#_Toc106633863)

[Figure 43 bag of words and TFIDF Bernoulli NB 21 39](#_Toc106633864)

[Figure 44 Gantt chart 52](#_Toc106633865)

# List of acronyms used in this project.

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| TFIDF | Term frequency–inverse document frequency |
| BOW | Bag of words |
| DT | Decision tree |
| RF | Random forest |
| NB | Naive Bayes |
| LDA | Linear discriminant analysis |
| QDA | Quadratic Discriminant Analysis |
| KNN | K-Nearest-Neighbors |
| SDG | Stochastic Gradient Descent |
| SVM | Support vector machines |
| MLP | Multi-layer Perceptron |
| SVC | Support Vector Classification |
| BERT | Bidirectional Encoder Representations from Transformers |
| LSTM | Long short-term memory |
| CNN | Convolutional Neural Network |

Table 1 List of acronyms

# Abstract

The analysis of human written language to yield emotion polarity, general positivity or negativity, and ratings for individual emotions such as rage is known as sentiment analysis. The use of sentiment analysis to detect and classify hostile human discourse is known as abusive language classification. Previous applications have been conducted studying the use of sentiment analysis on Twitter to categorise abusive written human discourse.

The goal of this study was to learn more about the applications of sentiment analysis algorithms, specifically if they can be used to reliably categorise abusive human language on a balanced Twitter dataset.

Aims: Test the accuracy of sentiment algorithms on a Twitter dataset by using the F1 score and AUC.

Test forecasting algorithms to see how accurately unseen/new tweets can be predicted.

Remove bias scores by confusion matrix evaluation.

Improve understanding of tweet classification.

There has been relatively little study on non-subjective ways of identifying abusive tweets. Previous research has found that the categorisation of offensive tweets is skewed. The goal of this study is to develop a non-biased prediction of abusive tweet classification and assess its accuracy. To categorise abusive tweets, this study employs affective computing approaches and Sentiment analysis algorithms.

The dataset used in this research is publicly available.

This research utilised tweets to see the performance of sentiment analysis algorithms in predicting abusive content from tweets correctly. 1000 tweets were mainly utilised in the study. In this study, sentiment analysis was demonstrated to properly classify the tweets present in a Twitter dataset.

The study then used the first 800 tweets to see if the remaining 200 could be predicted accurately. In this research Auto-sklearn was used, from it, Adaboost, ExtraTrees, Gradient Boosting, k Nearest Neighbours, Liblinear SVC, Libsvm SVC, MLP, Multinomial NB, Passive Aggressive, QDA, Bernoulli NB, LDA, Random Forest, SGD, Decision Tree methods were employed in the study with TFIDF and bag of words representation along with unigram, bigram and trigram word distribution. The words' sentences from the dataset were also removed to see their impact on the models' performance. To see if the prediction was accurate and if it was able to classify the tweets, the projected 200 tweets were compared to the actual 200 tweets. The results were also evaluated to see the accurate classification of tweets by class using a confusion matrix and 10 cross-validations were also performed overall 1000 tweets to test the model’s reliability.

When employing the different algorithms to detect abusive tweets, the researchers discovered that it was possible to quite accurate classify toxic tweets. It was determined that either TFIDF or bag of words representation when applied on a unigram word distribution in combination with Bernoulli NB algorithms gave the best results when 25 % of the least common words by their occurrence by percentage were removed from the data. This study has the potential to make a significant influence on the actual world. It may be utilised in future hate classifier research and give helpful information that can be used to enhance hate classifiers, resulting in improved hate crime detection on all types of online forums thus, lowering or even preventing different forms of online hate crime.

# Acknowledgements

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 his assistance and encouragement throughout my academic career.

There have been various tools and technologies used in this project which must be acknowledged and are as follows. (1) Python is a high-level, interpreted programming language that may be used for a variety of tasks. Python 3 was used as a primary programming language for this project (Van Rossum & Drake, 2009). (2) NumPy is perhaps the most important Python module for scientific computing. Numpy was with Python in this project (Harris et al., 2020). (3) In computer programming, Pandas is data manipulation and analysis software package designed for the Python programming language. Pandas were used in this project (McKinney et al., 2010). (4) Matplotlib is a graphing package for Python and NumPy, the Python numerical mathematics extension. Matplotlib was used in this project for creating graphs (Hunter, 2007). (5) Scikit-learn is a Python-based machine learning package that is available for free. Scikit-learn was used in this project to use various machine learning algorithms (Pedregosa et al., 2011). (6) Auto-sklearn is a toolset for automating machine learning. Auto-sklearn was used in this project (Feurer et al., 2020).

The Kaggle user Ashwin U Iyer has to be acknowledged for providing the dataset used in this project. According to Iftikhar, 2021 the dataset used in this project has been provided by the Kaggle user Ashwin U Iyer and can be found online at; <https://www.kaggle.com/ashwiniyer176/toxic-tweets-dataset>.

# Introduction

## Background and Motivation

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

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 inhibits 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 a 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 and hate-filled content can have considerable psychological risks to a content moderator.

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 and stopping them from expressing themself fully and openly in a positive manner. According to a 2014 Pew Report 4, 73 %of adult Internet users have witnessed or experienced online harassment, with 40 % 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).

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).

According to Ortigosa et al., 2014 the study reported in Das & Chen, 2001, which analyses messages made on stock boards to extract market sentiment, was one of the first to use the phrase "sentiment analysis" as it is known. Sentiment analysis is used in a variety of fields, including business and marketing, politics, health, and public policy. Machine learning methods benefit from more data since they take more time and data to train (Drus & Khalid, 2019). The availability of massive datasets, along with advances in algorithms and exponential increases in computer power, has sparked unprecedented interest in machine learning in recent years (Schmidt et al., 2019). Sentiment analysis is performed used on data obtained by various platforms the most popular being Twitter (Drus & Khalid, 2019). Using machine learning to perform sentiment analysis can produce models that can detect sentiments from tweets with above 90 % accuracy (Iftikhar, 2021).

Observing the research from Ortigosa et al., 2014, Drus & Khalid, 2019 and Schmidt et al., 2019 it can be concluded that sentiment analysis in a short time frame became a major area of study because of the availability of large datasets and advances in algorithms and exponential increases in computer power. As Drus & Khalid, 2019 state that the most amount of sentiment analysis is performed on Twitter thus this research keeping aims to do the same so results from this research can be compared with a wide range of research and any potential improvements noted in this research can be thoroughly evaluated.

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. This project aimed to contribute to 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). The motivation of this project is to help improve online hate classifiers and reduce online textual abuse by providing a better understanding of online hate classifiers. The main objectives of this project are as follows:

### Objectives

The main objectives of this project are as stated by Iftikhar, 2021 which are:

* To create abusive language, detection models.
* To review the created models' performance.
* To create a model that can detect abusive langue on unseen data with high performance.
* It is of high importance that the model has good results on all metrics while keeping the actual positive and actually negative classifications balanced whether they are correctly or incorrectly represented to keep the bias as low as possible (See Gaps in Research).
* The previous works on the same topic can be used and improved upon in this research.
* This research can be used to provide valuable information for future research or online hate classifiers.

## Structure

This dissertation will address a wide range of important subjects. The literature review notes in chapter 2 will help in understanding why this study is necessary by looking at previous studies. The technique underlying the implementation process will be covered in chapter 3 to assist gain insight and understanding of each aspect of the code for a wide range of audiences. In the stated assessment metrics, Chapter 4 will summarise the outcomes obtained using the suggested model. In chapter 5, I'll go through any flaws in the project as well as potential adjustments that may be made if there had more time. Chapter 6 will be the project's conclusion it will briefly summarise the entire dissertation. Chapter 7 will be a reflection which will be a self-reflection on the project analysing the ethical, social, and legal elements in subsections. The last chapter will also provide an overview of the Project Management process, covering the tools and hazards involved.

# Literature review

## Introduction

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

Machine learning applications for abusive language detection can provide very accurate predicting models. 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 (See Background and Motivation).

## 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 as 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 the 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 a 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 on 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 the reduction of hate speech online and assisting law enforcers to 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 of a different race, behaviour, physical appearance, sexual orientation, class, gender, ethnicity, disability, and religion by helping a research body create 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 the 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 with 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 them 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 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 demonstrate 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 as 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 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 were 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 moods, and providing early-warning signs to better understand certain elements of 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 the 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 an 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.

According to Iftikhar, 2021 8 different machine learning methods were employed on a dataset comprising 14509 tweets and three classifications in the study by Abro et al., 2020. Hate speech was classified as class 0 in 16 percent of the tweets. Furthermore, 50% of tweets were classified as non-offensive, while the remaining 33% were classified as offensive but not hate speech. There is a significant imbalance between the training and testing instances of each class while training and testing that data (See Table 2 Hate speech detection classes and their instances).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Class** | **Total instances** | **Training instances** | **Testing instances** |
| 0 | Hate Speech | 2399 | 1909 | 490 |
| 1 | Non offensive | 7274 | 5815 | 1459 |
| 2 | Offensive but not hate speech | 4836 | 3883 | 953 |
|  | **Total** | **14509** | **1607** | **2902** |

Table 2 Hate speech detection classes and their instances

Naive Bayes, SVM, KNN, Decision Tree, Random Forest, AdaBoost, MLP, and Logistic Regression were among the algorithms employed in the study performed by Abro et al., 2020. SVM using TFIDF features representation and bigram features produced the highest recall 79 %, precision 77 %, accuracy 79% and F-measure in their project at 77 %. As shown in Figure 1 Confusion Matrix (Features: Bigram (TFIDF), Classifier: SVM) the work of (Abro et al., 2020), SVM employing TFIDF features representation with bigram features yielded class 0 with 155 correct and 335 incorrect predictions, class 1 with 1427 correct and 32 incorrect predictions, and class 3 with 698 correct and 255 incorrect predictions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| true label | 0 | 155 | 54 | 281 |
| 1 | 5 | 1427 | 27 |
| 2 | 122 | 133 | 698 |
|  | 0 | 1 | 2 |
| predicted label | | | | |

Figure 1 Confusion Matrix (Features: Bigram (TFIDF), Classifier: SVM)

Malik et al., 2021 in their research used Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Stacking, Bagging, Multi-Layer Perceptron, Convolutional Neural Networks and LSTM algorithms on an unbalanced dataset like the one used in this research. They performed fastText embedding and BERT embedding. In their research, they concluded that the best results obtained were using Convolutional Neural Networks with an 81 % F1-score.

Anjum & Katarya, 2021 applied logistic regression, support vector machine (SVM), Multinomial Naïve Bayes and random forest (RF) algorithms. They had three labels in their dataset hate, non-hate and offensive. They used TFIDF, TFIDF on POS tags, and Sentence Embeddings representations. They used 3 unbalanced Twitter datasets and balanced them using oversampling in their research. The best results they obtained were 91 % F1-score using SVM (Linear SVC) and TFIDF representation. They do not make mention the type of distribution used in their research unigram bigram or trigram.

Chakravartula, 2019 used a dataset comprised of hate speech, target, and aggressiveness labels. Bag of words (BoW), Word Embeddings and Sentence Embeddings representations are used in their research. They used Logistic Regression, SVM, XGBoost and Multinomial Naive Bayes classifier as predictive algorithms. They concluded that the Multinomial Naive Bayes classifier in combination with Bag of words representation gave the best result of a 73 % F1 score.

Warmsley, 2017 in their research when cross-evaluating different algorithms on a multi-classification Twitter dataset were able to obtain a precision of 91 %, recall of 90 %, and F1 score of 90 % using the Logistic regression algorithm. They had three labels hate, offensive and neither. Neither was classified correctly at 95 %, an offensive was classified at 91 % time correctly and hate was classified at 61 % times correctly. This shows that a high F1 score can be achieved which doesn’t mean that the result won’t be the outcome of a bise towards one or two classes. They used a bag of word embedding. The algorithms used in this part of their research were Logistic Regression, Naive Bayes, Decision Trees, Random Forests, and Linear Support Vector Machines.

The data from Twitter could be in the form of a binary or multiclass classification problem. The data used in this study will be binary classified. In current studies Naive 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 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 of 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 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 do not focus on balance datasets 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.

# Research Methodology

## Dataset

When choosing the datasets for this study, the following criteria were 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.

According to Iftikhar, 2021 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, label 0 denotes non-toxic tweets, and 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 equal representation.

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 the ethical implications of using the dataset in mind. Sensitive information on people such as their name and location is 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 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. Naive 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 to the pre-processed data as well. Using auto-sklearn 3 best-performing algorithms are chosen and are further evaluated.

### Naïve Bayes (NB) or Gaussian 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 the 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 (See Figure 2 Decision Tree) (Song & Ying 2015).

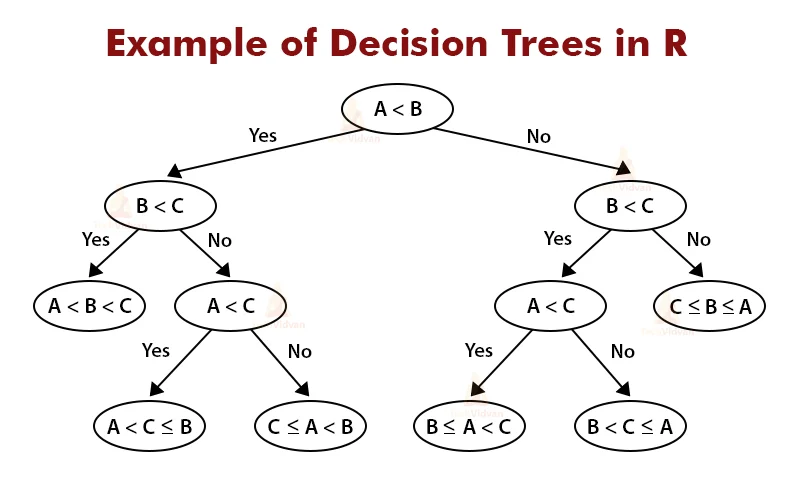


Figure 2 Decision Tree

### 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 SDG ("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

ETC (extra tree classifier) uses the meta-estimator, which increases prediction accuracy by training/fitting a large number of weak learners (randomised decision trees) on different samples of the dataset. It is also an ensemble learning model for classification purposes, similar to RF (random forest), hence it is compared to RF. The sole difference between ETC and RF is how the trees in the forest are built. ETC creates decision trees using the original training sample, whereas RF uses bootstrap samples from the original dataset to create decision trees. Each tree is given a random sample of k features from the feature-set at each test node. Based on some mathematical criteria, each decision tree must choose the optimal feature to partition the data (typically Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees (Rustam et al., 2021).

### Gradient Boosting

GB (gradient boosting) constructs an additive model in a stage-by-stage manner, allowing for the optimization of any differentiable loss function. The negative gradient of the binomial or multinomial deviance loss function is used to fit n classes\_ regression trees in each stage. A specific instance of binary classification is when just one regression tree is produced ("sklearn.ensemble.GradientBoostingClassifier", 2022).

### k Nearest Neighbours

In many circumstances, K-Nearest-Neighbors (KNN) is a non-parametric supervised classification approach that is simple yet effective. Due to its effective performance, efficient results, and simplicity, the KNN classifier is the most used classifier for pattern recognition. Pattern recognition, machine learning, text classification, data mining, object identification, and many more fields use it (Hota & Pathak, 2018).

The KNN algorithm classifies unknown data points by analogy, i.e. by comparing them to comparable training data points. Euclidean distance is used to determine similarity. The attribute values are adjusted to avoid bigger range characteristics from outweighing smaller range ones. The unknown pattern is assigned the most prevalent class among the classes of its nearest neighbours in the KNN classification. If there is a tie for the pattern between two classes, the class with the shortest average distance to the unknown pattern is allocated. A global distance function dist can be calculated by combining several local distance functions depending on specific parameters. The easiest approach to adding up the values is as shown in the equation (Hota & Pathak, 2018):

dist. (XT , X) = ∑n i=1 distAi (XT∙Ai , X∙Ai)

Where X T denotes the test tuple, X denotes the closest neighbour, and Ai (i=one to n) denotes the data points' characteristics. Global distance is the weighted sum of local distances. Specific weights wi can be provided to the characteristics Ai to indicate their relevance in determining the proper classes for the samples. The weights are generally between 0 and 1. Irrelevant characteristics weight zero. The mentioned equation can be modified to get the following equation (Hota & Pathak, 2018):

dist(XT,X) = ∑n i=1 wi × distAi (XT ∙Ai , X∙Ai )

The average weighted distance is obtained using the previous equation and is as follows (Hota & Pathak, 2018):

avgdist(XT ,X) = ∑n i=1 wi × distAi (XT ∙Ai , X∙Ai )

∑ n i=1 wi

### Liblinear SVC

LinearSVC. LinearSVC is a supervised machine learning approach for classifying, predicting, and detecting models. LinearSVC is based on linear kernels, but the model is implemented using LIBLINEAR rather than LIBSVM; it allows for greater flexibility in terms of penalties and loss functions, and it should scale well to huge numbers of data. The one-vs-rest approach is implemented by LinearSVC (Elbagir & Yang, 2018).

### Libsvm SVC

Support Vector Classification (SVC) is a LIBSVM-based implementation of SVC. SVC's probability model is built through cross-validation, which allows the results to differ from those produced using prediction. Furthermore, in extremely limited datasets, it will generate useless findings. The one-vs-one approach is utilised to achieve SVC multiclass support; SVC is used to classify linear and nonlinear data (Elbagir & Yang, 2018).

### MLP

A feedforward artificial neural network called a multilayer perceptron (MLP) translates a collection of input vectors to a set of output vectors. It is the most user-friendly and straightforward deep neural network (Qiao et al., 2020).

### Multinomial NB

The application of the Naive Bayes (NB) method was increased with this multinomial naive Bayes classifier. It may be used to classify discrete features in multinomial distributed data using NB multinomially. Integer feature counts are required for the multinomial distribution; however, fractional counts, such as the TF-IDF, perform well in reality (Elbagir & Yang, 2018).

### Passive Aggressive

Passive Aggressive Algorithms is a classification and regression algorithm. According to Nikam & Dalvi, 2020 Crammer et al. suggested a family of online learning algorithms. If v is the classifier's input feature vector, then the following Eq gives the final score (Nikam & Dalvi, 2020).

s = f(vw) = 𝑓(∑𝑖 𝑤𝑖𝑣𝑖)

The function f returns the desired output of two vectors, where w is the weight of a feature vector and f is the weight of a feature vector (Nikam & Dalvi, 2020).

### QDA

QDA stands for Quadratic Discriminant Analysis. A quadratic decision boundary classifier is created by fitting class conditional densities to data and using Bayes' rule ("sklearn.qda.QDA — scikit-learn 0.15-git documentation", 2022).

### 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

Both classification and regression issues are solved using RF (random forest). The RF model is a bagging-based ensemble model. It creates several trees and then votes among them to get a consensus. The accuracy of the prediction improves as the number of trees grows. By employing a bootstrap sampling strategy, RF mitigates the problem of over-fitting (Rustam et al., 2021).

### SGD

SGD is an acronym for stochastic gradient descent. It is a powerful and efficient method for learning classifiers that falls under the category of convex loss functions. SGD provides multi-class classification because it integrates several binary classifiers in the OvA (one-vs-all),   The loss gradient estimates each sample independently, and this estimator uses regularised linear models. A diminishing strength schedule is used to update the model. The regularizer adds the squared Euclidean norm L2, the absolute Euclidean norm L1, or a mix of both to the loss function that decreases model parameters toward the zero vector. SGD has been used to solve large-scale, sparse machine learning issues that are common in text categorization and natural language processing (Elbagir & Yang, 2018).

## Pre-processing and 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).

The mentioned pre-processing techniques are commonly used in binary text classification on similar problems. It can be observed that the mentioned techniques were also used by Elbagir & Yang, 2018, Valle-Cruz et al., 2020, Fehn Unsvåg & Gambäck, 2018, and Anandarajan et al., 2018 just to name a few hence they were considered a reliable and creditable method of data pre-processing to be used in this research.

A 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).

# Experiments, Evaluation and Results

## Experimental design

In this research, the performance of multiple models trained using various techniques, and 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.

The words to be removed in from the dataset are explained in the pre-processing section this was done following practices performed in other research like this one (See Pre-processing). There will also be a random removal of tweets from the Twitter dataset because of the hardware of the machine used in this research and the time constraints of this research the detailed reasoning for which can be found in the experimental result section (See Experimental 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 comparable between the source papers and this study because of 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 3 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 3 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 4 True positive rate).

Figure 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 5 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 6 Confusion matrix).

Figure False positive rate

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

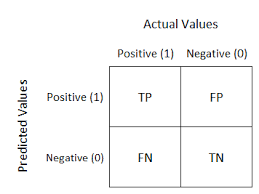


Figure 6 Confusion matrix

## Experimental results

In binary classification applications such as review helpfulness classification, the existence of numerous unlabelled examples and the incompleteness of positive labels are prevalent difficulties. According to prior research in the classification field, all unlabelled occurrences are considered negative examples. A classification model that learns to categorise binary cases with partial positive labels while presuming all unlabelled data to be negative examples, on the other hand, is likely to produce a biased classifier (Wang et al., 2020).

This study focuses on having a balance between positive and negative tweets. The machine used in this research had memory constraints and the research had time constraints. Thus, for this study 1000 labelled tweets were selected to be used 500 being negative tweets, 500 being positive randomly selected from the whole dataset. Specifically, when auto-sklearn was applied to the data 1000 samples were time costly. But initially, 1000, 2000, and 2500 samples were tested using Naive Bayes and Decision Tree Models with both bag of words and TFIDF representations using unigram, bigram, and trigram distributions. This was done to demonstrate the results of varying sample distribution sizes. The algorithms used in this test were selected based on the previous research by Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 which proved that Naive Bayes and Decision Tree Models can give very good F1 scores on a labelled Twitter dataset. The dataset is different from the one used in their experiment and the number of samples is fewer but this experiment is done to observe if the results from their experiment can be applied to other labelled Twitter datasets.

### Decision Tree and Naive Bayes default models on 1000 samples

Naive Bayes outperforms the decision tree model using a 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 the Naive Bayes algorithm giving 79 % F1 score and 79 % AUC (See Table 3 Results default DT VS NB) when using 1000 samples.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | N-grams | Representation | Test F1 score | AUC |
| Naive Bayes | unigram | BOW | 79 % | 79 % |
| Naive Bayes | bigram | BOW | 54 % | 61 % |
| Naive Bayes | trigram | BOW | 42 % | 55 % |
|  | | | | |
| Naive Bayes | unigram | TFIDF | 79 % | 79 % |
| Naive Bayes | bigram | TFIDF | 54 % | 61 % |
| Naive 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 % |

Table 3 Results default DT VS NB

Observing the results from the Naive 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 are higher on the bag of words representation by 1 % (See Figure 8 default NB BOW CM and Figure 7 default NB TFIDF CM). The model with the bag of word representation has classified 38 % non-toxic tweets and 42 % toxic tweets (See Figure 8 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 7 default NB TFIDF CM).

Naive Bayes

Naive Bayes

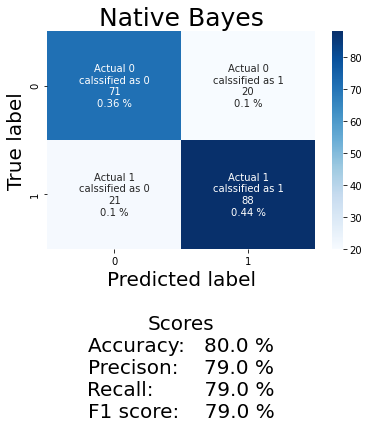


Figure default NB TFIDF CM

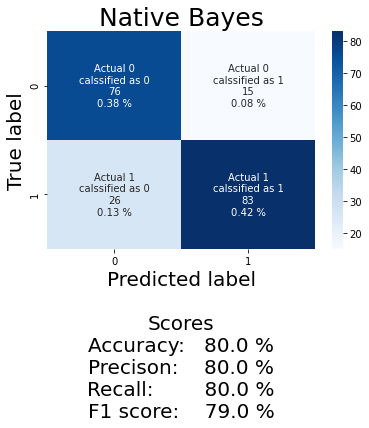


Figure default NB BOW CM

As this is a rounded result 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 10 default NB BOW AUC) performs better than the TFIDF represented model with an AUC of 0.7937796148805323 (See Figure 9 default NB TFIDF AUC) by 0.004536747656013862 % greater AUC.

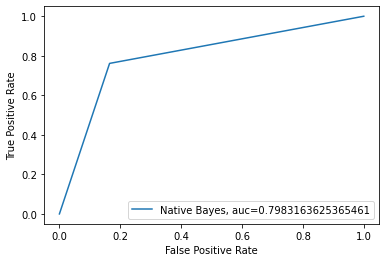
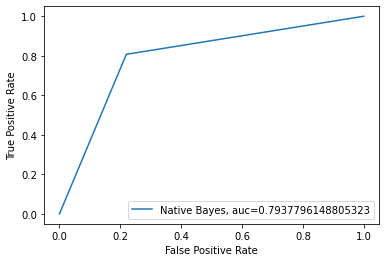


Figure default NB TFIDF AUC

Figure 10 default NB BOW AUC

### Decision Tree and Naive Bayes default models on 2000 samples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | N-grams | Representation | Test F1 score | AUC |
| Naive Bayes | unigram | BOW | 84 % | 83 % |
| Naive Bayes | bigram | BOW | 66 % | 68 % |
| Naive Bayes | trigram | BOW | 56 % | 62 % |
|  | | | | |
| Naive Bayes | unigram | TFIDF | 83 % | 83 % |
| Naive Bayes | bigram | TFIDF | 66 % | 68 % |
| Naive 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 samples results are observed to determine the linear improvement of both algorithms in 2000. Naive Bayes gives the best results with an F1 score of 84 % and AUC of 83 % using a bag of words model representation and a unigram distribution (See Table 4 Results default DT VS NB 2).

Table 4 Results default DT VS NB 2

Comparing results with 1000 samples and 2000 samples it can be observed that given 1000 samples the Naive Bayes algorithm has roughly around 10 % improvement with a unigram distribution, has 22 % improvement 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 4 Results default DT VS NB 2 and Table 3 Results default DT VS NB).

Now comparing the results with 1000 samples and 2000 samples 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 a 15 % improvement, the bigram distribution has 44 % and the trigram distribution has 25 % performance improvement (See Table 4 Results default DT VS NB 2 and Table 3 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 11 Improvement formula). Currently, the decision tree model using bigram distribution has the most overall improvement. If the improvement was linear then given enough samples the decision tree model using bigrams would have the best overall performance without hyperparameter optimization.

Figure Improvement formula

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

### Decision Tree and Naive Bayes default models on 2500 samples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | N-grams | Representation | Test F1 score | AUC |
| Naive Bayes | unigram | BOW | 79 % | 79 % |
| Naive Bayes | bigram | BOW | 68 % | 70 % |
| Naive Bayes | trigram | BOW | 55 % | 61 % |
|  | | | | |
| Naive Bayes | unigram | TFIDF | 79 % | 79 % |
| Naive Bayes | bigram | TFIDF | 68 % | 70 % |
| Naive 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 samples by 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 sample set and decision tree algorithm with the trigram distribution and TFIDF model representation has an improvement of about roughly 1 % from the 2000 sample set (See Table 4 Results default DT VS NB 2 and Table 5 Results default DT VS NB 3).

Table 5 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 samples 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 samples set to 1000

After applying auto-sklearn when max samples were set to 1000 (because of the 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 Bernoulli NB, SGD and LDA with a unigram distribution (See Table 6 results auto-sklearn).

When using the bag of words representation 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 the mentioned models had this accuracy when they were applied to 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 in both cases (See Table 6 results auto-sklearn).

When using the TFIDF representation SDG had 84 % accuracy, Bernoulli NB had an accuracy of 83 %, MLP had an accuracy of 80 %, LDA had an accuracy of 80 % and Passive-Aggressive had 80 % accuracy. All the mentioned models had this accuracy when they were applied to a unigram distribution. When models were applied on bigram and trigram distributions the accuracy was lower than 64% and 56% respectively which was obtained using liblinear\_svc and random\_forest in the mentioned order (See Table 6 results auto-sklearn).

|  |  |  |  |
| --- | --- | --- | --- |
| Bernoulli NB | unigram | BOW | 85 % |
| SDG | unigram | TFIDF | 84 % |
| Bernoulli NB | unigram | TFIDF | 83 % |
| LDA | unigram | BOW | 81 % |
| MLP | unigram | TFIDF | 80 % |
| LDA | unigram | TFIDF | 80 % |
| Passive Aggressive | unigram | TFIDF | 80 % |
| SGD | unigram | BOW | 78 % |
| Random Forest | unigram | BOW | 78 % |

Table 6 results auto-sklearn

### 

### Bernoulli NB default models on 1000 samples

Applying Bernoulli NB on 1000 tweets in combination with bag of words representation the results from the unigram distribution were 85 % F1 score and 0.8489395758303322 AUC. The results using a bigram distribution were 63 % F1 score and 0.6634653861544618 AUC. Using the trigram distribution 58 % F1 score and 0.5867346938775511 AUC were obtained. The same results were obtained using a TFIDF representation when applied with the same combination.

Observing the confusion matrix from both bag of words and TFIDF representations using the SDG algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.46 % and toxic tweets are classified correctly at 0.39 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 12 bag of words Bernoulli NB and Figure 14 TFIDF SDG).

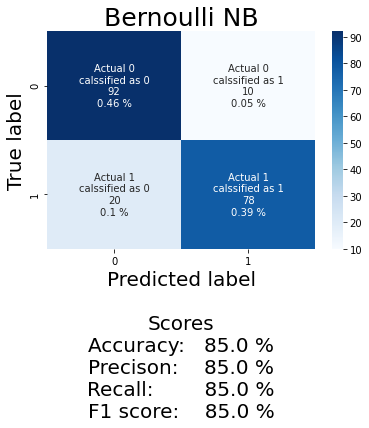
**



Figure bag of words Bernoulli NB

Figure TFIDF Bernoulli NB

### Bernoulli NB hyperparameter optimization models on 1000 samples

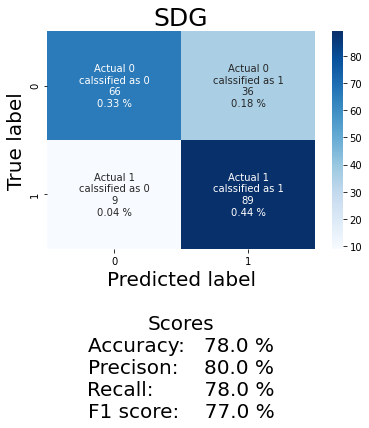
When performing hyperparameter optimization on Bernoulli NB using grid search no performance improvement was seen and hence the default model was used.

### SDG default models on 1000 samples

Applying SDG on 1000 tweets in combination with bag of words representation the results from the unigram distribution were 81 % F1 score and 0.8150260104041617 AUC. The results using a bigram distribution were 61 % F1 score and 0.6389555822328932 AUC. Using the trigram distribution 47 % F1 score, and 0.5614245698279312 AUC were obtained.

Using TDIDF representation on 1000 tweets in combination with SGD algorithms the results from the unigram distribution were 77 % F1 score and 0.7776110444177671 AUC. The results using a bigram distribution were 63 % F1 score and 0.6593637454981992 AUC. Using the trigram distribution 44 % F1 score, and 0.5510204081632653 AUC were obtained.

Observing the confusion matrix from both bag of words and TFIDF representations using the SDG algorithms and a unigram distribution it can be seen that when using bag of words representation all the non-toxic tweets are classified correctly and 0.18 % of toxic tweets are classified correctly (See Figure 15 bag of words SDG) in contrast when using a TFIDF representation 0.48 % toxic tweets are classified correctly and only 0.1 % non-toxic tweets are classified correctly (See Figure 14 TFIDF SDG). Both the mentioned results have a high bias towards toxic or non-toxic tweets hence SDG is not a reliable algorithm for word classification in this example (See Figure 15 bag of words SDG and Figure 14 TFIDF SDG).



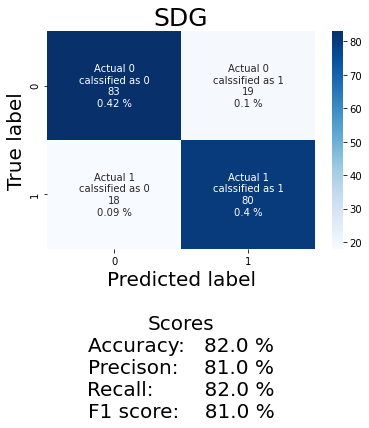
****

Figure TFIDF SDG

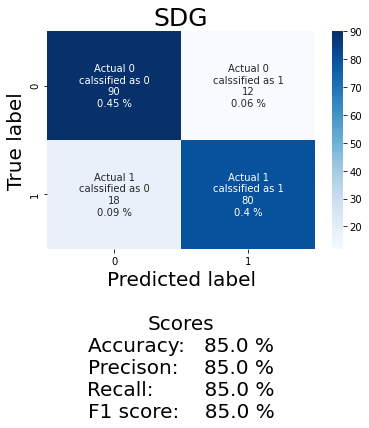
Figure bag of words SDG

### SDG hyperparameter optimization models on 1000 samples

The hyperparameter optimization was obtained using a random search. Applying SDG with the hyperparameter loss set to 'log', penalty set to 'elasticnet', tol set to 0.0003030218719035368, eta0 set to 0.005874526878891886, l1\_ratio set to 0.13337152523182383 and power\_t set to 0.7729065744463737 on 1000 tweets in combination with bag of words representation the results from the unigram distribution were 83 % F1 score and 0.8305322128851541 AUC.

When the hyperparameter loss is set to 'log', penalty is set to 'elasticnet', tol is set to 0.0003030218719035368, eta0 is set to 0.005874526878891886, l1\_ratio set to 0.13337152523182383 and power\_t is set to 0.5729065744463737 with TFIDF representation the SDG algorithm with a unigram distribution gave 84 % F1 score and 0.839935974389756 AUC.

Observing the confusion matrix from both bag of words and TFIDF representations using the SDG algorithms with hyperparameters tuned and a unigram distribution it can be seen that when using bag of words representation 0.45 % of non-toxic tweets are classified correctly and 0.4 % of toxic tweets are classified correctly (See Figure 16 bag of words SDG 2). When using a TFIDF representation 0.43 % of non-toxic tweets are classified correctly and only 0.41 % of toxic tweets are classified correctly (See Figure 17 TFIDF SDG 2). Both the mentioned results are distributed relatively equally hence SDG classifier with hyperparameter optimization applied is a reliable algorithm for word classification in this example (See Figure 16 bag of words SDG 2 and Figure 17 TFIDF SDG 2).



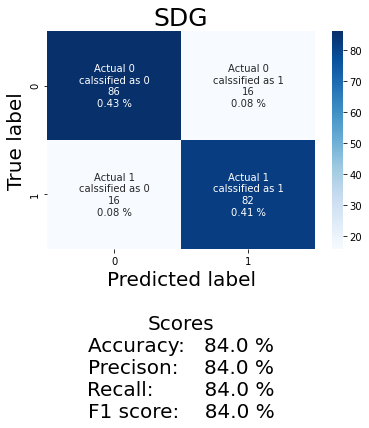


Figure bag of words SDG 2

Figure TFIDF SDG 2

### LDA default models on 1000 samples

Applying LDA on 1000 tweets in combination with bag of words representation the results from the unigram distribution were 69 % F1 score and 0.6979791916766708 AUC. The results using a bigram distribution were 38 % F1 score and 0.5194077631052422 AUC. Using the trigram distribution 49 % F1 score and 0.576530612244898 AUC were obtained.

Using TDIDF representation on 1000 tweets in combination with SGD algorithms the results from the unigram distribution were 62 % F1 score and 0.6489595838335334 AUC. The results using a bigram distribution were 42 % F1 score and 0.4867947178871549 AUC. Using the trigram distribution 49 % F1 score and 0.5718287314925969 AUC were obtained.

Observing the confusion matrix from both bag of words and TFIDF representations using the LDA algorithm and a unigram distribution it can be seen that when using bag of words representation 0.28 % of non-toxic tweets are classified correctly and 0.48 % of toxic tweets are classified correctly (See Figure 18 bag of words LDA). When using a TFIDF representation 0.48 % of non-toxic tweets are classified correctly and only 0.17 % of toxic tweets are classified correctly (See Figure 19 TFIDF LDA). Both the mentioned results have a high bias towards toxic or non-toxic tweets hence LDA is not a reliable algorithm for word classification in this example (See Figure 18 bag of words LDA and Figure 19 TFIDF LDA).

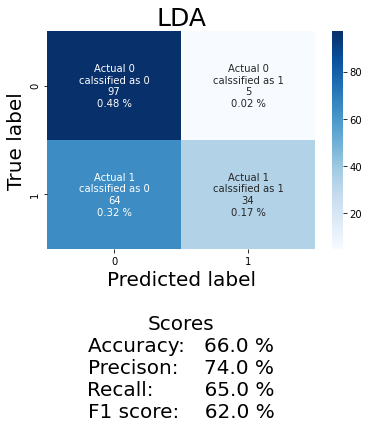


Figure bag of words LDA

### 

Figure TFIDF LDA

### LDA hyperparameter optimization models on 1000 samples

The hyperparameter optimization was obtained using random search. Applying LDA with the hyperparameter tol set to 1e-14, store\_covariance set to 'True', solver set to 'eigen', shrinkage set to 0.8 and n\_components set to None on 1000 tweets in combination with bag of words representation the results from the unigram distribution were 81 % F1 score and 0.809123649459784 AUC. Using the same combination with TFIDF representation gave an 85 % F1 score and 0.8497398959583834 AUC.

Observing the confusion matrix from both bag of words and TFIDF representations using the LDA algorithms with hyperparameters tuned and a unigram distribution it can be seen that when using bag of words representation 0.44 % of non-toxic tweets are classified correctly and 0.38 % of toxic tweets are classified correctly (See Figure 21 bag of words LDA 2). When using a TFIDF representation 0.44 % of non-toxic tweets are classified correctly and only 0.41 % of toxic tweets are classified correctly (See Figure 20 TFIDF LDA 2). Both the mentioned results are distributed relatively equally hence LDA classifier with hyperparameter optimization applied is a reliable algorithm for word classification in this example (See Figure 21 bag of words LDA 2 and Figure 20 TFIDF LDA 2).

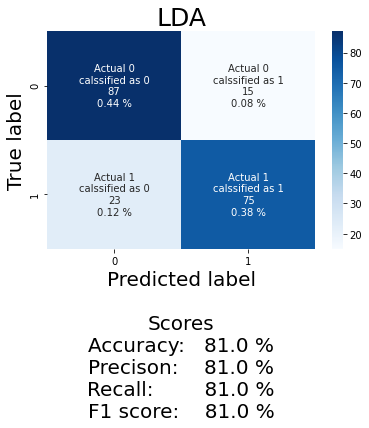
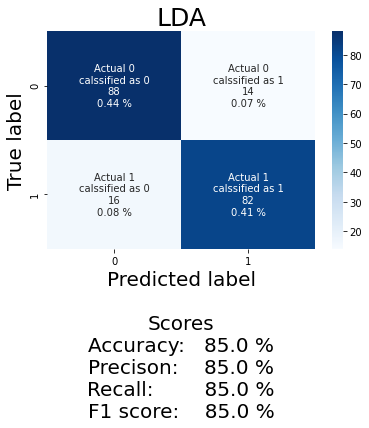


Figure TFIDF LDA 2

Figure 21 bag of words LDA 2

### Cross-validation

On 1000 tweets in combination with bag of words and TFIDF representation with a unigram distribution the Bernoulli NB default model (See Bernoulli NB default models on 1000 samples), LDA hyperparameter optimization model (See SDG hyperparameter optimization models on 1000 samples) and LDA hyperparameter optimization model (See LDA hyperparameter optimization models on 1000 samples) had the best F1 test scores. Their scores were relatively close to determining which algorithm with which model representation performs the best on the 1000 tweets when 10 cross-validations are performed.

#### Bernoulli NB cross validation

The Bernoulli NB classifier had a cross-validation score of 83.10 % with a standard deviation of 2.84 % using a bag of word and TFIDF representation.

#### SDG cross-validation

The SDG classifier with the hyperparameter loss set to 'log', penalty set to 'elasticnet', tol set to 0.0003030218719035368, eta0 set to 0.005874526878891886, l1\_ratio set to 0.13337152523182383 and power\_t set to 0.7729065744463737 had a cross-validation score of 80.00 % with a standard deviation of 3.63 % using a bag of word representation. The same classifier with the hyperparameter loss is set to 'log', penalty is set to 'elasticnet', tol is set to 0.0003030218719035368, eta0 is set to 0.005874526878891886, l1\_ratio set to 0.13337152523182383 and power\_t is set to 0.5729065744463737 had a cross-validation score of 80.70 % with a standard deviation of 2.83 % using a TFIDF representation.

#### LDA cross-validation

The LDA classifier with the hyperparameter tol set to 1e-14, store\_covariance set to 'True', solver set to 'eigen', shrinkage set to 0.8 and n\_components set to None had a cross-validation score of 81.20 % with a standard deviation of 2.52 % using a bag of word representation. Using the same hypermeters, the LDA model had a cross-validation score of 81.90 % and a standard deviation of 2.66 % using a TFIDF representation.

#### Cross-validation results evaluation

The best results are obtained by Bernoulli NB with a cross-validation score of 83.10 % and a standard deviation of 2.84 % on both bag of words and TFIDF models. The second-best score is by LDA with TFIDF representation with a score of 81.90 % and a standard deviation of 2.66 %. The third-best score is obtained by using LDA with a bag of word representation with a score of 81.20 % and a standard deviation of 2.52 %. The fourth-best score was obtained by using SDG with a cross-validation score of 80.70 % and a standard deviation of 2.83 using a TFIDF representation. SDG performs the worst with a cross-validation score of 80.00 % and a standard deviation of 3.63 using a bag of words representation (See Table 7 Cross-Validation results).

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Representation | cross validation score | Standard deviation |
| Bernoulli NB | BOW | 83.10 % | 2.84 % |
| Bernoulli NB | TFIDF | 83.10 % | 2.84 % |
|  | | | |
| LDA | BOW | 81.20 % | 2.52 % |
| LDA | TFIDF | 81.90 % | 2.66 % |
|  | | | |
| SGD | BOW | 80.00 % | 3.63 % |
| SGD | TFIDF | 80.70 % | 2.83 % |

Table 7 Cross-Validation results

### Removing least frequent words removed by the percentage of their occurrence in the data frame

There is a total of 8762 words in the 1000 sample Tweets chosen. In this section least frequent words will randomly be removed by the percentage of their occurrence in the data frame. The meaning of the statement “least frequent words will randomly be removed by the percentage of their occurrence in the data frame” can be explained by compartmentalising the said statement.

The first compartment statement “least frequent words” can be easily explained by an example which is if the word bad appears in the data frame 10 times and the word ugly appears in a data frame 5 times then the word ugly can be concluded that the word ugly appears in the data frame fewer times then the word bad hence it has a lower frequency of appearance then the word bad and if the data frame only had two words bad and ugly then ugly would be the least frequent appearing words in the data frame.

The second compartment statement “removed by the percentage of their occurrence in the data frame” can be explained by simply taking a percent of words that occur the least number of times in a data frame and removing them from the said data frame. For example, the total number of words in the data frame is 8762. 5 % of these words would be 438.1 words. Thus, any words appearing less than the rounded number 438 words will be removed from the data frame.

The final compartment statement “randomly be removed by percentage of their occurrence in the data frame” can again be explained by an example which is if there were three words present in a data frame ugly, bad and hate. The words ugly, bad and hate all appeared in the data frame 5 times more than any row in the data frame with the words ugly, bad or hate could be removed at random. This is done until the conditions explained in the second compartment statement are met and the words removed to meet a certain selected percentage. But doing so could remove other words from the row as well for example if the sentence in a row of a data frame was, he is bad and troubled the by removing the row the words he is and troubled are also removed with the word bad. Thus, repeating the same experiment with the same percentage of words to be removed could have varying results on the data frame depending upon the words removed.

The mentioned is done to test what effect will take place when the least frequent words will randomly be removed by the percentage of their occurrence in the data frame. The words removed will be from a lower percentage to a higher percentage for example removing 5 % words than 10 % and so on. This is done to test what effect removing more words by frequency of their occurrence by percentage in a data frame has on the overall performance of a model.

The dataset obtained after removing the least frequent words will have bag of word and TFIDF representations applied to them using a unigram distribution and a Bernoulli NB algorithm will be used to access the results from these sets. Only Bernoulli NB will be applied to the mentioned set because of the time constrictions of this research and it achieved the best performance when it was evaluated against other algorithms (See Auto-sklearn with max samples set to 1000 and Cross-validation results evaluation). As the least frequent words are removed at random repeating the experiment should give varying results.

#### Removing 5 % of least frequent words

There is a total of 8762 words in the 1000 tweets samples chosen. After removing 5 % of the most frequently occurring words. The Bernoulli NB model has an 84 % F1 score and 0.8504889605807038 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.4 % and toxic tweets are classified correctly at 0.44 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 22 bag of words and TFIDF Bernoulli NB).

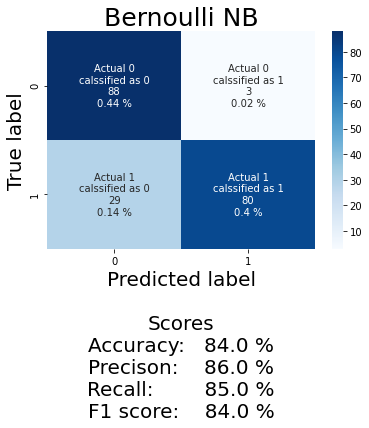


Figure bag of words and TFIDF Bernoulli NB

#### Removing 10 % least frequent words

After removing 10 % of the most frequently occurring words. The Bernoulli NB model has an 84 % F1 score and 0.8424941795728313 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.46 % and toxic tweets are classified correctly at 0.38 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 23 bag of words and TFIDF Bernoulli NB 2).

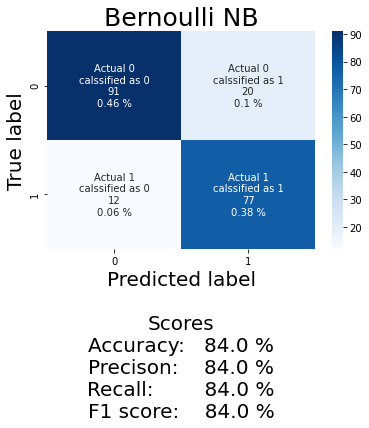


Figure 23 bag of words and TFIDF Bernoulli NB 2

#### Removing 15 % least frequent words

After removing 15 % of the most frequently occurring words. The Bernoulli NB model has an 84 % F1 score and 0.844437775110044 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.44 % and toxic tweets are classified correctly at 0.4 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 24 bag of words and TFIDF Bernoulli NB 3).

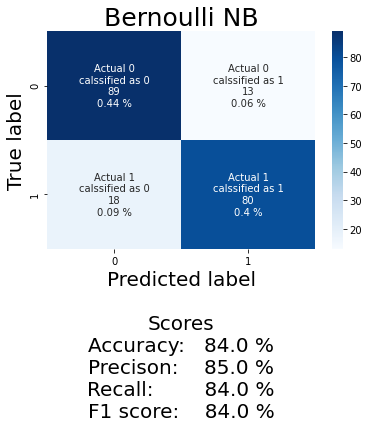


Figure 24 bag of words and TFIDF Bernoulli NB 3

#### Removing 20 % least frequent words

After removing 20 % of the most frequently occurring words. The Bernoulli NB model has an 85 % F1 score and 0.8624357294082065 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.43 % and toxic tweets are classified correctly at 0.42 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 25 bag of words and TFIDF Bernoulli NB 4).

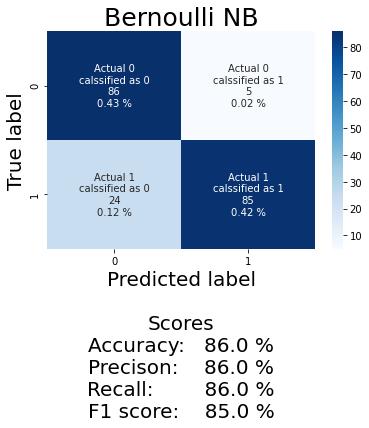


Figure 25 bag of words and TFIDF Bernoulli NB 4

#### Removing 25 % least frequent words

After removing 25 % of the least frequently occurring words. The Bernoulli NB model has an 88 % F1 score and 0.8859473526173556 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.44 % and toxic tweets are classified correctly at 0.44 %. As the classification of toxic and non-toxic tweets correctly are balanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 26 bag of words and TFIDF Bernoulli NB 4).

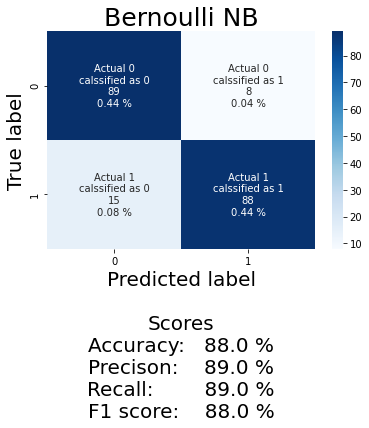


Figure 26 bag of words and TFIDF Bernoulli NB 4

#### Removing 30 % least frequent words

After removing 30 % of the most frequently occurring words. The Bernoulli NB model has an 82 % F1 score and 0.8208020050125313 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.48 % and toxic tweets are classified correctly at 0.35 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 27 bag of words and TFIDF Bernoulli NB 5).



Figure 27 bag of words and TFIDF Bernoulli NB 5

#### Removing 40 % least frequent words

After removing 40 % of the most frequently occurring words. The Bernoulli NB model has an 83 % F1 score and 0.8386430083677789 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.4 % and toxic tweets are classified correctly at 0.44 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 28 bag of words and TFIDF Bernoulli NB 6).



Figure bag of words and TFIDF Bernoulli NB 6

#### Removing 56 % least frequent words

After removing 56 % of the most frequently occurring words. The Bernoulli NB model has a 74 % F1 score and 0.7452206986287659 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.38 % and toxic tweets are classified correctly at 0.36 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 29 bag of words and TFIDF Bernoulli NB 7).



Figure 29 bag of words and TFIDF Bernoulli NB 7

#### 

#### Removing least frequent words by percentage evaluation

Bernoulli NB performance keeps on increasing even when the number of words decreases until 56 % of most frequent words are removed from the 1000 rows dataset that’s when the performance of Bernoulli NB algorithms decreases significantly (See Removing 56 % least frequent words). When 25 % of words are removed the best result is obtained at 88 % F1 score and 0.8859473526173556 AUC.

|  |  |  |  |
| --- | --- | --- | --- |
| % Words removed | % Words left in the dataset | F1 score | AUC |
| 5 | 95 | 84 % | 0.8504889605807038 |
| 10 | 90 | 84 % | 0.8424941795728313 |
| 15 | 85 | 84 % | 0.844437775110044 |
| 20 | 80 | 85 % | 0.8624357294082065 |
| 25 | 75 | 88 % | 0.8859473526173556 |
| 30 | 70 | 82 % | 0.8208020050125313 |
| 40 | 60 | 83 % | 0.8386430083677789 |
| 56 | 44 | 74 % | 0.7452206986287659 |

Table 8 bag of words and TFIDF Bernoulli NB

The words are removed at random so the same results cannot be obtained by just removing 25 % least frequent words to prove this the same experiment was repeated by removing the 25 % of least frequent words and the Bernoulli NB algorithms were applied to this dataset with the same conditions as the previous experiment (See Removing 25 % least frequent words and Figure 30 bag of words and TFIDF Bernoulli NB 8).

Removing 25 % of random words again from the dataset with 1000 tweets samples chosen this time the F1 score is 83 % and AUC is 0.8348370927318295.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.44 % and toxic tweets are classified correctly at 0.4 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 30 bag of words and TFIDF Bernoulli NB 8).

Thus, by observing the results from the two experiments two things can be concluded: (1) Removing approximately 5 % – 50 % least frequent words can result in an increase or decrease in a model’s performance and (2) the ratio of words removed does not necessarily increase the model’s performance.

Observing the results from the first confusion matrix and comparing them with this one it can also be observed that in the first instance when the 25 % words were randomly removed the classification of toxic and non-toxic tweets were the same (See Figure 26 bag of words and TFIDF Bernoulli NB 4) but in this case, the results are slightly different (See Figure 30 bag of words and TFIDF Bernoulli NB 8).

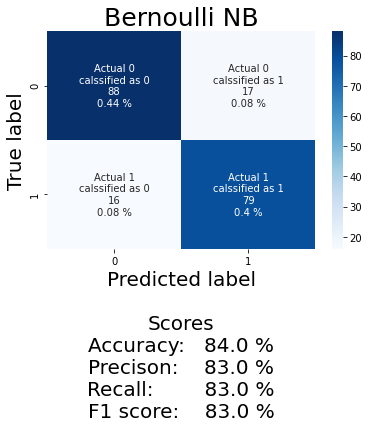


Figure 30 bag of words and TFIDF Bernoulli NB 8

### Removing the least frequent words by the number of their occurrence in the data frame

There is a total of 8762 words in the 1000 tweets samples chosen. In this section words will be removed by the number of times they occur in a sentence the number will be specified in steps and the frequency of words will be in ascending order starting from 1. The dataset obtained after removing the least occurring words will have bag of word and TFIDF representations applied to them using a unigram distribution and a Bernoulli NB algorithm will be used to access the results from these sets. Only Bernoulli NB will be applied on the mentioned set because of the time constrictions of this research and it achieved the best performance when it was evaluated against other algorithms (See Auto-sklearn with max samples set to 1000 and Cross-validation results evaluation). As the least frequent words are not removed at random hence this experiment if repeated will give the same results if all other conditions are the same as the conditions in this experiment.

This experiment is conducted to see the effect of removing the least frequent words by their number of occurrences in a data frame. This is done as an extra step to validate that when least frequent words will randomly be removed by the percentage of their occurrence in the data frame it is specific words in the sentence or data fame row that enhance the F1 score of a model. If the models' performance could simply be enhanced optimally by removing the least frequent words by the number of their occurrence in a data frame then it could be proved that the words that have the most positive effect on a models performance by removal are the least frequent words by the number of there occurrence in a data frame or least frequent words randomly be removed by the percentage of their occurrence in the data frame.

#### Removing words if they occur in a sentence more than 1 time (about 45 % of data removed)

After removing words that only occur more than 1 time. The Bernoulli NB model has an 84 % F1 score and 0.8429966432712847 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.48 % and toxic tweets are classified correctly at 0.36 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 31 bag of words and TFIDF Bernoulli NB 9Figure 29 bag of words and TFIDF Bernoulli NB 7).

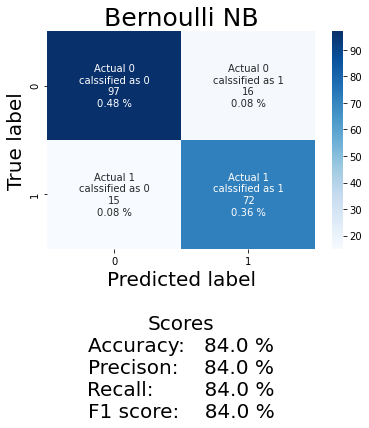


Figure 31 bag of words and TFIDF Bernoulli NB 9

#### Removing words if they occur in a sentence more than 2 times (about 43 % of data removed)

After removing words that only occur more than 2 times. The Bernoulli NB model has an 81 % F1 score and 0.81328320802005 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.44 % and toxic tweets are classified correctly at 0.37 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See

Figure 32 bag of words and TFIDF Bernoulli NB 10Figure 29 bag of words and TFIDF Bernoulli NB 7).

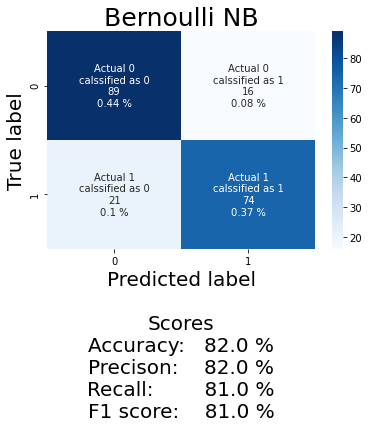


Figure 32 bag of words and TFIDF Bernoulli NB 10

#### Removing words if they occur in a sentence more than 3 times (about 50 % of data removed)

After removing words that only occur more than 3 times. The Bernoulli NB model has 81 % F1 score and 0.8071658615136876 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.36 % and toxic tweets are classified correctly at 0.46 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 33 bag of words and TFIDF Bernoulli NB 11Figure 29 bag of words and TFIDF Bernoulli NB 7).

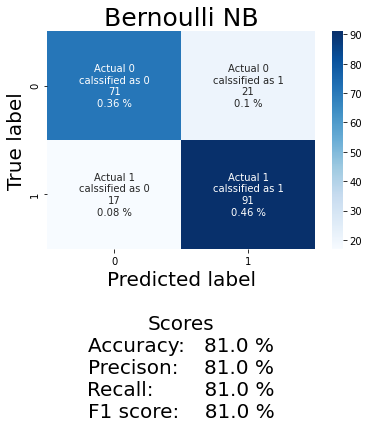


Figure 33 bag of words and TFIDF Bernoulli NB 11

#### Removing words if they occur in a sentence more than 4 times (about 55 % of data removed)

After removing words that only occur more than 4 times. The Bernoulli NB model has an 82 % F1 score and 0.8242753623188406 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.45 % and toxic tweets are classified correctly at 0.38 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 34 bag of words and TFIDF Bernoulli NB 12Figure 29 bag of words and TFIDF Bernoulli NB 7).

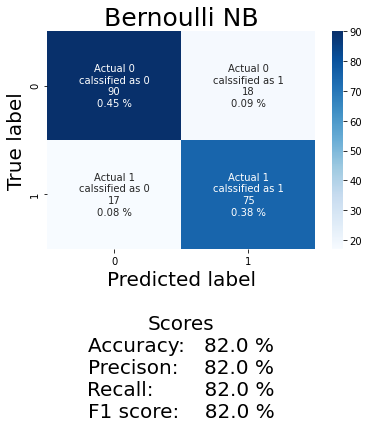


Figure 34 bag of words and TFIDF Bernoulli NB 12

#### Removing words if they occur in a sentence more than 5 times (about 59 % of data removed)

After removing words that only occur more than 5 times. The Bernoulli NB model has an 81 % F1 score and 0.8096955128205129 AUC using both bag of the word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.42 % and toxic tweets are classified correctly at 0.38 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 35 bag of words and TFIDF Bernoulli NB 13Figure 29 bag of words and TFIDF Bernoulli NB 7).

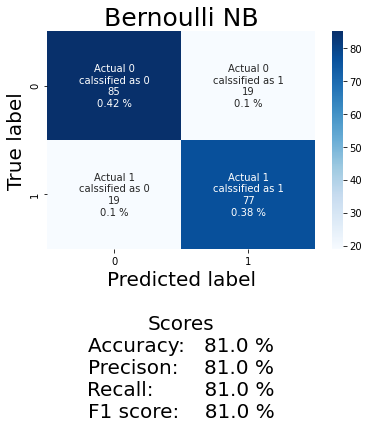


Figure 35 bag of words and TFIDF Bernoulli NB 13

#### Removing words if they occur in a sentence more than 6 times (about 62 % of data removed)

After removing words that only occur more than 6 times. The Bernoulli NB model has a 77 % F1 score and 0.7768429487179488 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.42 % and toxic tweets are classified correctly at 0.38 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 36 bag of words and TFIDF Bernoulli NB 14Figure 35 bag of words and TFIDF Bernoulli NB 13Figure 29 bag of words and TFIDF Bernoulli NB 7).



Figure 36 bag of words and TFIDF Bernoulli NB 14

#### Removing the least frequent words by their occurrence in a sentence evaluation

Bernoulli NB performance keeps on decreasing or increasing between 1 % – 4 % as the number of words decreases by their occurrence in a sentence (See Table 9 bag of words and TFIDF Bernoulli NB 2).

|  |  |  |  |
| --- | --- | --- | --- |
| word occurrence frequency per sentence | % Words removed | F1 score | AUC |
| 1 | 30 | 84 % | 0.8429966432712847 |
| 2 | 43 | 81 % | 0.81328320802005 |
| 3 | 50 | 81 % | 0.8071658615136876 |
| 4 | 55 | 82 % | 0.8242753623188406 |
| 5 | 59 | 81 % | 0.8096955128205129 |
| 6 | 62 | 77 % | 0.7768429487179488 |

Table 9 bag of words and TFIDF Bernoulli NB 2

### Removing the most frequent words by the percentage of their occurrence in the data frame

There is a total of 8762 words in the 1000 tweets samples chosen. In this section words will be removed by the number of times they occur in a sentence the number will be specified in steps and the frequency of words will be in ascending order starting from 1. The dataset obtained after removing the most occurring words will have bag of word and TFIDF representations applied to them using a unigram distribution and a Bernoulli NB algorithm will be used to access the results from these sets. Only Bernoulli NB will be applied on the mentioned set because of the time constrictions of this research and it achieved the best performance when it was evaluated against other algorithms (See Auto-sklearn with max samples set to 1000 and Cross-validation results evaluation). As the most frequent words are removed at random hence this experiment if repeated should give varying results.

This experiment is conducted to see the effect of removing the most frequent words by their number of occurrences in a data frame. This is done as an extra step to validate that when most frequent words will randomly be removed by the percentage of their occurrence in the data frame it is specific words in the sentence or data fame row that enhance the F1 score of a model. If the models' performance could simply be enhanced optimally by removing the least frequent words by the number of their occurrence in a data frame then it could be proved that the words that have the most positive effect on a models performance by removal are the most frequent words by the number of there occurrence in a data frame or least frequent words randomly be removed by the percentage of their occurrence in the data frame.

#### Removing the most frequent word (about 8 % data removed)

The most frequently occurring word is removed from the dataset. The most frequent word is ‘user’, and it occurs in the dataset 683 times. The Bernoulli NB model has an 85 % F1 score and 0.8478677286016736 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.48 % and toxic tweets are classified correctly at 0.38 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 37 bag of words and TFIDF Bernoulli NB 15Figure 29 bag of words and TFIDF Bernoulli NB 7).

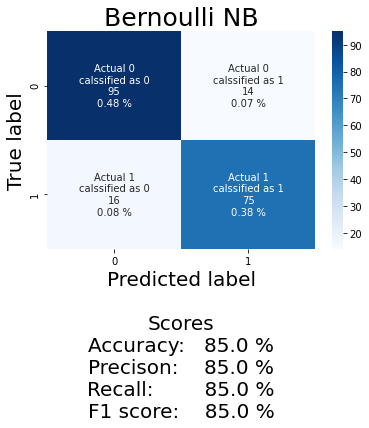


Figure 37 bag of words and TFIDF Bernoulli NB 15

#### Removing the 2nd most frequent word (about 9 % data removed)

The most frequently occurring word is removed from the dataset. The 2nd most frequent word is ‘amp’, and it occurs in the dataset 79 times. The Bernoulli NB model has an 81 % F1 score and 0.8075070252910478 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.45 % and toxic tweets are classified correctly at 0.36 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 38 bag of words and TFIDF Bernoulli NB 16Figure 29 bag of words and TFIDF Bernoulli NB 7).



Figure 38 bag of words and TFIDF Bernoulli NB 16

#### Removing the 14 most frequent words (about 13 % of data removed)

The most frequently occurring word is removed from the dataset. The 12 most frequent words from the dataset. The Bernoulli NB model has an 81 % F1 score and 0.8081091930951425 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.44 % and toxic tweets are classified correctly at 0.36 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 39 bag of words and TFIDF Bernoulli NB 17Figure 38 bag of words and TFIDF Bernoulli NB 16Figure 29 bag of words and TFIDF Bernoulli NB 7).



Figure 39 bag of words and TFIDF Bernoulli NB 17

#### Removing the 55 most frequent words (about 25 % of data removed)

The most frequently occurring word is removed from the dataset. The 55 most frequent words from the dataset. The Bernoulli NB model has a 74 % F1 score and 0.7516290726817043 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.42 % and toxic tweets are classified correctly at 0.32 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 40 bag of words and TFIDF Bernoulli NB 18Figure 38 bag of words and TFIDF Bernoulli NB 16Figure 29 bag of words and TFIDF Bernoulli NB 7).

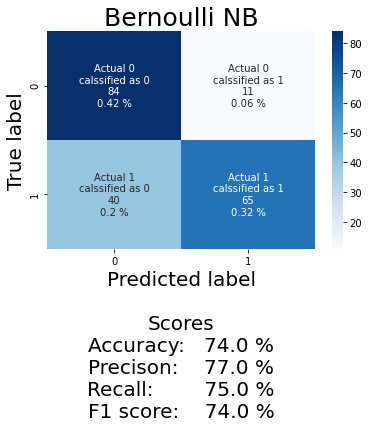


Figure 40 bag of words and TFIDF Bernoulli NB 18

#### Removing the 320 most frequent words (about 45 % of data removed)

The most frequently occurring word is removed from the dataset. The 320 most frequent words from the dataset. The Bernoulli NB model has a 71 % F1 score and 0.7136213621362135 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.43 % and toxic tweets are classified correctly at 0.28 %. As the classification of toxic and non-toxic tweets correctly is too unbalanced hence this algorithm is not a reliable algorithm for word classification in this example (See Figure 39 bag of words and TFIDF Bernoulli NB 17 Figure 41 bag of words and TFIDF Bernoulli NB 19Figure 38 bag of words and TFIDF Bernoulli NB 16Figure 29 bag of words and TFIDF Bernoulli NB 7).



Figure 41 bag of words and TFIDF Bernoulli NB 19

#### Removing the most frequent words evaluation

Bernoulli NB performance keeps on decreasing as the number of most frequent words decreases by their occurrence in a sentence except when 13 % of most frequent words are removed then the performance is improved slightly as AUC for when 13 % words are removed is 0.000602167804094722 higher than when 9 % words are removed (See Table 10 bag of words and TFIDF Bernoulli NB 3Table 9 bag of words and TFIDF Bernoulli NB 2).

|  |  |  |  |
| --- | --- | --- | --- |
| word occurrence frequency per sentence | % Words removed | F1 score | AUC |
| 1 | 8 | 85 % | 0.8478677286016736 |
| 2 | 9 | 81 % | 0.8075070252910478 |
| 12 | 13 | 81 % | 0.8081091930951425 |
| 55 | 25 | 74 % | 0.7516290726817043 |
| 320 | 45 | 71 % | 0.7136213621362135 |

Table 10 bag of words and TFIDF Bernoulli NB 3

But as the above results are obtained by removing the most frequent words at random hence if repeated the results obtained would be different.

Removing 25 % of random most frequent words again from the dataset with 1000 tweets samples chosen this time the F1 score is 82 % and AUC is 0.8262304921968787.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.44 % and toxic tweets are classified correctly at 0.390 %. As the classification of toxic and non-toxic tweets correctly is not too unbalanced hence this algorithm is a reliable algorithm for word classification in this example (See Figure 42 bag of words and TFIDF Bernoulli NB 20).

Thus, by observing the results from the two experiments two things can be concluded: (1) Removing approximately 8 % – 25 % most frequent words can result in an increase or decrease in a model’s performance and (2) the ratio of words removed does not necessarily increase the model’s performance.

Observing the results from the first confusion matrix and comparing them with this one it can also be observed that in the first instance when the 25 % words were randomly removed the correct classification of toxic is 44 % and non-toxic tweets are 39 % (See Figure 40 bag of words and TFIDF Bernoulli NB 18) but in this case, the results are slightly different (See Figure 42 bag of words and TFIDF Bernoulli NB 20).

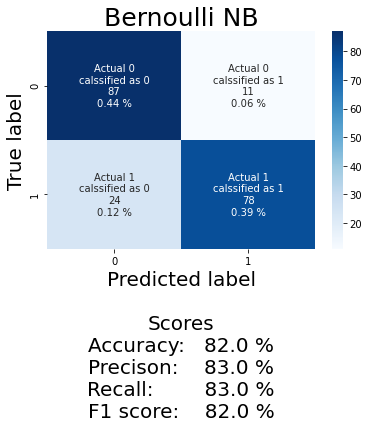
****

Figure 42 bag of words and TFIDF Bernoulli NB 20

### Removing words frequent or by their occurrence in a sentence cross-evaluation

Observing Bernoulli NB performance when least frequent words are removed by the percentage of their occurrence in the data frame (See

Removing least frequent words by percentage evaluation), most frequent words are removed by the percentage of their occurrence in the data frame (See Removing the most frequent words evaluation) and least most frequent words are removed by the number of their occurrence in the data frame (See Removing the least frequent words by their occurrence in a sentence evaluation).

The following observations can be made. (1) Removing both most frequent and least frequent words can increase or decrease a model’s performance. (2) The increase or decrease in model performance is entirely not because of the number of words removed and (3) The increase or decrease in model performance is caused by the kind of words removed (even if the number of words is the same and their occurrence by percentage or by their number is equal results will vary based on the words removed) (See Table 11 bag of words and TFIDF Bernoulli NB 4).

|  |  |  |  |
| --- | --- | --- | --- |
| Least frequent words by percent | | | |
| % Words removed | % Words left in the dataset | F1 score | AUC |
| 5 | 95 | 84 % | 0.8504889605807038 |
| 10 | 90 | 84 % | 0.8424941795728313 |
| 15 | 85 | 84 % | 0.844437775110044 |
| 25 | 75 | 88 % | 0.8859473526173556 |
| 25 | 75 | 83 % | 0.8348370927318295 |
| 56 | 44 | 74 % | 0.7452206986287659 |
|  | | | |
| Least frequent words by occurrence | | | |
| word occurrence by number per sentence | % Words removed | F1 score | AUC |
| 5 | 59 | 81 % | 0.8096955128205129 |
| 6 | 62 | 77 % | 0.7768429487179488 |
|  | | | |
| Most frequent words by percentage | | | |
| word occurrence by frequency per sentence | % Words removed | F1 score | AUC |
| 1 | 8 | 85 % | 0.8478677286016736 |
| 2 | 9 | 81 % | 0.8075070252910478 |
| 12 | 13 | 81 % | 0.8081091930951425 |
| 55 | 25 | 74 % | 0.7516290726817043 |
| 55 | 25 | 82 % | 0.8262304921968787 |
| 320 | 45 | 71 % | 0.7136213621362135 |

Table 11 bag of words and TFIDF Bernoulli NB 4

### Replacing words starting with not with their synonym or not\_

In this step of the research, an experiment was carried out all the words starting with not were either replaced with their synonyms or with not\_ this was done to test the performance of a machine learning algorithm's ability to detect natural human language correctly.

Replacing all the words starting with not with their synonyms or by not\_ from the data set to see if it increases model performance. The Bernoulli NB model has an 80 % F1 score and 0.798919027124412 AUC using both bag of word and TFIDF representations.

Observing the confusion matrix from both bag of words and TFIDF representations using the Bernoulli NB algorithms and a unigram distribution it can be seen that in both cases non-toxic tweets are classified correctly at 0.37 % and toxic tweets are classified correctly at 0.43 %. As the classification of toxic and non-toxic tweets correctly is too unbalanced hence this algorithm is not a reliable algorithm for word classification in this example (See Figure 43 bag of words and TFIDF Bernoulli NB 21Figure 41 bag of words and TFIDF Bernoulli NB 19Figure 38 bag of words and TFIDF Bernoulli NB 16Figure 29 bag of words and TFIDF Bernoulli NB 7).

The mentioned results are worse than when the same set of constraints was used on the model without the words starting with not being replaced with their synonyms or by not (See Bernoulli NB default models on 1000 samples). So, it can be concluded that Bernoulli NB is good at detecting the occurrence of the word not when combined with TFIDF or bag of word representation and unigram distribution.

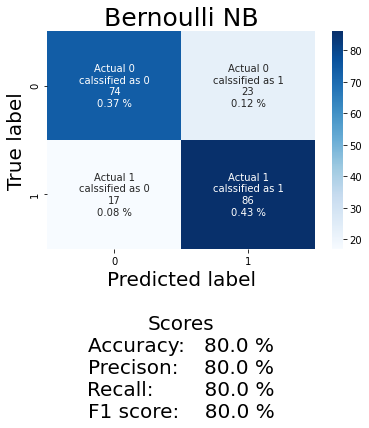


Figure 43 bag of words and TFIDF Bernoulli NB 21

### Cross Evaluation between the best results of this study and previous studies

Firstly, the research carried out by Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 has a very accuracy score. There is a possibility that the number of labels used in their experiment could be unbalanced and there could be a possibility that one label in their dataset could be highly lower than others. Hence in this research, an experiment was carried out using the same algorithms. As the pre-processing stage is different for this experiment than the pre-processing for the experiment of Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 thus the results will be different. This experiment may also use different embedding/representations than the experiment of Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021. This experiment was done to demonstrate if an algorithm performs well given a set of conditions for a dataset will it also perform as good given a different but similar dataset. The mentioned test could not be performed fully because of the lack of detail available in the experiments of Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 but using the data from this experiment future experiments could be carried out to test this observation.

Now Mohiyaddeen & Siddiqui, 2021 were able to achieve an accuracy of 93 % in their experiment using a Naive Bayes algorithm in this research the measure of algorithms performed was determined using the F1 score. The F1 score using Naive Bayes algorithm encoding using Bow or TFIDF representation and using a unigram distribution 79 % F1 score was achieved when the dataset had 1000 samples. This cannot be effectively compared to Mohiyaddeen & Siddiqui, 2021 attempt at using the same algorithm on a similar dataset (See Table 12 A comparison between Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 research and this research).

It was worth investigating if the increase in sample size increases model performance. As Mohiyaddeen & Siddiqui, 2021 dataset could have had more samples than the samples used in this experiment. The mentioned was attempted in this research. When 2000 samples were used in combination with the BOW representation and unigram distribution 84 % F1 score was achieved. When the same was done using TFIDF word representation 83 % F1 score was achieved (See Table 12 A comparison between Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 research and this research). In this instance, the performance of the model was improved but it could not be concluded from a single experiment that an increase in samples increases model performance.

When 2500 samples were used in combination with the BOW representation and unigram distribution 79 % F1 score was achieved. When the same was done using TFIDF word representation 79 % F1 score was achieved (See Table 12 A comparison between Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 research and this research). This experiment contradicted the statement increase in sample size increases model performance. But it has been seen during a previous experiment that removing specific words or sentences from a data frame can improve a models performance by a huge margin (See Removing 25 % least frequent words) when 25 % of most frequently occurring words were removed randomly by the percentage of their occurrence in the data frame the Bernoulli NB model had 88 % F1 score in that instance.

It could be said that Mohiyaddeen & Siddiqui, 2021 achieved this high accuracy by removing specific words from their data frame or by having a large class imbalance between their classes but the concrete proof would still be required for future research could carry experiments related to this and may find something useful for the development of online hate classifiers as a result.

The model used in Dhamija et al., 2021 used a decision tree classifier and it achieved a higher result than the Naive Bayes model used in Mohiyaddeen & Siddiqui, 2021 research on hate classification. This research used both algorithms and found that Naive Bayes performs better than the decision tree algorithm when performing hate classification (See Table 12 A comparison between Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 research and this research). These results could be because of the number of classes used in this research or the difference in the dataset used in Mohiyaddeen & Siddiqui, 2021, Dhamija et al., 2021, and this research. It could be because of various other reasons. This could be found by experimenting on different datasets using the same-mentioned algorithms.

|  |  |  |  |
| --- | --- | --- | --- |
| Mohiyaddeen & Siddiqui, 2021 | | | |
| Model | Distribution | Representation | Accuracy |
| Naive Bayes | Not Given | Not Given | 93 % |
| Dhamija et al., 2021 | | | |
| Model | Distribution | Representation | Accuracy |
| Decision tree | Not Given | Not Given | 99 % |
| This research | | | |
| Using 1000 samples | | | |
| Model | Distribution | Representation | Test F1 score |
| Naive Bayes | unigram | BOW | 79 % |
| Naive Bayes | unigram | TFIDF | 79 % |
| Decision Tree | unigram | BOW | 66 % |
| Decision Tree | unigram | TFIDF | 65 % |
| This research | | | |
| Using 2000 samples | | | |
| Model | N-grams | Representation | Test F1 score |
| Naive Bayes | unigram | BOW | 84 % |
| Naive Bayes | unigram | TFIDF | 83 % |
| Decision Tree | unigram | BOW | 76 % |
| Decision Tree | unigram | TFIDF | 73 % |
| This research | | | |
| Using 2500 samples | | | |
| Model | N-grams | Representation | Test F1 score |
|  |  |  |  |
| Naive Bayes | unigram | BOW | 79 % |
| Naive Bayes | unigram | TFIDF | 79 % |
| Decision Tree | unigram | BOW | 76 % |
| Decision Tree | unigram | TFIDF | 72 % |

Table 12 A comparison between Dhamija et al., 2021 and Mohiyaddeen & Siddiqui, 2021 research and this research

The models produced in the research of Malik et al., 2021, Chakravartula, 2019 and Abro et al., 2020 have poorer performance than the performance of the best model in this research. Abro et al., 2020 model had a 77 % F1 score, Malik et al., 2021, the model had an 81 % F1 score and Chakravartula, 2019 had a 73 % F1 score comparatively this research had an 88 % F1 score performance. As Malik et al., 2021 used the CNN algorithm with fastText and BERT representation the only comparison between there and this research that could be made was a performance comparison as this research does not use CNN. Chakravartula, 2019 used the Multinomial NB algorithm, Bag of words representation and 73 % F1 score. This research had poor performance using Multinomial NB with a Bag of words representation and it was decided that it will not worth investing time in studying in detail hence the results were not recorded. The best performing Bayesian algorithm used in this research was Bernoulli NB with an F1 score of 88 % which performed comparatively better than the Chakravartula, 2019 model by 15 %.

In the research performed by Abro et al., 2020 when they used the SVM algorithm in combination with TFIDF representation and bigram distribution on an unbalanced dataset they received the result of a 77 % F1 score. In this research, the unigram distribution performed better than both the bigram and trigram distribution whether it was used with TFIDF or BOW representation (See Experimental results). Abro et al., 2020 experiment suggested that with bigram distribution, TFIDF representation using SVM algorithm relatively good results good be obtained. This research did not use the SVM algorithm because of hardware and time constraints but further experiments could be carried out in future research to determine if SVM performs better on bigram and trigram distribution or was it specifically for the case of the dataset and classes used by Abro et al., 2020.

The research by Warmsley, 2017 uses a Logistic regression algorithm and a Bag of words representation. They do not give the distribution of words in their research. They achieve a 90 % F1 score. Warmsley, 2017 in their experiment used an attempt to create a model for a multiclass classification problem. They give model performance results for each class. According to Warmsley, 2017 they had three labels hate, offensive and neither. Neither was classified correctly at 95 %, the offensive was classified at 91 % time correctly and hate was classified at 61 % times correctly. As hate was classified correctly only 61 % time it can be seen why a balanced dataset should be important. Comparatively to Warmsley, 2017 research this research with its best model was able to classify both toxic and nontoxic tweets exactly equally with an F1 score of 88 % (See Removing 25 % least frequent words).

The research of Anjum & Katarya, 2021 has a very similar approach to this research but they use Oversampling to balance the dataset. They achieved a 91 % F1 score using the SVM(Linear SVC) algorithm and TFIDF representation. They did not provide which word distribution they used. As their model learned using an oversample class it would be worth studying its predictability when it is applied on a large new similar dataset with a large number of columns compromising the mentioned over sampled class. If the results are still consistent with Anjum & Katarya, 2021 research then this could potentially be a very good model to build future hate classifiers. In comparison to Anjum & Katarya, 2021 research this research has a poorer performance even using its best performance algorithm but a relevant performance measure between both these approaches can only be made when both mentioned models are applied to a new similar dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Study** | **Algorithm** | **Representation / embedding** | **Distribution** | **F1 scores** | **Balanced dataset** |
| This research | Bernoulli NB | TFIDF | Unigram | 88 % | Yes |
| This research | Bernoulli NB | Bag of words | Unigram | 88 % | Yes |
| Abro et al., 2020 | SVM | TFIDF | bigram | 77 % | No |
| Malik et al., 2021 | CNN | fastText and BERT | Not Given | 81 % | No |
| Anjum & Katarya, 2021 | SVM(Linear SVC) | TFIDF | Not Given | 91% | Yes (Oversampling) |
| Chakravartula, 2019 | Multinomial NB | Bag of words | Not Given | 73 % | Not Given |
| Warmsley, 2017 | Logistic regression | Bag of words | Not Given | 90 % | No |
| **Study** | **Algorithm** | **Representation / embedding** | **Distribution** | **Accuracy** | **Balanced dataset** |
| Mohiyaddeen & Siddiqui, 2021 | Naive Bayes | Not Given | Not Given | 93 % | Not Given |
| Dhamija et al., 2021 | Decision tree | Not Given | Not Given | 99 % | Not Given |

Table 13 Comparison between the current study with previous studies

## Practical implications

The classification model created in this experiment can be deployed and be used in another research and is available at <https://github.coventry.ac.uk/iftikhars/6001_Individual_Project>. It's doubtful that "perfect" classifiers for online hatred will ever be created, especially given the subjective nature of what constitutes online hate (Salminen et al., 2020). As a result, machine learning models in this research should be viewed as "assistants," or decision-making aids, rather than the absolute truth. Their value stems from the model's capacity to swiftly scan many comments—far too numerous to humanly inspect—and identify the most probable hostile content for quarantining. Then, based on human judgement and relevant ethical considerations, other users or moderators can make the ultimate choice on whether to remove such content (Salminen et al., 2020).

Because the annotation of abusive comments for model training is often decontextualized, meaning it overlooks community-specific ways of using language and differing criteria for what is hateful, the cautious use of classifiers for automatic moderation is equally vital. According to Salminen et al., 2020 as Castelle wisely observed, "Because the flags are provided by users who have seen the entire interaction, many comments are considered offensive in context but not offensive when standing alone." (Page 8) Ethical issues in online hate detection are critical for these reasons. As a result, it is not recommended to allow the model to decide on message blocking or removal on its own (perhaps apart from situations where false positives play only a smaller role). Rather, using the human-in-the-loop approach, hate detection algorithms may be used to flag comments for human moderation (Salminen et al., 2020).

As Lucy Trevelyan LLB states Bullying and cyberbullying are not crimes in and of themselves. There are, however, criminal, and civil statutes that can be used to prosecute cyberbullying perpetrators (Trevelyan, 2022). So, tracking extreme cases cyber bullying could protentional help in the prevention of cases of bullying leading to legal actions. Saving the victim, the abuse, saving the legal system time and cost as an indirect consequence.

## Limitations and future research

This research had two main limitations (1) Hardware limitations (Appropriate hardware to process the whole dataset was not available when conducting this research). (2) time constraint (this research had a fixed time in which it could be conducted), (3) binary categorization might be seen as a restriction in and of itself, but the scope of this study was a binary classification, and this model has a lot of usefulness when it comes to reporting hostile social media remarks and (4) This model could only be applied to Twitter and would not work as well on other social media platforms.

Several points of improvement have been highlighted for future research. First, removing specific words increases a model’s performance immensely (See Experimental results). Further research could be carried out using the same dataset and constraints and it could be determined the removal of which specific words increases the model’s performance.

Second, Hyperparameter optimization was performed using random search and a constricted set of hyperparameters because of hardware constraints even so the SGD and LDA model (See SDG hyperparameter optimization models on 1000 samples and Cross-validation results evaluation) with hyperparameters optimized almost matched the performance of the best performing the Bernoulli NB model (See Bernoulli NB default models on 1000 samples) given the same set of constraints. This limitation leaves space for further improvement using different architectures and sets of parameters.

Third, auto-sklearn was not utilized to its full potential and was applied to the dataset with constrictions due to hardware constrictions of the machine used in this experiment (See Auto-sklearn with max samples set to 1000). The application of automated machine learning tools could be further, and a better comparison could be made between traditional machine learning and automated machine learning technologies in hate detection in future research.

Finally, for the study of the creation of a Twitter-based hate classifier this research uses a publicly accessible data base and the code used in this research is available online so the models created in this research could be recreated, used, and improved in the future research.

# Discussion

Following an examination of the findings, it was discovered that it is feasible to perform hate speech classification, with the Bernoulli NB model proving to be the most accurate method in the case of this experiment.

This study concentrated on polarity and sentiment scores, which only included positive and negative tweets, and employed 14 algorithms to assess the dataset's 1000 attributes from which 3 best performing algorithms were further evaluated. The Bernoulli NB had the best performance but when removing the least frequent words by percent it achieved the best results (See Removing 25 % least frequent words) as the results are from randomly removed least frequent words by percentage the results will be hard to replicate and thus require further investigation into specific words removal and their impact on model performance (See Limitations and future research).

All the pre-set objectives of this research have been met as (1) abusive language detection models have been created. (2) Created models' performance had been reviewed in depth. (3) A model that can detect abusive langue on unseen data with high performance had been created. (4) The created model had good results on all metrics while keeping the actual positive and actually negative classifications balanced. (5) Previous works on the same topic were improved upon in this research as this research provided insight into prior research. (6) This research can be used to provide valuable information for future research or online hate classifiers.

# Conclusion

To reduce toxicity on social media sites, online hate detection is required. In this study, various machine learning models for online hate detection were tested and it was discovered that after removing 25 % of least frequently occurring words using Bernoulli NB algorithm, with bag of words or TFIDF features representation, unigram distribution gave the best results on 1000 selected tweets with an 88 % F1 score and it had equal classification performance of both toxic and non-toxic tweets (See Removing 25 % least frequent words).

This project was carried out by keeping legal, social, ethical, and practical concerns kept in mind (See Legal and social, ethical and practical concerns). This project was completed keeping the British computing society ethos in mind (See Legal and social, ethical, and practical concerns concerning this project). All the set objectives of this project have been met (See Objectives and Discussion). This project has real-world practical applications and can contribute to future research (See Practical implications and Limitations and future research).

The model created in this project could directly be applied to create Twitter-based hate classifiers, but the data used in this experiment was substantially low hence before any large-scale practical applications it is recommended that tests should be carried out with a larger scale of data than the amount of data used in this project. This project also highlights finding better word reduction methodologies and finding specific words which can be removed to increase a model’s performance and potentially optimise a machine learning model (See Limitations and future research).

# Reflection

During this study, I learned how to do research formally. This entailed learning how to locate relevant prior works and basing my research methodology on what worked effectively in past studies. Because my study entailed tests and a huge number of different findings from each test, I had to acquire statistical analysis techniques so that I could compare the data from each test and draw conclusions. I learned how to build the structure and goals for the rest of the study around these results.

I learned new things, such as the overall conclusion that using my dataset and technique, it was able to classify tweets correctly and build an effective model with limited time, resources, and data. This was information that did not exist before the completion of my investigation. My programming abilities improved on a smaller scale as I learned how to implement a range of different forecasting algorithms. I learned how to deal with setbacks using agile principles in combination with Gantt charts. Practical coding skills I learned specifically during this project are pre-processing techniques such as removal of Twitter-specific information (user mentions, emoticons, retweets, URLs, and hashtag symbols; only retaining textual content), tokenization, lowercasing, stop word removal (with different stop word lists for the datasets) and lemmatization. I also learned the application of TFIDF, the Bag of words model and using auto-sklearn. Through Dr Elshaw I learned the methodology and thinking that goes into research work.

Most of the coding-related aspects of this project went well as I had experience with machine learning before this research learning new techniques was not challenging. Time management became an issue during this project because of my sickness. I was unable to complete the project in the given time frame and had to apply for a deferral. I achieved what I set out to do during this project by applying sentiment analysis algorithms, to categorise abusive human language on a balanced Twitter dataset and got good results by the measures I set when I started this project.

If I, did it over, I would get early access to the Coventry Universities supercomputer so I could get better results, or I would select a project focusing on a specific word removal aspect of this experiment to understand which word removal has the most impact on improving a models performance.

## Project Management

Failure in a project is caused by a lack of a defined approach. Project management has typically been dominated by large-scale technical projects. Project management, on the other hand, has become a basic business practice for most organisations today (Maylor, 2001). Agile, Scrum, Kanban, and Waterfall are common methodologies used in software project management (Andrei et al., 2019).

Agile, Scrum, Kanban and Waterfall project management methodologies were looked at for the management of this project and for the time management aspect of this project the Gantt charts will be used. I decided to use Gantt charts for this project as I had used them before and found them to be an effective way of time and task management. According to Maylor, 2001 Gantt Charts are a crude tool on their own and they recommend using a hybrid approach. Keeping Maylor, 2001 statement in mind in this research I aim to study other project management methodologies and use the one which would be most suited for this project in combination with Gantt charts.

### Gantt charts

Because they appreciate the benefits of splitting enormous projects into smaller manageable tasks, Gantt charts are crucial in the history of contemporary project management. They also take into consideration the fact that some tasks may be interdependent. Gantt charts are still in use today and are regarded as an important tool in the toolbox of any project manager (Seymour & Hussein, 2014).

### Agile

Agile Methodologies are a collection of software development methodologies that are iterative and incremental. Adaptive planning, iterative and evolutionary development, quick and flexible reaction to change, and fostering communication are the four primary features that all agile techniques have. Its major focus is on adhering to the "Light but sufficient" ideals and being people-oriented and communication-centred. Because it is referred to as a lightweight method, it is better suited to the creation of little projects. According to agile software development, production teams should begin with basic and predictable approximations of the final demand, then gradually increase the depth of these requirements throughout the development process. At all phases of the production process, this incremental requirement refinement refines the design, coding, and testing. As a result, the work product for requirements is as precise and helpful as the final programme. “The principle of agile software development proposes that at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly” (Kumar and Bhatia, 2012).

According to Kumar and Bhatia, 2012 the Agile Manifesto is a document that outlines the agile software development methodology. Some of the manifesto's signatories formed the Agile Software Development Alliance, or Agile Alliance, in 2001 as a non-profit organisation dedicated to refining the manifesto's concepts into a set of twelve principles. The agile manifesto's twelve principles are as follows:

1. The main objective is to satisfy the client by delivering valuable software on time and consistently (Kumar and Bhatia, 2012).
2. Changes in requirements are welcome, especially if they occur late in the development process. Agile procedures take advantage of the change to help customers gain a competitive advantage (Kumar and Bhatia, 2012).
3. Deliver functioning software regularly, anything from a few weeks to a few months, with a preference for the shorter timeframe (Kumar and Bhatia, 2012).
4. Throughout the project, businesspeople and developers must collaborate regularly (Kumar and Bhatia, 2012).
5. Build initiatives around people who are passionate about what they're doing. Give them the space and support they require and trust them to do the task (Kumar and Bhatia, 2012).
6. Face-to-face communication is the most efficient and effective way of transmitting information to and within a development team (Kumar and Bhatia, 2012).
7. Working software is the most important indicator of progress (Kumar and Bhatia, 2012).
8. Sustainable development is aided by agile procedures. Sponsors, developers, and consumers should all be able to keep up a steady pace indefinitely (Kumar and Bhatia, 2012).
9. A constant focus on technical excellence and smart design improve agility (Kumar and Bhatia, 2012).
10. Simplicity, or the art of minimising the amount of effort that isn't done, is critical (Kumar and Bhatia, 2012).
11. Self-organizing teams produce the finest architectures, requirements, and designs (Kumar and Bhatia, 2012).
12. The team considers how to become more successful at regular intervals, then tweaks and adapts its behaviour accordingly (Kumar and Bhatia, 2012).

### Scrum

Scrum is a framework for building complicated projects and organising work that is founded on a set of values (courage, focus, commitment, respect, and openness), principles, and practices that serve as a foundation for all team members. When faced with changing constraints, whether financial or technological, the Scrum framework offers great flexibility and versatility, and its key to success is always focusing on the highest-priority tasks, each of which goes through a process of seven steps: requirement elaboration, design, development, comprehensive testing, integration, documentation, and approval (Andrei et al., 2019).

The Scrum framework is made up of three roles, three artefacts, and five events (Andrei et al., 2019).

#### Scrum Roles:

**Product owner** – oversees choosing which features and in what order they should be deployed. The product owner will keep in touch with the other team members and is in charge of the solution being created.

**Development team** – consists of typical software developers that specialise in numerous disciplines (architect, tester, developer, and UI/UX). The development team will be self-organized and determine how to continue to meet the product owner's objectives most effectively feasible.

**Scrum master** – is a leader who assists the rest of the team in understanding and using the Scrum framework appropriately. The scrum master should not be mistaken for a project manager because they have no authority over what the development team chooses. However, the scrum master can assist the team by shielding them from outside interference.

#### Scrum Artifacts:

**Product backlog** – a prioritised list of things that need to be resolved. Changes to existing components, "bugs" or faults that need to be rectified, infrastructure enhancements, and so on can all be included in the backlog.

**Sprint backlog** – is a prioritised list of things that should be handled during the sprint (it's similar to a product backlog, but lighter because the items should be addressed in a short period, as opposed to a product backlog, which may represent months of effort).

**Increment** – a set of things from the product backlog that was completed during a sprint.

#### Events:

**Sprint** – Work is completed in time-boxed iterations that are usually never more than 30 days when employing a scrum methodology. The sprints last the same amount of time and are solely focused on creating a single feature with no interruptions or goal-altering interventions.

**Sprint planning** – is a meeting where the sprint backlog is created and the current sprint's goals are set, as well as any ambiguities, are resolved. The sprint objective is agreed upon by the product owner and the development team, and the sprint backlog is then ordered per that goal.

**Daily scrum** – a daily meeting, generally lasting no more than 15 minutes and taking place at the same time every day, that focuses on what has been completed since the previous daily scrum and what needs to be prioritised for the following day. Its purpose is to keep team members' work in sync and to keep management informed about the project's progress.

**Sprint retrospective** – the purpose of a sprint retrospective is to objectively assess what was completed and analyse what went well and what did not, to determine what can be improved for future feature sprints. The Scrum Team will identify changes that should be implemented during the following Sprint after the Sprint Retrospective, demonstrating the Scrum framework's flexibility.

### Kanban

Kanban is divided into five stages: visualising the process, restricting work-in-progress, controlling the workflow, making each step clear, and eventually maturing into a single, exact system (Andrei et al., 2019).

According to Andrei et al., 2019 Kanban’s goal is to eliminate any "bottlenecks" in a streamlined process, maximising efficiency, and collaborative teamwork throughout the whole team. This purpose led to the design of the Kanban board, which has four separate types of tasks that should be associated with the present project:

**Ideas** – these are tasks that are still ambiguous, and there are various conversations among members about whether the work or feature should be done ("is it too time-consuming?" "Is it profitable?" "What are the pros and cons?") (Andrei et al., 2019).

**To do** – these tasks have passed the first stage and are ready to be put into action. The task's assignment is still the sole issue to resolve. Kanban aims to enhance workflow by distributing tasks such that no members are "blocked," which can refer to either waiting for another person to accomplish their portion of the work or having too many "to do" activities running at the same time (Andrei et al., 2019).

**In process** – the assignment has been assigned and is currently being worked on by a team member (Andrei et al., 2019).

**Done** – the assignment is finished, and there is no further work to be done on this specific feature (Andrei et al., 2019).

### Waterfall

According to Andrei et al., 2019 The Waterfall model was the first software development approach employed, and it is similar to other industries' designs. This technique allows a project to be divided into many defined stages, each needing the preceding phase's analysis and work:

**Requirements** – a thorough examination of business requirements and exhaustive documentation of all features (Andrei et al., 2019).

**Design** – deciding on the necessary technologies and laying up the whole software infrastructure and interface (Andrei et al., 2019).

**Coding** – utilising the diagrams and blueprints from the design phase, solving all issues, optimising solutions, and implementing each component provided in the requirements phase (Andrei et al., 2019).

**Testing** – thoroughly testing all implemented features and components, as well as resolving any issues that arise (Andrei et al., 2019).

**Operations** – “deployment to a production environment” (Andrei et al., 2019).

### Choosing a project management approach

Each approach has its own set of advantages and disadvantages. As a result, there is no one-size-fits-all answer for all projects. Various elements should be addressed, such as the number of individuals on the team, how adaptable the needs are, and the project's duration (Andrei et al., 2019).

The methodology adopted is unique to each team and must be chosen expressly for that project, as no technique can meet all requirements. The waterfall is often used by small teams for short projects with well-defined requirements, whereas Agile is more adaptable and preferable when continual input is critical (Andrei et al., 2019).

As continual supervisor input was important for this project the agile approach was decided to be used for this project as everting required for this project was publicly available and could be implemented with self-study. The project was divided into chucks using Gantt charts hence it was decided each chuck would be built keeping the agile principles in mind then supervisors’ feedback could be used to improve the project further in the next chuck.

### Problems with generic work management methodologies

There are two main problems with generic project management methodologies. The first one is the risk of burning out (See Burn out problem) and the second one is prostration (See Prostration problem).

To avoid burnout control over job activities, working longer hours and spending more time on administrative duties and paperwork were targeted in this research to be more productive. As there were multiple modules to focus on so by default there was a large sum of work in a short period. Most of the work in this module was paperwork intensive and external factors were also unavoidable, so they were considered too. The solution I came up with for these situations was micro project management. The micro-project management technique used in this project is called the Pomodoro technique. By using the Pomodoro technique tasks were managed daily in chunks and were easily trackable this led to a boost in productivity and helped me reduce the chances of Procrastination and burnout.

The solution I came up with for these situations was micro project management. The micro-project management technique used in this project is called the Pomodoro technique (See Pomodoro technique). By using the Pomodoro technique tasks were managed daily in chunks and were easily trackable this led to a boost in productivity and helped me reduce the chances of Procrastination and burnout.

#### Burn out problem

People working in modern organisations frequently appear to forget that everyone is endowed with a certain amount of energy that must be prudently handled for it to serve as a source of professional accomplishments and a sense of fulfilment throughout their careers (Moczydłowska, 2016). In the 1970s, burnout became the subject of societal attention and research for the first time. Similar symptoms have been mentioned in previous literature and nonfiction, including excessive tiredness and a loss of idealism and enthusiasm for one's vocation. The main story goes like this: people go into a job with high hopes, passion, and the desire to succeed. Things have changed throughout time, and people today feel exhausted, frustrated, angry, and cynical, as well as a sense of ineffectiveness and failure (Maslach & Leiter, 2017).

Internal and environmental factors might contribute to academic burnout. Internal factors include perfectionism's maladaptive influence, dispositional attentiveness following traumatic situations, and students' performance-avoidance attainment aspirations. External factors include parental rejection and overprotection, as well as the school environment (Yanting, 2022). Overall, higher levels of emotional weariness were linked to having less control over job activities, working longer hours, spending more time on administrative duties and paperwork, seeing more managed care customers and fewer direct pay clients, and dealing with more unfavourable client behaviours (Rupert & Morgan, 2005). When dealing with burnout, Gautam Gulati, adjunct associate clinical professor of psychiatry at the University of Limerick in Ireland, told the Bulletin that signs indicative of underlying depression, a common and curable disorder, should not be ignored. Poor sleep, lack of energy, reduced appetite, lack of interest in normally fun activities, feeling down and sad, or having suicidal ideas can all be signs of a depressive episode (Winter, 2020).

#### Prostration problem

Procrastination is the act of deferring appointments, tasks, ideas, choices, crucial discussions, or the completion of a degree to a later day, month, or year (As stated by Gargari et al. 2011 according to Almalki et al., 2020 as stated by Almalki et al., 2020). Procrastination affects many people, according to studies (According to Lukas and Berking 2017; Stead et al. 2010; Ferrari and Daz-Morales 2014 as stated by Almalki et al., 2020), and some of the causes are linked to mental health issues (according to Lukas and Berking 2017 as stated by Almalki et al., 2020). Procrastination has a huge impact on a student's behaviour, health, and psychology in the classroom (As stated by Hooda and Saini 2016 according Almalki et al., 2020).

Procrastination can take many different forms. For example, some students put off submitting an assignment, finishing a project, or studying for a test until the deadline approaches. The underlying causes of students' procrastination include, first, a lack of desire caused by a fear of failing (According to Hooda and Saini 2016 as stated by Almalki et al., 2020). Second, time management is crucial for graduate students, as everything from reading numerous papers to scouring the internet and libraries for papers and books to preparing for the thesis takes a significant amount of time and effort. Perfectionism is one of the factors that cause graduate students to procrastinate (According to Onwuegbuzie 2000 as stated by Almalki et al., 2020).

#### Pomodoro technique

The Pomodoro technique was created in the early 1900s by Francesco Cirillo, who called it after the tomato-shaped timer he used to keep track of his work as a university student (According to Henry 2019 as stated by Almalki et al., 2020). The technique is useful for time management and self-control (As stated by Cirillo 2006 according to Almalki et al., 2020). The Pomodoro Technique's concept is simple: if you have a big assignment, break it down into smaller chunks with a brief rest in between (According to Henry 2019 as stated by Almalki et al., 2020).

### My project management approach

My project planning was well-organized, and I created a Gantt chart that differed somewhat from the one in my original project proposal. I kept track of each task's progress and discovered that I was generally ahead of schedule but due to suffering from a medical condition I had a setback to which the university extended my submission period upon application. This allowed me to add additional activities to the chart and perform more research while still finishing the project write-up and presentation before the deadline (See Figure 44 Gantt chart).

Every day the project was micromanaged by using the Pomodoro technique. Most days of the project were focused on doing 1 hour of work and a 15-minute break after words. Each day of work consisted of a 1 hours walk awarded upon completion of work done each day.

Graphical user interface, table

Description automatically generated

Figure 44 Gantt chart

Looking at each stage of the project their timeline has been given (See Figure 44 Gantt chart) there are as follows:

### Project selection

A suitable project had to be selected for this I researched current problems, looked at potential solutions and choose a set of problems that I would like to solve and that I could contribute to after which I discussed them with Coventry University staff and decided that I would like to study and improve upon the study of applying sentiment analysis on an online form.

### Finding a supervisor

Finding a suitable supervisor was a must as I had selected my project well before other students, I investigated supervisors that had an interest in studying natural language processing, machine learning and AI. I did consider asking Dr Richard Hide, Dr Trang Doan and Dr Mark Elshaw for supervision of my project.

### Creating a Gantt chart

This step was decided to be carried out to mark my progress and manage my progress. In combination with agile principles.

### Asking Dr Elshaw for supervision

I decided that I would ask Dr Mark Elshaw to supervise my project as he was highly creditable in my area of interest and I apricated his methodology of teaching from my experience studying under him.

### Research on the Selected Project

Dr Elshaw recommended that we should focus on a specific platform rather than researching the application of sentiment analysis on online forums in general. We decided to focus on Twitter as it was a popular platform with data publicly available and its API also allowed easy access to data if required.

I started to perform research on the selected topic and began formulating a project proposal which would become the foundation for this project.

### Write the project proposal

After initial research had been done on the project it was written with the help of Dr Mark Elshaw and Mr Peter Every. A Twitter dataset was also selected to be used in the project at this stage and was approved by Dr Mark Elshaw. It is the same dataset that will be used in this project. Dr Elshaw also provided me with two templates to use as a guiding block for further research for my project.

### Fill in the Ethical application

This project needed to be ethically approved by the relevant department at Coventry University before it could be carried out. The Ethical application was filled in with the help of Dr Mark Elshaw and Mr Peter Every.

### Completing pre-processing

After the Ethical application had been approved and the initial project proposal had been submitted it was a university study break, so I decided to self-study by applying sentiment analysis on the selected Twitter data set.

I had prior experience learning machine learning from Dr Trang Doans hence I had a general idea of what to research (See Pre-processing and feature representation). After my research following the agile principle as stated by Kumar and Bhatia, 2012 working software is the most important indicator of progress I decided to start the coding process. I completed the pre-processing stage during my university study break.

### Model application

As I had completed the pre-processing stage, I decided to test the accuracy of different models on my data this stage was also done during my university study break this was initially done on 1000 samples and was decided again to use the said samples for the implementation used in this project (See Agile collaboration and Experiments, Evaluation and Results).

### Project completion

As the coding part of the project was completed, I decided to focus on completing the overall project which included further research and evaluating my findings as this stage required constant collaboration with Dr Mark Elshaw I decided to use the agile approach henceforth.

I firstly showed Dr Elsaw everything I had done to which he approved except where I was replacing words that started with not \_ or their synonym (See Replacing words starting with not with their synonym or not\_) when I showed Dr Elsaw this, he commented that I could do this as a secondary approach to compare to my initial research in which I do not use this methodology. I followed his advice the results of which have been presented in this research (See Replacing words starting with not with their synonym or not\_).

#### Agile collaboration

According to Kumar and Bhatia, 2012 throughout the project, business people and developers must collaborate regularly and Face-to-face communication is the most efficient and effective way of transmitting information to and within a development team. Keeping the mentioned in mind I started the remainder of my project. I would talk with Dr Elshaw on teams and Outlook. I would complete a task he would give me on Outlook and receive his feedback on teams on the completed task.

My first hurdle was the machine I was using was not able to process the required data fully as it did not have the memory needed. After I discussed this with Dr Elshaw, he got permission for me to use Coventry Universities Supercomputer. I aimed to complete my project henceforth, but I fell ill and after consulting with Dr Elshaw, Mr Every and my GP I decided to apply for a deferral.

During this time, I remained in contact with Dr Elshaw. He was very supportive and would take time not in his regular work hours for our meetings and he helped me significantly during this process.

As the Coventry Universities Supercomputer had a long queue for usage. I asked Dr Elshaw if I could use 1000 samples for my project to which he approved saying if I could justify why I was doing so and hence I decided to go with the mentioned methodology (See Experiments, Evaluation and Results). Now every aspect of the project was completed coding experimenting-wise.

I completed my whole project and presented it to Dr Elshaw to which he proposed correction after the corrections had been made to the report, I decided to submit my work.

My collaboration with Dr Elshaw was on teams and Outlook the record of which can be seen in the table below (See Table 14 Collaboration with Dr Elshaw).

|  |  |  |
| --- | --- | --- |
| **Date** | **My question** | **Dr Elshaw’s response** |
| 12/12/2021 06:30 am | Dataset approval. | Approved |
| 12/12/2021 06:30 am | is binary classification a good approach | Yes,+ try looking at auto ml it will save you time in hyperparameter optimisation. |
| 16/12/2021 01:30 am | Should I perform a Turing test for my project? | complete a simple project first then think of more complicated aspects. |
| 23/12/2021 07:23 pm | Could I have guidance on how to write a report could you recommend me from previously done works any good ones? | Yes please look at resources 1 and resource 2. |
| 03/01/2022 09:02 am | Can I use google colab for my project would it be allowed? | Yes, you can. |
| 05/01/2022 07:35 pm | Could I set max\_features = 2000 when using TFIDF and Bag of words model. | Don’t do that as you are losing data. |
| 05/01/2022 07:35 pm | My pcs hardware is not good enough could you get access for me to use the universities supercomputer? | I will do that. |
| 22/01/2022 21:33 | I want to use TFIDF and Bag of words is that ok? | It is up to you if you can justify your choice. |
| 22/01/2022 22:36 | Is there a word limit on the report | There is no word limit, but you need to write a well-structured report. |
| 04/02/2022 02:04 pm | I am not feeling very well I don’t think I will be able to complete the project on time. | I would advise speaking to Peter Every about this. |
| 21/03/2022 05:01 pm | Our deferral has been approved we can proceed accordingly. | Ok. |
| 06/04/2022 19:20 | I have tried using partial\_fit but it doesn’t seem to work have you any experience with that? | “It looks like it trains using a number of samples at a time.  It should work.” |
| 14/04/2022 9:08 | “Dr Elshaw i haven't finished the full literature review yet but i have two questions. You can answer them in our meeting.   1. The gap that we are looking for in previous research. according to [https://hcis-journal.springeropen.com/articles/10.1186/s13673-019-0205-6](https://eur01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fhcis-journal.springeropen.com%2Farticles%2F10.1186%2Fs13673-019-0205-6&data=05%7C01%7Ciftikhars%40uni.coventry.ac.uk%7Cbcc9955a5b8d4246644508da235fdb0c%7C4b18ab9a37654abeac7c0e0d398afd4f%7C0%7C0%7C637861191846637386%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=VG%2B%2BOTkv2nXO1TLXJq6bTN768Yqw6dGGiGCRFOhdzis%3D&reserved=0) there is not much research done in cross platform sentiment analysis. As there research is very similar to ours i was thinking using there basic idea but applying our twitter learned model and testing it on Facebook and Instagram data. 2. As we discussed before we could also do sub tests on chunks and compare the results on a complete test set + as we also discussed before we can check how many frequent words we can remove to get a dataset that gives us good results.     keeping point 1 and 2 in mind i would like to ask as this is a bachelors level dissertation and the timing is limited. would it be ok for us to do just a research based approach and try experimentation if we have time to + doing both 1 and 2 would be time consuming so if required which should be given priority?” | “Seem fine, you do not need to focus on cross platform” |
| 19/04/2022 6:24 | “This is my current written progress a short review would be appreciated.” |  |
| 21/04/2022 11:26 |  | “This is good so far.  What need in literature review is more example of Machine Learning research into toxic text classification or if not enough of that similar areas such as spam or fake news.” |
| 23/04/2022 13:35 | “ive been doing experiments with different number of features as there is a que to the super computer whats the minmum number of features that would be accipetable for the experiment.   with 1000 and 2000 samples I’m getting very good results actually.” | “That is good, present the results and discuss this in the report.” |
| 24/04/2022 13:47 | Gird search is taking too long is it ok if I just use random search for hyperparameter optimization? | Use random search. |
| 01/05/2022 | Could you kindly check my work and provide feedback? |  |
| 11/05/2022 |  | “Consider the impact that this research might have on the real world.”  “Compare the performance of your model with the existing models in the literature.”  “Include the research question, the objectives of the project and the motivation”  correct your font usage.  “consider other studies related to binary text classification.”  “Number the sections and subsections  ”  “Include an image of decision tree”  “Put the equations in appropriate form”  “Explain the different ways you decide which words you are going to include in the models”  “How did you decide on the thousand features? Features are the columns”  “What do you mean by selecting randomly the least frequent words? Explain”  “Why 56%? Why are you reducing the features explain”  “Are we not using the frequency in the tweet so not sure why you are doing this. The frequencies in the tweets is the important thing. Explain why this is a viable approach.”  “Compare your results with those with those studies in the literature review”  “consider if you met the objectives of your project.”  “What skills did you get out of the project. When is the main achieve. What went less well? What would you change?”  “Talk about the supervision meetings. Have an example document developed for supervision meetings.  Anything did as a response to supervision meetings  Any tools you used to supervise project.”  “Consider the social legal and ethical issues. For instances freedom of speech issues, should a machine be deciding what is appropriate etc.” |
| 05/06/2022 | Could I submit my work? | In this team meeting, Dr Elshaw told me to add a risk analysis table and to add practical concerns taken into account during this project.   Dr Elshaw also instructed me to change my definition of features to samples in some cases as I have them mixed up.  Dr Elshaw had other concerns. I had already corrected and guided him through them.  You have done the work if you are happy with it then submit it. |

Table 14 Collaboration with Dr Elshaw

I would complete a task mentioned in Table 14 Collaboration with Dr Elshaw then I would discuss it with Dr Elshaw upon his approval I would move to the next task. I completed all tasks given to me by Dr Elshaw and acquired his approval before making a final submission.

Certain risks were looked at during this project and their corresponding solutions were also considered (See Table 15 Risk analysis).

|  |  |
| --- | --- |
| **Risk** | **Solution** |
| If no suitable dataset is found for this project. | 1. Oversampling could be performed. 2. A different project could be selected. |
| If the machine used for this project cannot process the data. | 1. Get access to the Coventry University supercomputer. 2. Use lower samples than the total samples in this project. |
| Being unable to find high-performing machine learning algorithms for the project. | Use current high-performing algorithms from prior research and have them compete against auto-sklearn algorithms to choose the best algorithms for this project. |
| If this project crosses any ethical, social, and legal concerns? | To keep this project from crossing any ethical, social, and legal concerns the British computing society’s code of conduct was followed and literature on ethical, social and legal concerns was studied during this project by highly creditable sources. |
| If this project cannot be completed in the given timeframe. | To make sure the project was completed in the given timeframe CoventryUniversities' policies on extensions and deferrals were considered. They were provided by Mr Peter Every and the deferrals were applied for in this project due to personal circumstances.  The agile approach, Pomodoro techniques and Gantt charts were used to manage the project on time. |
| If the set target set for the project cannot be met at the time of project completion. | The targets that can be met should be focused on more. It should be attempted that the set targets are such that they can be met easily. If any targets cannot be met due to any issue guidance should be asked from the project supervisor and all of the project details should be given in the final report without holding anything back. |

Table 15 Risk analysis

This report was fully revised on 20/06/2022 before submission.

## Legal and social, ethical and practical concerns

### Social and ethical concerns

The below ethical concerns regarding AI have been studied by Dr Rowena Rodrigues extensively by Dr Bernd Carsten Stahl. Bernd Carsten Stahl is a Professor of Critical Research in Technology and Director of the Centre for Computing and Social Responsibility at De Montfort University, Leicester, UK.

The first group of concerns are those that develop because of machine learning characteristics. Artificial neural networks are at the heart of many of the machine learning approaches that have contributed to AI's present success. The opacity, unpredictability, and requirement for enormous datasets to train the technologies are all elements of these techniques that raise ethical problems. In most cases, neither the developer, the deployer, nor the user can predict how the system will respond to a set of inputs in advance. Past behaviours are not a perfect prediction of future behaviour is identical settings since the system learns and is hence adaptable and dynamic.

As Buttarelli, 2018 said privacy and data protection is a major and widely discussed ethical problems. Although privacy and data protection are not synonymous, the fundamental privacy concern in AI ethics is informational privacy, and data protection may be regarded as a mechanism to preserve informational privacy. Data security is jeopardised by AI based on machine learning. On the one hand, it requires enormous data sets for training, and access to those data sets may generate data protection concerns. The fact that AI and its capacity to recognise patterns may offer privacy problems, even when no direct access to personal data is feasible, is more fascinating and more particular to AI (Stahl, 2021).

According to Jagielski et al., 2018 as stated by Stahl, 2021 data security problems are inextricably connected to data protection concerns. Not just for AI, but for all ICT, cybersecurity is a constant issue. AI systems, on the other hand, may be vulnerable to new sorts of security flaws, such as model poisoning attacks.

Machine learning systems' outputs are dependent on the quality of the training data, which can be difficult to determine. Data integrity can be jeopardised by security breaches, as well as technological or organisational issues. This means that machine learning systems' dependability may need to be examined differently than other types of systems, which might be an ethical concern if the system's output has an ethical value. For example, an AI system used in pathology to identify disease signs may operate well under research settings, with well-labelled training data, and perform on par with, if not better than, a skilled pathologist. This does not guarantee that a system using the same model would perform similarly in clinical scenarios, which may be one of the reasons why, despite AI's enormous promise for medicine, very few AI systems are now in use in clinical settings as stated by Topol, 2019 according to (Stahl, 2021).

Machine learning systems are, by definition, opaque, or at least not in the same sense that other ICT systems are. The commercial secrecy of algorithms and models may further limit transparency in private systems. Although the term "transparency" is debatable, a lack of openness creates issues of accountability. It is more difficult to recognise and resolve issues of bias and discrimination when there is a lack of openness according to USACM, 2017 (Stahl, 2021).

According to CDEI, 2019 as stated by Stahl, 2021 bias is a widely discussed ethical issue in AI. One significant difficulty is that machine learning systems might unintentionally or purposefully reproduce existing biases. There are several high-profile examples of such scenarios, such as when machine learning is used to duplicate gender prejudices in recruiting or when machine learning is used to perpetuate racial disparities in probation processes as stated by Raso et al., 2018 according to Stahl, 2021. Discrimination based on specific (often referred to as protected) qualities is not just an ethical concern, but it has long been recognised as a violation of human rights, and it is thus unlawful in many jurisdictions. Because AI poses a threat to this fundamental right, attention has been drawn to the potential for machine learning to infringe on the right to equality and non-discrimination as stated by Access Now Policy Team, 2018 (Stahl, 2021).

According to BmVI, 2017 and as stated by Stahl, 2021 safety is another important ethical concern in machine learning, especially in systems that interact directly with the physical environment, such as autonomous cars or systems that manage crucial healthcare provisions. While not now prominent in the public discourse, safety will undoubtedly become more significant when machine-learning-enabled technologies begin to interact physically with humans in greater numbers.

### Legal concerns

The below legal concerns regarding AI have been studied by Dr Rowena Rodrigues extensively. Dr Rowena Rodrigues has a PhD from the University of Edinburgh and is a Senior Research Manager (Policy, ethics, and emerging technology) at Trilateral Research Ltd: London, London, GB

Artificial intelligence (AI) law and human rights concerns are being studied and argued, as well as how they are being addressed, gaps and obstacles, and how they influence human rights concepts. Algorithmic transparency, cybersecurity vulnerabilities, unfairness, bias, and discrimination, lack of contestability, legal personhood issues, intellectual property issues, negative effects on workers, privacy and data protection issues, liability for damage, and lack of accountability are some of the issues that need to be addressed (Rodrigues, 2020).

#### Lack of algorithmic transparency

The lack of algorithmic transparency (Bodo et al., 2018; Coglianese & Lehr, 2018; Lepri et al., 2018 according to Rodrigues, 2020) is a major concern that is at the forefront of legal debates around AI (EDPS, 2016; Pasquale, 2015 as stated by Rodrigues, 2020). Given the widespread use of AI in high-risk domains, Cath, 2018 notes that demand is rising to design and control AI in a way that is accountable, fair, and transparent (Rodrigues, 2020). The lack of algorithmic transparency is problematic; as Desai and Kroll, 2017 explain according to Rodrigues, 2020 why by citing cases of people who were denied employment, loans, no-fly lists, or benefits without understanding why it happened other than the decision was processed by some programme. The difficulty is exacerbated by the fact that information on the functionality of algorithms is sometimes purposely poorly accessible (Mittelstadt et al., 2016 according to Rodrigues, 2020).

#### Cyber security vulnerabilities

According to Rodrigues, 2020 paper by Osoba and Welser, 2017 states that the use of AI weapons without human mediation; issues related to AI vulnerabilities in cyber security; how the use of AI for surveillance or cyber security for national security opens a new attack vector based on the "data diet vulnerability"; the use of network intervention methods b  it also cites domestic security-related challenges, such as governments' (increasing) use of artificial agents for citizen monitoring (e.g., predictive policing algorithms). These have been criticised for their potential to jeopardise citizens' basic rights, according to Couchman, 2019 as stated by Rodrigues, 2020. Such challenges are important because they expose key infrastructures to threats that have serious consequences for society and people, posing a threat to life and human security, as well as access to resources. Cyber security flaws are also a serious hazard since they are frequently disguised and found only after it is too late (after the damage is caused) (Rodrigues, 2020).

#### Unfairness, bias, and discrimination

Unfairness (As stated by Smith, 2017 accoutring to Rodrigues, 2020), bias (As stated by Courtland, 2018 accoutring to Rodrigues, 2020), and discrimination (As stated by Smith, 2017 accoutring to Rodrigues, 2020) are all issues that have been identified as a major challenge (As stated in Hacker, 2018 accoutring to Rodrigues, 2020) in the use of algorithms and automated decision-making systems, such as in health (As stated by Danks & London, 2017 accoutring to Rodrigues, 2020), employment, credit, criminal justice (As stated by Berk, 2019 accoutring to Rodrigues, 2020), and insurance (As stated by Smith, 2017 accoutring to Rodrigues, 2020). Protests and legal challenges are likely in August 2020 over the implementation of a contentious assessment algorithm to award scores to GCSE students in England (As stated by Ferguson & Savage 2020 accoutring to Rodrigues, 2020).

In the United Kingdom, the application of machine learning algorithms to assist police decision-making is still in its early stages, and there is a paucity of research on how using an algorithm affects officers' decision-making in practice. Furthermore, there is a paucity of research on the usefulness and efficiency of various systems, as well as their cost-effectiveness, influence on individual rights, and amount to which they fulfil legitimate policing goals. There is a dearth of explicit advice and standards of practice that outline the acceptable limits that police departments should use when testing predictive algorithmic technologies. Procurement contracts between the police and private sector providers of predictive policing technologies have several difficulties. It is proposed that all relevant public procurement agreements for machine learning algorithms include a requirement for the supplier to be able to retroactively deconstruct the algorithm to determine which factors influenced the model's predictions, as well as a requirement for the supplier to be able to provide an expert witness who can provide details about the algorithm's operation if needed, such as in an evidential context (Babuta, Oswald., & Rinik, 2018).

#### Lack of contestability

Individuals have the right to dispute and request a review of automated decision-making that materially impacts their rights or legitimate interests under European Union data protection law (As stated in GDPR, 2016/679 according to Rodrigues, 2020). Data subjects have the right to object at any time to the processing of personal data on them that is based on tasks carried out in the public interest or legitimate interests, on grounds relating to their circumstances. Furthermore, under Article 22(3) GDPR, data controllers must take reasonable steps to protect a data subject's rights, freedoms, and legitimate interests, including the right to request human intervention from the controller, to voice their views, and to dispute the decision. However, Hildebrandt, 2016 points out that ML systems' opacity may lessen both the responsibility of their ‘owners' and the contestability of their choices (Rodrigues, 2020). The lack of contestability - regarding algorithmic systems, i.e., the lack of a clear mechanism to question them when they yield unexpected, detrimental, unjust, or discriminating results - is highlighted by Edwards and Veale, 2017 (Rodrigues, 2020).

#### Legal personhood issues

The question of whether AI (and/or robotics systems) fit within current legal categories or if a new category with its distinctive features and implications is still being debated. (As stated by Resolution of the European Parliament, dated February 16, 2017, according to Rodrigues, 2020).

#### Intellectual property issues

The Universal Declaration of Human Rights (UDHR, Article 27), the International Covenant on Economic, Social, and Cultural Rights (ICESCR, Article 15), the International Covenant on Civil and Political Rights (ICCPR, Article 19), and the Vienna Declaration and Programme of Action (VDPA) 1993 all include intellectual property rights (Rodrigues, 2020). They have a "human rights character" and "have become contextualised in diverse policy areas," according to WIPO, 1998 as stated by Rodrigues, 2020. AI introduces several intellectual property concerns, such as who owns AI-generated or created works or innovations. Should the innovations of artificial intelligence be considered previous art? Who owns the data collection from which a machine learning algorithm must learn? Who should be held accountable for AI-generated creativity and invention if it infringes on others' rights or violates other legal provisions? (As stated by Source: CEIPI, undated according to Rodrigues, 2020).

#### Adverse effects on workers

According to Rodrigues, 2020 the influence of AI and robots on the workplace is highlighted in the IBA Global Employment Institute study, 2017 and seen as a global concern. Changes in future employee requirements, lower demand for workers, labour relations, creation of new job structures and new types of jobs, dismissal of employees, inequality in the "new" job market, integration of untrained workers in the "new" job market, labour relations (and its possible implications for union activities and collective bargaining aspects, challenges for employee representatives, changes in the structure of the workforce The possible loss of worker autonomy is also significant, according to Frontier Economics, 2018 (Rodrigues, 2020). These concerns have major economic (e.g., poverty) and social (e.g., homelessness, displacement, violence, despair) human rights implications. They create ethical questions and challenges that are difficult to answer but must be addressed (Rodrigues, 2020).

#### Privacy and data protection issues

In addition to compromising other rights, legal academics, and data protection enforcement bodies (as stated by CNIL 2017; ICO 2017 according to Rodrigues, 2020) consider that AI creates significant privacy and data protection problems according to Gardner, 2016 as stated by Rodrigues, 2020. Informed consent and monitoring are examples of this. According to Brundage, 2018 as stated by Rodrigues, 2020, a breach of people's data protection rights (e.g., right to access to personal data, right to prohibit processing that is likely to cause harm or distress, right not to be subjected to a decision based exclusively on automated processing, and so on).

#### Liability for damage

People and property may be harmed as a result of the deployment and usage of AI technology. For example, Gluyas and Day, 2018 give various examples, such as autonomous autos driving over pedestrians, a partially controlled drone crashing and causing harm, and an AI software programme incorrectly diagnosing medical therapy (Rodrigues, 2020). They go on to say  “As there are many parties involved in an AI system (data provider, designer, manufacturer, programmer, developer, user and AI system itself), liability is difficult to establish when something goes wrong and there are many factors to be taken into consideration…”(Rodrigues, 2020).

#### Lack of accountability for harms

American and European governments today appear to be differing on how to solve present accountability gaps in AI, write Wachter, Mittelstadt, and Floridi, 2017 according to Rodrigues, 2020. Legal accountability methods for AI abuses might include a 'right to explanation,' data protection and information and transparency protections, audits, or other reporting duties, according to Edwards Veale, 2017 (Rodrigues, 2020). Doshi-Velez et al., 2017 examine the legal scenarios in which explanation is now necessary, as well as the technological factors that must be considered if AI systems are to deliver the types of explanations that people are currently obligated to provide (Rodrigues, 2020).

### Practical concerns

Making sure this project follows legal, social, and ethical concerns the British computing society code of conduct was looked at. The BCS Code of Conduct is a one-of-a-kind and significant validation of your integrity as well as an IT professional's code of ethics ("BCS Code of Conduct", 2022). The code is based on four main principles which are as follows ("BCS Code of Conduct", 2022):

#### [You make IT for everyone](https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/)

You want everyone to have access to IT as you work together to address challenges in your field and society. You share what you know, maintain high standards, and always act professionally and fairly.

PUBLIC INTEREST

You shall:

1. Respect public health, privacy, security, and the well-being of others, as well as the environment.
2. Take into account the lawful rights of third parties.
3. Conduct your professional activities without regard to sex, sexual orientation, marital status, nationality, colour, ethnicity, ethnic origin, religion, age, handicap, or any other condition or requirement.
4. Support equitable access to the advantages of information technology, and wherever possibilities occur, aim to promote the inclusion of all segments of society.

#### [Show what you know, learn what you don't](https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/)

You have integrity and demonstrate expertise, but you recognise that you don't know everything, which is why you continue to study and improve and never take on assignments for which you lack the necessary abilities and resources.

PROFESSIONAL COMPETENCE AND INTEGRITY

You shall:

1. Only do work or provide services that are within your professional capabilities.
2. NOT claiming any degree of expertise that you don't have.
3. keep awareness of technical changes, processes, and standards that are important to your field; increase your professional knowledge, abilities, and competence continuously.
4. ensuring that you have a thorough awareness of the law and that you follow it when doing your professional duties.
5. seek, accept, and provide honest critiques of work; appreciate and value opposing ideas; and seek, accept, and offer honest criticisms of work.
6. prevent causing harm to others, their property, reputation, or employment via deceptive, malevolent, or careless behaviour or inaction.
7. Any offer of bribery or unethical inducement will be rejected, and no such offer will be made.

#### [Respect the organisation or individual you work for](https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/)

You operate with care and devotion, always acting in the best interests of your customer or organisation. While preserving discretion and ethical standards, you accept personal and group accountability for your activities.

DUTY TO RELEVANT AUTHORITY

You shall:

1. carry out your professional obligations with care and effort in compliance with the standards of the competent authorities, while always exercising your professional judgement.
2. Try to stay away from any circumstance that might lead to a conflict of interest between you and your appropriate authorities.
3. Accept professional responsibility for your work and the work of colleagues who are identified as operating under your supervision in a specific setting.
4. Except with the approval of your relevant authority or as required by law, do not divulge or authorise the disclosure of sensitive information, or use it for personal advantage or to benefit a third party.
5. NOT falsify or conceal information about the performance of products, systems, or services (unless legally obligated to keep such information private), or take advantage of others' lack of relevant expertise or experience.

#### [Keep IT real. Keep IT professional. Pass IT on](https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/)

You are an ambassador for the IT industry as a BCS member, and you use your voice to help promote it favourably throughout the world. You assist your IT colleagues and other team members in their personal and professional development.

**DUTY TO THE PROFESSION**

You shall:

1. Accept your responsibility to protect the profession's reputation and refrain from doing any actions that can jeopardise it.
2. Participate in the establishment, use, and enforcement of professional standards to enhance them.
3. maintain the BCS, The Chartered Institute for IT, reputation, and good status.
4. In your professional connections with all members of BCS and members of other professions with whom you work in a professional capacity, operate with honesty and respect.
5. Members should be encouraged and supported in their professional growth.

### Legal and social, ethical, and practical concerns concerning this project

This project aims to not cross ethical or social boundaries. This project had many ethical concerns due to keeping the privacy of Twitter users in mind. The dataset used in this research was publicly available and has aspects that could be sensitive information relating to a Twitter user removed or replaced. so, this research to the best of my knowledge does not cross any legal boundaries. This project was completed following the ethos of the British computing society.

# References

1. 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.
2. Almalki, K., Alharbi, O., Al-Ahmadi, W., & Aljohani, M. (2020, July). Anti-procrastination online tool for graduate students based on the pomodoro technique. In International Conference on Human-Computer Interaction (pp. 133-144). Springer, Cham.
3. Andrei, B. A., Casu-Pop, A. C., Gheorghe, S. C., & Boiangiu, C. A. (2019). A study on using waterfall and agile methods in software project management. Journal Of Information Systems & Operations Management, 125-135.
4. 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>
5. auto-sklearn — AutoSklearn 0.14.7 documentation. Automl.github.io. (2022). Retrieved 24 April 2022, from <https://automl.github.io/auto-sklearn/master/>.
6. Babuta, A., Oswald, M., & Rinik, C. (2018). Machine learning algorithms and police decision-making: legal, ethical and regulatory challenges.
7. BCS Code of Conduct. Bcs.org. (2022). Retrieved 14 June 2022, from https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/.
8. 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).
9. 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.
10. Drus, Z., & Khalid, H. (2019). Sentiment Analysis in Social Media and Its Application: Systematic Literature Review. Procedia Computer Science, 161, 707-714. https://doi.org/10.1016/j.procs.2019.11.174
11. Elbagir, S., & Yang, J. (2018). Sentiment Analysis of Twitter Data Using Machine Learning Techniques and Scikit-learn. Proceedings Of The 2018 International Conference On Algorithms, Computing And Artificial Intelligence. https://doi.org/10.1145/3302425.3302492
12. 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>
13. 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>
14. Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., & Hutter, F. (2020). Auto-sklearn 2.0: Hands-free automl via meta-learning. arXiv preprint arXiv:2007.04074.
15. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
16. Hate crime | The Crown Prosecution Service. Cps.gov.uk. (2022). Retrieved 15 April 2022, from https://www.cps.gov.uk/crime-info/hate-crime.
17. 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>.
18. Hota, S., & Pathak, S. (2018). KNN classifier based approach for multi-class sentiment analysis of twitter data. International Journal Of Engineering &Amp; Technology, 7(3), 1372. <https://doi.org/10.14419/ijet.v7i3.12656>
19. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science &amp; Engineering, 9(3), 90–95.
20. 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
21. Iftikhar, S. (2021). Investigation in using machine learning algorithms to detect if a tweet has a toxic language. Coventry: Coventry Univeristy.
22. 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.
23. Joshi, A., Bhattacharyya, P., & Carman, M. (2017). Automatic Sarcasm Detection. ACM Computing Surveys, 50(5), 1-22. https://doi.org/10.1145/3124420
24. 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
25. Kumar, G., & Bhatia, P. K. (2012). Impact of agile methodology on software development process. International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(4), 46-50.
26. 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
27. Malik, P., Aggrawal, A., & Vishwakarma, D. (2021). Toxic Speech Detection using Traditional Machine Learning Models and BERT and fastText Embedding with Deep Neural Networks. 2021 5Th International Conference On Computing Methodologies And Communication (ICCMC). https://doi.org/10.1109/iccmc51019.2021.9418395
28. Maslach, C., & Leiter, M. (2017). Understanding Burnout. The Handbook Of Stress And Health, 36-56. https://doi.org/10.1002/9781118993811.ch3
29. Maylor, H. (2001). Beyond the Gantt chart:. European Management Journal, 19(1), 92-100. https://doi.org/10.1016/s0263-2373(00)00074-8
30. McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).
31. Mejova, Y. (2009). Sentiment analysis: An overview. University of Iowa, Computer Science Department (p. 5-6).
32. Moczydłowska, J. (2016). Organisational reasons of job burnout. Ekonomia I Zarzadzanie, 8(2), 7-12. https://doi.org/10.1515/emj-2016-0011
33. 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
34. Navigli, R. (2009). Word sense disambiguation. ACM Computing Surveys, 41(2), 1-69. https://doi.org/10.1145/1459352.1459355
35. Ortigosa, A., Martín, J., & Carro, R. (2014). Sentiment analysis in Facebook and its application to e-learning. Computers In Human Behavior, 31, 527-541. https://doi.org/10.1016/j.chb.2013.05.024
36. Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.
37. 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).
38. Prakash, T. N., & Aloysius, A (2020). A Comparative study of Lexicon based and Machine learning based classifications in Sentiment analysis.
39. 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
40. Qiao, Y., Zhang, B., & Zhang, W. (2020). Malware Classification Method Based on Word Vector of Bytes and Multilayer Perception. ICC 2020 - 2020 IEEE International Conference On Communications (ICC). https://doi.org/10.1109/icc40277.2020.9149143
41. Rupert, P., & Morgan, D. (2005). Work Setting and Burnout Among Professional Psychologists. Professional Psychology: Research And Practice, 36(5), 544-550. https://doi.org/10.1037/0735-7028.36.5.544
42. Rustam, F., Khalid, M., Aslam, W., Rupapara, V., Mehmood, A., & Choi, G. (2021). A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. PLOS ONE, 16(2), e0245909. https://doi.org/10.1371/journal.pone.0245909
43. 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
44. 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
45. Schmidt, J., Marques, M., Botti, S., & Marques, M. (2019). Recent advances and applications of machine learning in solid-state materials science. Npj Computational Materials, 5(1). https://doi.org/10.1038/s41524-019-0221-0
46. 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.
47. sklearn.ensemble.GradientBoostingClassifier. scikit-learn. (2022). Retrieved 25 April 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.
48. sklearn.qda.QDA — scikit-learn 0.15-git documentation. Scikit-learn.org. (2022). Retrieved 25 April 2022, from https://scikit-learn.org/0.15/modules/generated/sklearn.qda.QDA.html.
49. Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.
50. Stahl, B. (2021). Ethical Issues of AI. Springerbriefs In Research And Innovation Governance, 35-53. https://doi.org/10.1007/978-3-030-69978-9\_4
51. Trevelyan, L. (2022). Cyberbullying and the law - InBrief.co.uk. InBrief.co.uk. Retrieved 30 May 2022, from https://www.inbrief.co.uk/offences/cyberbullying/.
52. Rodrigues, R. (2020). Legal and human rights issues of AI: Gaps, challenges and vulnerabilities. Journal Of Responsible Technology, 4, 100005. https://doi.org/10.1016/j.jrt.2020.100005
53. Valle-Cruz, D., López-Chau, A., & Sandoval-Almazán, R. (2020). Impression analysis of trending topics in Twitter with classification algorithms. Proceedings Of The 13Th International Conference On Theory And Practice Of Electronic Governance. https://doi.org/10.1145/3428502.3428570
54. Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.
55. Wang, X., Ounis, I., & Macdonald, C. (2020). Negative Confidence-Aware Weakly Supervised Binary Classification for Effective Review Helpfulness Classification. Proceedings Of The 29Th ACM International Conference On Information &Amp; Knowledge Management. https://doi.org/10.1145/3340531.3411978
56. Warmsley, D. (2017). On the Detection of Hate Speech, Hate Speakers and Polarized Groups in Online Social Media.
57. Winter, G. (2020). How other professions cope with burnout. The Bulletin Of The Royal College Of Surgeons Of England, 102(6), 252-255. https://doi.org/10.1308/rcsbull.2020.172
58. Wojcik, S., & Hughes, A. (2019). Sizing up Twitter users. PEW research center, 24.
59. Yanting, M. (2022). Analysis on the Reasons of Academic Burnout Among Chinese Adolescents (Revised). Journal Of Psychology Research, 12(1). https://doi.org/10.17265/2159-5542/2022.01.001