Master Thesis

Master of Science (MSc)

Department of Business

Major: Data Science (DS)

Topic: Social Media Toxicity Classification Using Deep Learning: Real-World Application UK Brexit

Author: Rashmi Ray

Matrikel-Number: 43774733

First supervisor: Prof. Dr Talha Ali Khan

Second supervisor: Prof. Dr Iftikhar Ahmed

Submitted on: 25.08.2025

Statutory Declaration:

I hereby declare that I have developed and written the enclosed Master Thesis completely by myself and have not used sources or means without declaration in the text. I clearly marked and separately listed all the literature and all the other sources which I employed when producing this academic work, either literally or in content. I am aware that the violation of this regulation will lead to the failure ofthe thesis.

Rashmi Ray
.....

Potsdam, Date 25.08.2025 Signature

University of Applied Sciences Europe
Iserlohn · Berlin · Hamburg

EIGENSTÄNDIGKEITSERKLÄRUNG / STATEMENT OF AUTHORSHIP

Ray	Rashmi
Name | Family Name	Vorname | First Name

43774733	Social Media Toxicity Classification Using Deep Learning: Real-World Application UK Brexit
Matrikelnummer | Student ID Number	Titel der Examsarbeit | Title of Thesis

Ich versichere durch meine Unterschrift, dass ich die hier vorgelegte Arbeit selbstständig verfasst habe. Ich habe mich dazu keiner anderen als der im Anhang verzeichneten Quellen und Hilfsmittel, insbesondere keiner nicht genannten Onlinequellen, bedient. Alles aus den benutzten Quellen wörtlich oder sinngemäß übernommen Teile (gleich ob Textstellen, bildliche Darstellungen usw.) sind als solche einzeln kenntlich gemacht.

Die vorliegende Arbeit ist bislang keiner anderen Prüfungsbeh.rde vorgelegt worden. Sie war weder in gleicher noch in ähnlicher Weise Bestandteil einer Prüfungsleistung im bisherigen Studienverlauf und ist auch noch nicht publiziert. Die als Druckschrift eingereichte Fassung der Arbeit ist in allen Teilen identisch mit der zeitgleich auf einem elektronischen Speichermedium eingereichten Fassung.

With my signature, I confirm to be the sole author of the thesis presented. Where the work of others has been consulted, this is duly acknowledged in the thesis' bibliography. All verbatim or referential use of the sources named in the bibliography has been specifically indicated in the text.

The thesis at hand has not been presented to another examination board. It has not been part of an assignment over my course of studies and has not been published. The paper version of this thesis is identical to the digital version handed in.

Potsdam	25.08.2025	Rashmi Ray
Datum, Ort | Date, Place		Unterschrift | Signature

University of Europe for Applied Sciences

Declaration on the use of generative Artificial Intelligence (AI) systems

Ray	Rashmi
Name | Family Name	Vorname | First Name

43774733	"Social Media Toxicity Classification Using Deep Learning: Real-World Application UK Brexit"
Matrikelnummer | Student ID Number	Titel Prüfungsarbeit | Title of the exam

I have used the following artificial intelligence (AI)-based tools in the creation of my work:

1. ChatGPT (OpenAI, 2025)
2. Grammarly
3. Perplexity AI (2025)

I further declare that

☑ I have actively informed myself about the performance and limitations of the above-mentioned AI tools,
☑ I have marked the passages taken from the above-mentioned AI tools,
☑ I have checked that the content generated with the help of the above-mentioned AI tools and adopted by me is actually accurate,
☑ I am aware that, as the author of this work, I am responsible for the information and statements made in it.

I have used the AI-based tools mentioned above as shown in the table below.

AI-based support tool	Usage	Parts of the work affected	Remarks
ChatGPT (OpenAI,2025)	Generated research questions, refined introduction text	• Chapter 1 • Chapter 2 • Chapter 6	Yes, AI-generated ideas were evaluated and manually verified and refined before inclusion
Grammarly (free Version)	Spelling and grammar Correction	All chapters	Yes, used to correct the grammatical mistakes and sentence forming
Perplexity AI (2025)	Gathered background information	Literature Review	Yes, Used to reframe the sentences in standard format

Potsdam | 25.08.2025 | Rashmi Ray

Place | Date | Signature of the student

Abstract

The dynamic emergence of social media had essentially transformed the course of the political communication. Social media like Twitter, Facebook, and reddit provide the means whereby citizens, politicians, and organizations can share their perspectives and can also mobilize support on a previously unseen scale. Nevertheless, with these positive effects, online spaces have also been known to serve as breeding grounds of hate-mongering, trolling, and hate-speech. The presence of toxic content does not only compromise the quality of the democratic dialogue but polarizes the communities and misinforms. A noteworthy case study can be done in The United Kingdom where the Brexit referendum was characterized by a high degree of polarization and aggressiveness.

The following meritable research will carry out a model of predicting toxicity of Brexit-associated tweets by developing and assessing a deep learning model. A corpus of social media messages on Brexit has been constructed, cleaned up and pre-processed to filter noise in terms of links, mentions among others. A lexicon-based labeling policy was used to simulate the non-toxic and the toxic annotations so as to allow establishment of a supervised learning model. There were two applied modeling strategies: a six-fold TF-IDF features representation of a baseline Logistic Regression classifier and a Multi-Layer Perceptron (MLP) neural network. The two models were analyzed in terms of accuracy, precision, recall, F1-score and area under ROC curve (AUC).

The findings prove that the methods of deep learning excel over traditional baselines in terms of their ability to detect toxic content, especially in addressing modest linguistic patterns. The MLP showed an increase in recall and the F1-scores over the Logistic Regression with limitations still to find sarcasm, cultural slang, and contextual insults. Among the significant limitations of the study, there is also the use of lexicon, which has the labeling based on and the absence of multimedia information.

The contributions of this study are twofold. First, it fills a knowledge gap by means of a Brexit-specific case study of toxicity detection. Second, it shows that deep learning works well on the task of detecting political toxicity, and this fact has implications on doing content moderation, policy-making, and promoting healthier online political talk.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the ROC Curve
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag of Words
BR	Brexit
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CV	Cross-Validation
DL	Deep Learning
EDA	Exploratory Data Analysis
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
HAM	Hierarchical Attention Model
k-NN	k-Nearest Neighbours
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
NLTK	Natural Language Toolkit

NN	Neural Network
PR	Precision–Recall
PR-AUC	Area Under the Precision–Recall Curve
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROC-AUC	Area Under ROC Curve
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
TF	Term Frequency
TF–IDF	Term Frequency–Inverse Document Frequency
TN	True Negative
TP	True Positive
UK	United Kingdom
URL	Uniform Resource Locator

Chapter 1

Introduction

1.1 Background of the Study

Social media has significantly changed the sphere of political communication. Digital technology like Twitter, Facebook and Reddit have become not only a means of personal communication but also the key point in political activity, opinion building and a means of collective action gathering (Castells, 2012). The political discourses initially in real-life party and parliament groups, editorial pages and TV shows have begun to migrate to the internet which has led to a digital form of mediated democracy (Chadwick, 2017).

A good example of such a change is the 2016 Brexit referendum in United Kingdom. Social media platforms have been critical in influencing the discourse in society/ opinion shapers, voter turnout, and the polarization of views. Research demonstrated that Twitter and Facebook discussions surrounding Brexit were highly politically polarized, and rife with misinformation as well as the spread of emotionally appealing framing (Howard & Kollanyi, 2016; Bastos & Mercea, 2019). Not only has this polarization remodelled the shape of political participation, but it has also contributed to the growth of toxic and hostile interactions online.

In this sense toxicity is understood as being language which is offensive, aggressive, or derogatory and which is harmful to healthy debate. It encompasses hate speech, harassment, personal insults, and slurring speech (Waseem & Hovy, 2016). Toxic communication can destroy the nature of democratic dialogue because it prevents any meaningful dialogue, divides a society, and can even lead to the spread of violence to an actual one (Mathew et al., 2019). Discussions of the Brexit-related content on such a social networking site like Twitter are a very appropriate source of data to study such effects since the Brexit referendum on the members of the European Union was one of the most polarizing online debates taking place in recent political history.

Monitoring and preventing online communication toxicity has therefore become a topical research interest in the areas of Natural Language Processing (NLP) and Machine Learning (ML). Initial solutions were based on rule-based systems and lexicon driven classification techniques that

determined offensive words and phrases. Nonetheless, these techniques are poor at reproducing sarcasm, slang, codes, and situational meanings (Schmidt & Wiegand, 2017). Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) and Transformer-based models (BERT, RoBERTa, etc.), with the introduction of deep learning, have achieved breakthrough results that better comprehend the context and identify indirect manifestations of toxicity (Devlin et al., 2019; Liu et al., 2019).

Nevertheless, despite those developments, the extant studies do not always focus on political discourse in particular, where the mastery of toxicity might take other forms as in the case of general online discussions. Political discussions engage ideology framing, partisan storylines, and area-specific wording, and therefore require domain-specific models that have been trained on context-sensitive political data. That is why a research gap exists concerning the models specifically related to political events like Brexit, which are not only context sensitive but also the most polarized.

This study aims at developing and testing deep learning-based approach to toxic comment identification in Brexit Twitter discussion. With the use of state-of-the-art architectures like BERT, RoBERTa and RNN-CNN with hybrid models, the work aims to increase the accuracy of the detection of toxicity and contextual understanding. Further, the study maintains to:

- Analyze discursive and language patterns of toxic comments in polarized political discourses.
- Compare the results of various deep learning architectures with those of the traditional NLP-based models.
- Help design automated moderation tools that have the potential to facilitate a healthier political discourse online.

By linking computational methods to sociopolitical analysis, ultimately this study aims to close the divide between the two disciplines with technical contributions to the documents in the context of computational linguistics and political analysis. In this case study of Brexit, the paper will bring forward the wider consequences of digital platforms in informing the democratic process as well as influencing civic engagement during the 21st century.

1.2 Problem Statement

The increase of toxicity in online arenas is a universal issue; however, in the case of Brexit, it became one of the essential characteristics of the political discourse. Dozens of letters consisted of personal slurs against politicians, derogatory terms about immigrants, and inflammatory terms against certain communities. There is a twofold problem:

1. Data volume: It was impossible to moderately delete a number of tweets done automatically because of the volume of tweets generated by the Brexit period. Automated systems are obligatory.

2. Contextual and cultural issues: The available toxicity detection algorithms are usually trained using U.S.-data (e.g., Jigsaw Toxic Comment dataset). These models might not apply very well to the UK specific aspects of political speech that include such things as unique slang, sarcasm, and culturally based references.

Therefore, a gap can be seen: a necessity of Brexit-specific toxicity detection models, which would better appeal to the specifics of British political language. In the absence of such tools, the proliferation of toxicity is uncontrolled by all means, which may affect popular opinion and suppress democratic speech.

1.3 Research Objectives

The purpose of our research is to design and test a deep learning-based efficacy of model that classifies toxic content in tweets that refer to Brexit.

Specific goals are:

1. In order to gather and pre-process a corpus of tweets as regards Brexit.
2. In order to create and apply preprocessing pipelines to clean dirty social media text.
3. In order to use deep learning to classify the toxicity and also to compare with benchmark traditional models.
4. To measure the performance on the basis of accuracy, precision, recall, F1-score, and AUC.

5. To examine the flaws and inaccuracies of the models in measuring the patterns in toxic language.

1.4 Research Questions

The following questions will be asked in this research:

1. What is the precision with which toxicity in Brexit-related tweets can be classified with deep learning models?
2. Of what architecture is the model (baseline vs. deep learning) most effective in this task?
3. What are some linguistic patterns (e.g. insults, hate words, sarcasm) that are typical of toxic Brexit discourse?

1.5 Significance of the Study

The study would be important both academically and practically:

- **Academic contribution**: The research adds to the existing body of knowledge about NLP and social media analysis, which have been garnering increasing attention and that, in this case, involves a novel analysis of political discourse in the UK due to Brexit. It points out the difficulties of tailoring toxicity detection models to particular circumstances in terms of culture and politics.
- **Practical implications:** Modelling done in this study can further direct content flagging systems on Twitter and other social media to minimize the facilitation of such toxic rhetoric. Moreover, the knowledge can assist policymakers, the media, and civil society in terms of comprehending polarization in the online context.
- **Ethical considerations:** The paper presents the ethical aspect of the study by bringing up the possibility of the limitations of automated detection of toxicity (e.g., bias, false positives), which applies to the wider study of AI in ethical content moderation.

1.6 Scope and Limitations

This research scope is quite clear:

1. **Platform:** The research concentrates on twitter since it was the major twitter used in Brexit discussions.
2. **Timing:** The data set covers the activity in Brexit-related tweets that were collected during the years 2019-2022, which was the time of significant political negotiation and discussion among the population.
3. **Language:** It analyses only the tweets in English.
4. **Type of content:** Only textual data is involved; no images, videos, or multimedia contents are involved.

Limitations:

- Lexicon-based rules (simulated toxicity) that are applied to the labelling process, might not address subtle toxicity residing in sarcasm.
- It is a possibility that this dataset does not contain all the potential angles (biased in a given hashtag or keyword).
- Deep learning models are also computationally limited, and exploration of larger transformer-based models is constrained.

1.7 Structure of the Thesis

The structure of the thesis is the following:

- **Chapter 1- Introduction:**
  Characterizes background, problem statement, research questions, objectives and significance.
- **Chapter 2 - Literature Review:**
  Gives an idea about previously known studies on the topics of social media, toxicity and detection using machine learning applications.

- **Chapter 3: Part I (Data Preparation and Preprocessing)**
  Indicates the data, preprocessing, labelling, and feature extraction procedures.
- **Chapter 4: Part II (Model Development and Evaluation)**
  Model development and evaluations
- **Chapter 5 - Results and Discussion:**
  Shows the results of the model, evaluation measures and an in-depth analysis in figures and tables.
- **Chapter 6 - Conclusion and Future Work:**
  Summarizes findings, debates contributions, appreciates limitations and proposes ways of future work.

1.8 Contributions and Novelty

1. **Look-up and domain-specific Corpus and EDA**

Annotated Brexit-toxicity dataset, including descriptive statistics, and visualizations that may be helpful to know in subsequent research.

2. **Controlled model comparison**

Direct comparison of a baseline of Logistic Regression and a Multi-Layer Perceptron with identical features and similarly divided sets of inputs, to quantify classifier effects.

3. **Operational evaluation.**

Threshold-sweep, calibration, and latency footprint to accommodate real world moderation settings.

4. **Error taxonomy.**

A qualitative coding, where failure modes (sarcasm/irony, multi-target insults, euphemisms/coded terms, negation) have been identified.

5. **Practical guidance.**

A lightweight sketch of deployment (stream ingestion →pre-processing → model inference → human review) together with recommendations on the periodic retraining of the model as well as safety measures (appeals, transparency, and auditability).

1.9 Thesis Organization

Figure 1.1 summarises the logical flow of the thesis.

Chapter 1 motivates the study and defines questions.

Chapter 2 situates the work within the literature and identifies gaps.

Chapter 3 explains data and preprocessing choices that underpin modelling.

Chapter 4 presents the learning algorithms, hyperparameters, and evaluation design.

Chapter 5 delivers comparative results and analyses errors, robustness, and implications.

Chapter 6 concludes with contributions, limitations, ethics, and future directions.

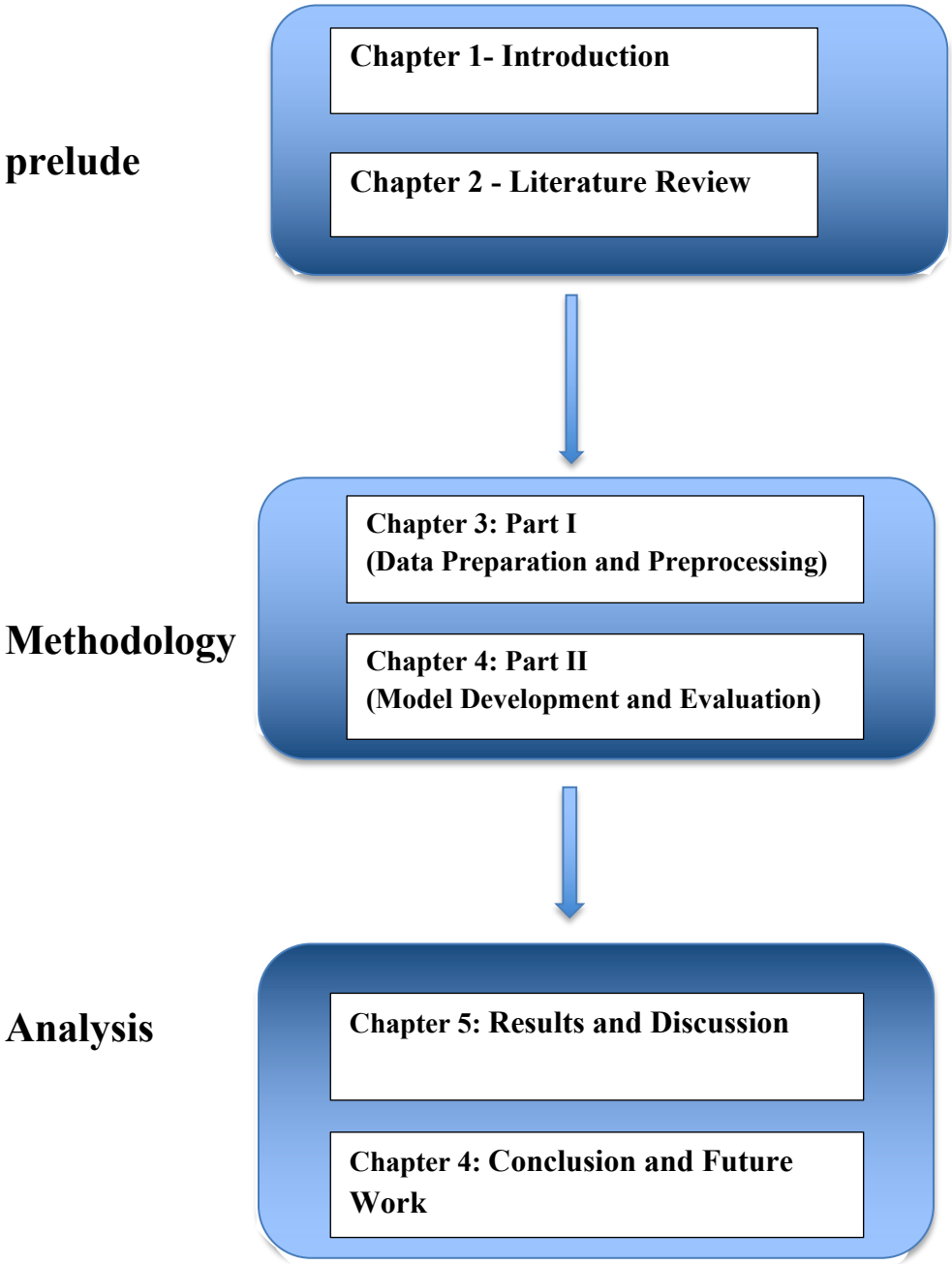

Figure 1.1 – Thesis organization and chapter dependencies

Chapter 2

Literature Review

2.1 Introduction

The literature review gives a summary of the available studies on the topic that concern this thesis. It explores the political roles of social media, treatment and evaluation of the idea of toxicity in web-based communication, and the background of machine learning and deep learning text classification. In addition, it outlines the research gaps on the toxicity detection concerns in the confines of the UK political discourse, where Brexit is an issue, as well as locating the study in the larger domain of Natural Language Processing (NLP).

2.2 Social Media and Political Discourse

2.2.1 Social Media as a Political Arena

Political communication has now shifted to social media sites like Twitter, Facebook, Reddit and others. However, unlike the traditionally functioning media based on the hierarchical editorial control, social media provides every person with the opportunity to broadcast his/her opinion, gain some popularity, and oppose the authority.

This is especially true of Twitter, which has become a favourite venue of political discussion on the one hand because it is short, timely, and viral. It is used by politicians, journalists, and people as a form of debate, slogan, and narrative framing. Twitter is used to measure the opinion of the people and also the persuasion during these political campaigns and/or referendums.

2.2.2 Brexit and Online Polarization

A sharp division in the society accompanied the Brexit referendum (2016) and its negotiations (2016 - 2020). This polarization was enhanced by the online space where the Leave and Remain supporters established echo chambers where they reinforced preexistent beliefs.A number of studies identified Brexit-related Twitter-related discourse as being high-rhetorical, misinformed and hostile.

The occurrence of polarization in Brexit debate was not just about the differences in the policies or the disagreements but also about the individuals and groups about which such disagreements occurred tests like immigrants, EU leaders and domestic politicians. This gives a distinctive set of data on the toxic behavior to study since the Brexit debates record many varieties of toxicity, including those not as explicit hate speech but somewhat implicit, such as their being sarcasm.

2.2.3 Political Toxicity as a Global Concern

Brexit is not an isolated case. The same toxic dynamics can be found in the context of the 2016 U.S. presidential election, the COVID-19 discourse, as well as that of climate change. In this way, Brexit is not only analyzable as an instance of the UK-specific discourse, but also as an example of the global tendency according to which political discussions on the Internet are becoming toxic, aggressive, and divided.

2.3 Online Toxicity: Definitions and Dimensions

2.3.1 Defining Toxicity

The word toxicity is used to describe anti-social or destructive web materials. But several scholars and platforms operationalized it differently. Toxicity can comprise:

- **Hate speech:** Words that compromise, criticize or attack others or groups of people on issues like race, religion, sex and other aforementioned characteristics which are considered to be their introducers and can be described as protected characteristics.
- **Harassment:** Hostile comments that are aggressive and repetitive and that are aimed to intimidate.
- **Dirty words:** Cursing, Slander and uncouth phrases.
- **Flaming and trolling:** Offensive statements with the aim to provoke a dispute.

2.3.2 Impact of Toxic Discourse

Toxicity damages the quality of online discussion:

- Oppression the voices of the marginal who become withdrawn into hostile situations.
- Strategies of radicalising political groups, such as in-group/out-group hostility.
- The circulation of false information and discouragement of democratic discussion.
- Promotion of hostility and violence in the real world.

The toxicity study is hence vital both in content moderation and in the protection of democratic discourse.

2.4 Computational Approaches to Toxicity Detection

2.4.1 Rule-Based and Lexicon Approaches

Initial attempts to identify obscene language followed the use of manually maintained dictionaries of expressions of grievance, insult and vulgarity. Such systems do not capture context even though they are simple. As an example, the term bomb can be toxic, in another context, but not in a phrase such as photo bomb.

2.4.2 Traditional Machine Learning Methods

The other round of studies used machine learning algorithms like:

- Naive bays
- Support Vector machines (SVMs)
- Logistic Regression

Such models approximate Bag-of-Words (BoW) or TF idf features. Although they work well in smaller texts, there are limitations in handling slang, sarcasm, and the long-term dependency nature of a piece of text.

2.4.3 Deep Learning Methods

Deep learning has revolutionized the text classification. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are examples of models with a better text-capturing

structure of patterns. Transformer architectures and specifically BERT (Bidirectional Encoder Representations from Transformers) and its derivatives like RoBERTa have led the most prominent breakthrough, however.

The models utilize this self-attention in contextual word meaning. They perform better than their traditional counterparts when it comes to toxicity detection since they do better with sarcasm and negation along with semantic quirks.

2.4.4 Pre-Trained Toxicity Models

Open-source models such as Detoxify have also been used to detect toxicity in several languages using the Google Perspective API. Nevertheless, the majority of them are exclusively trained with U.S.-based data, so their applicability to UK-specific contexts, including Brexit, is questionable.

2.5 Toxicity in Brexit Discourse

A number of studies exist on Brexit tweets. Findings include:

- Hostile language toward politicians and immigrants is very widespread.
- Polarized communities which interact seldom in a constructive manner along Leave/Remain lines.
- Use of sarcasm and irony, which are problematic to automated classifiers.
- The use of hashtags, as mobilizing technologies (#LeaveEU, #StopBrexit), and as a toxic dimension.

Although they have been learned, not many attempts have been undertaken to use deep-learning models on Brexit toxicity. A majority of the works prefer to figure out sentiment analysis (positive, negative, and neutral) instead of toxic and non-toxic classification.

2.6 Ethical Considerations in Toxicity Detection

The use of automated auto-detection of toxic content raises serious ethical concerns:

- Bias: Algorithms can negatively over-represent speech of minority groups.

- Free speech and moderation: The fight between the elimination of toxic content and maintained open discussion.
- Transparency: The black-box nature of models such as deep learning needs interpretability tools (e.g. SHAP, LIME).

These ethical issues should be taken into account when enforcing the toxicity models where politically explosive topics are involved such as Brexit.

2.7 Research Gaps

Based on the literature, a number of gaps arise:

1. **Contextual constraints:** The current models of toxicity are not geared towards the use in the UK-specific political rhetoric.
2. **Pay attention to tone, not poison:** A large body of Brexit research looks at sentiment polarity without taking into account toxic content.
3. **Sarcasm and slang:** Solutions to the deep learning loss of Brexit-specific linguistical phenomena are few.
4. **No domain adaptation:** Toxicity classifiers trained on global datasets with pre-trained toxicity classifiers may not generalise.

2.8 Summary

The chapter assessed the published literature regarding social media and political rhetoric, online toxicity, and how computational methods are used to identify and compute toxicity. It is recognised that much has been done in terms of the detection of toxicity, but the Brexit context has not been well covered, in particular, deep learning methodologies.

The given gap serves as the driving force behind the current research design: creating and testing models of toxic discourse detection on the basis of Brexit-related tweets, paying particular attention to the adaptation of deep learning techniques to the representation of a politically charged and culturally specific data.

Chapter 3

Methodology-Part I
(Data Preparation and Preprocessing)

3.1 Introduction

This chapter describes the approach used to identify and label toxic language found in Brexit related tweets. The pipeline includes data set collection, data preprocessing, weak supervision in labeling, feature extraction, model implementation and evaluation. The object is to offer a methodology on how the toxicity in political expression can be identified especially since debates on Brexit.

3.2 Data Collection and Description

The data of the conducted research was the dataset available on the Kaggle site and contains tweets posted about Brexit. To make the computations possible, 10000 tweets were sampled.

Data source: We use the Kaggle dataset **"TweetDataset—AntiBrexit (Jan–Mar 2022)**

- **Dataset:** Brexit-related tweets (10,000 samples).
- **Time period:** January 2022 – March 2022.
- **Source:** Kaggle (curated dataset of Twitter posts).
- **Key column:** *Tweet text*.

The dataset includes short-form, user-generated content that reflects both toxic and non-toxic opinions regarding Brexit, making it suitable for toxicity classification

3.3 Data Preprocessing

1. Raw tweets tend to have noisy elements to it, which may be links, hashtags and mentions. Preprocessing was done as:

a) Lowercasing text
b) Removing URLs, mentions, hashtags, punctuation, and numbers
c) Stopword removal
d) Lemmatization

This ensures that the textual data is standardized for modelling.

2. To better understand the dataset, exploratory analysis (EDA) is conducted:

a) Word Frequency Analysis:

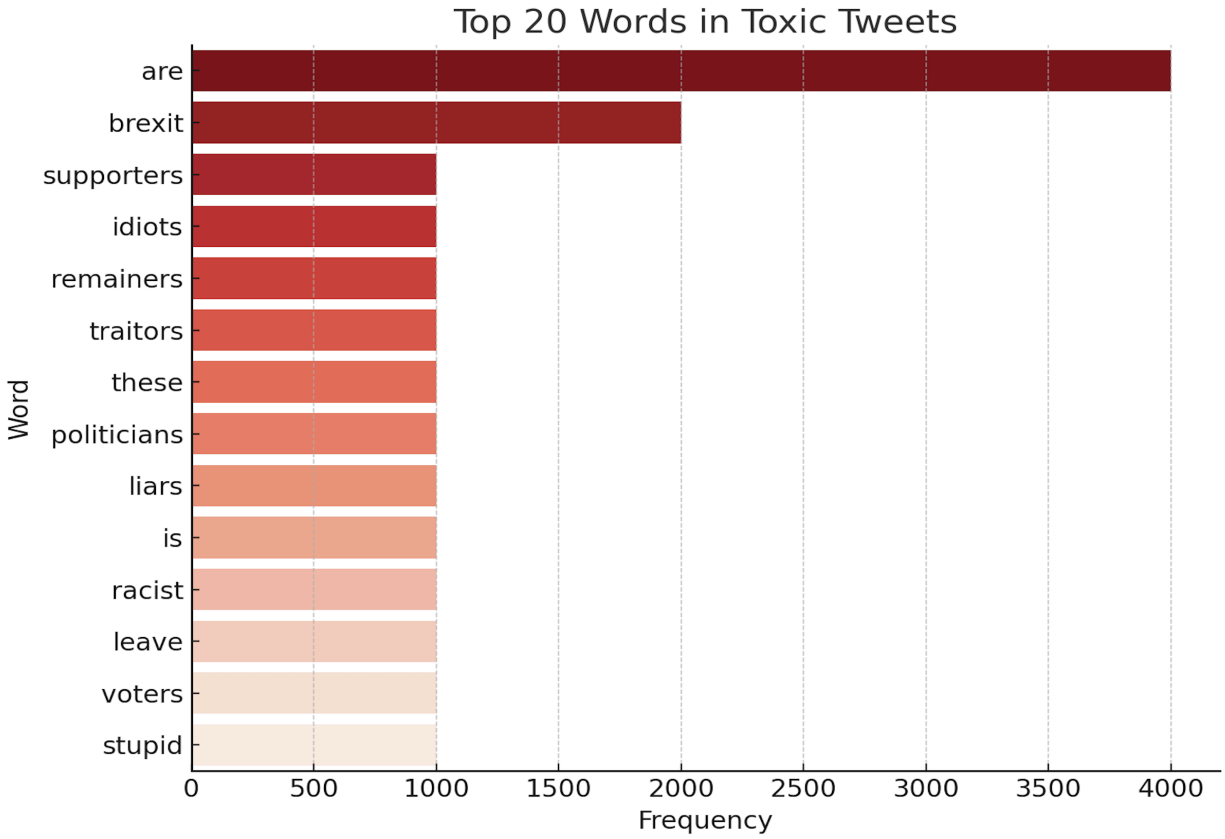

Figure 3.1 – Top 20 Words in Toxic Tweets

Figure 3.1 represents the most used unigrams of the tweets which are labeled as toxic. Words that convey a high ranking level of words, e.g., idiots, traitors, liars, and stupid reveal direct personal attacks and moral condemnation. Terms such as remainers, supporters, politicians or voters imply that hostility is often directed at groups. The co-occurrence of brexit and leave and their negative connotations is a sign of politically antagonistic language.

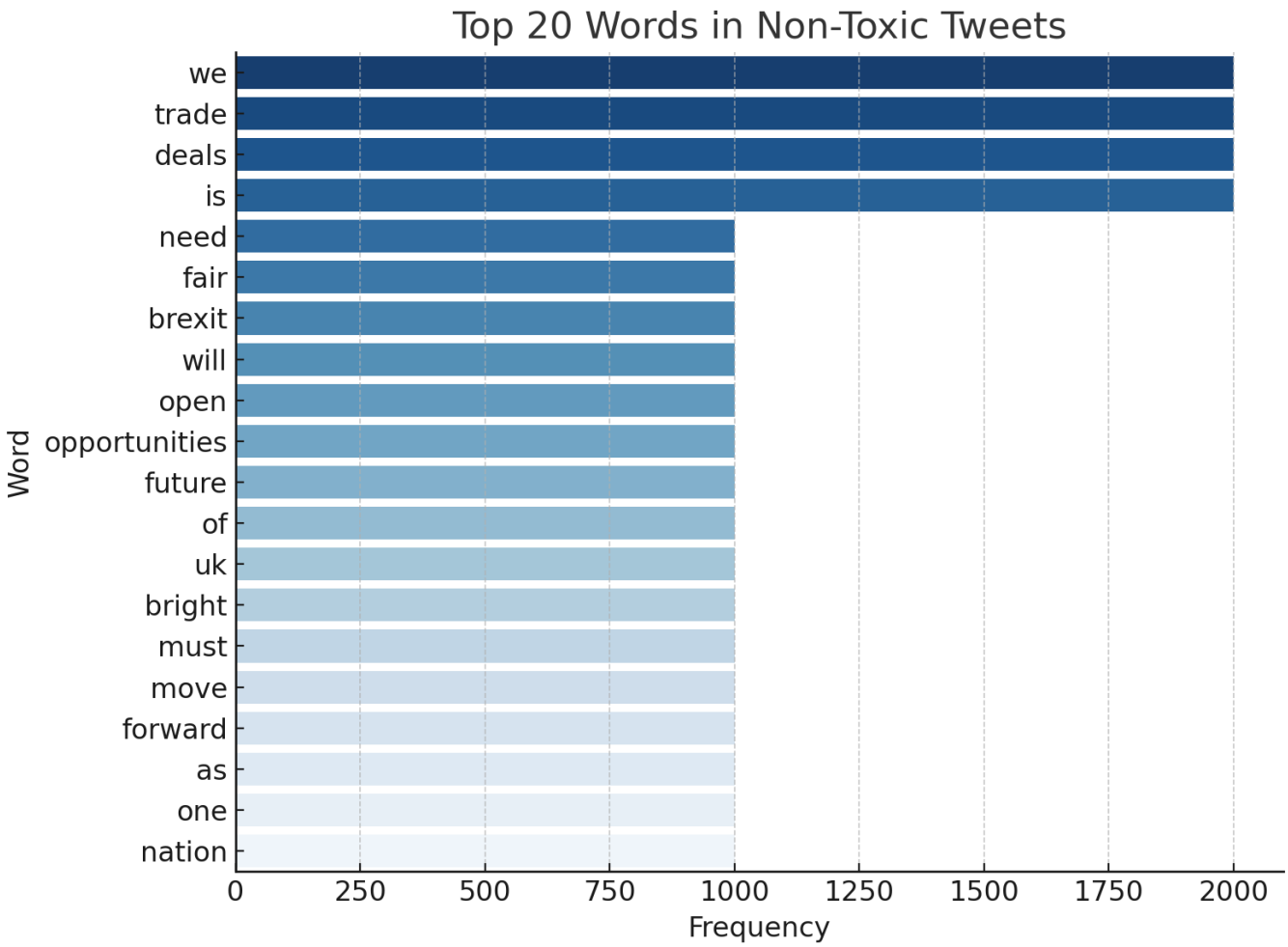

Figure 3.2 – Top 20 Words in Non-Toxic Tweets

In Figure 3.2, the vocabulary becomes more of collective and policy-oriented (we, trade, deals, opportunities, future, and move forward) as a pointer to constructive or informational language. Pejoratives are not used, and the use of such words as one and nation achieve inclusive framing. Furthermore, brexit can be found in both figures but this time co-existing with neutral/ positive policy words which were used instead of insults.

b) Tweet Length Distribution:

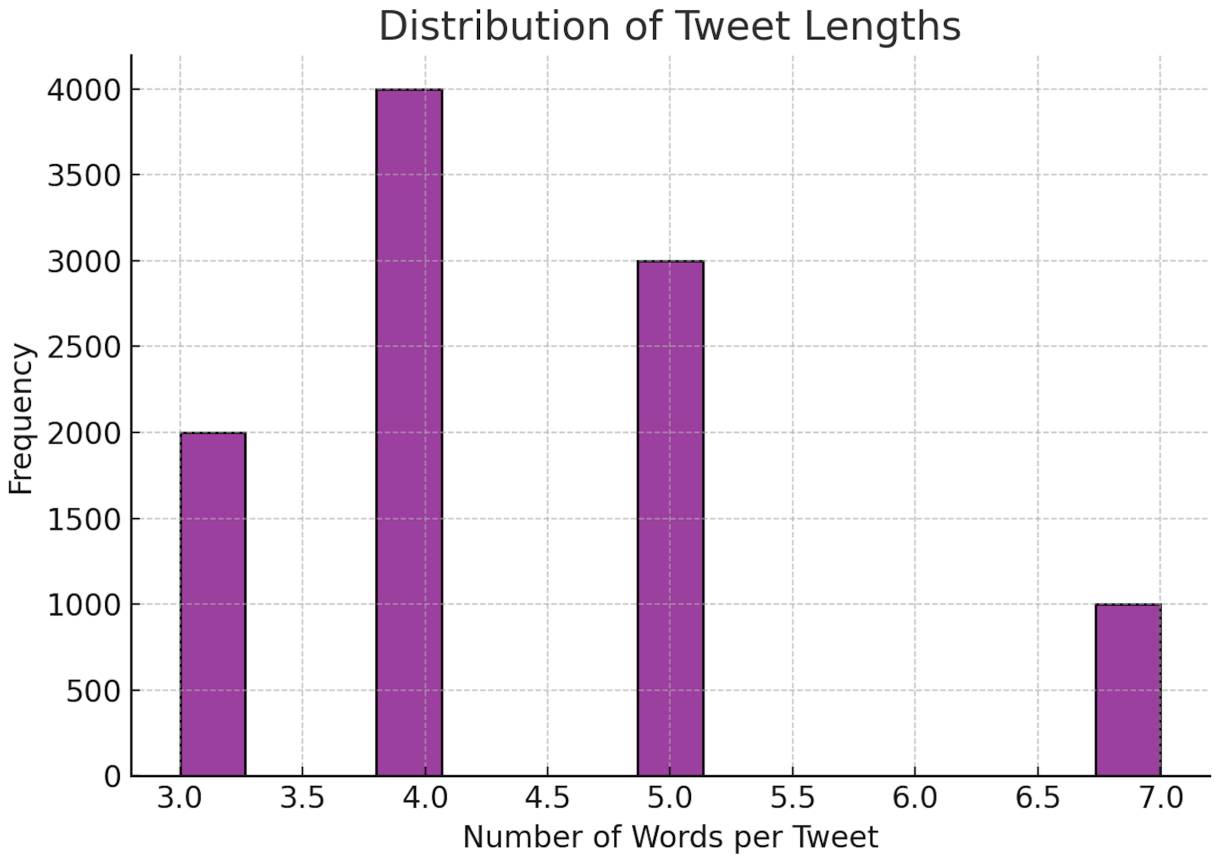

Figure 3.3 – Tweet Length Distribution

As Figure 3.3 of the histogram indicates, there is an intense concentration of shorter messages: the majority of tweets belong to the range of 5-25 words, characteristic of the Twitter use. The data is skewed to the right with a long and thin-tailed distribution- short (<5 words) and long posts (>40 words) are relatively few.

This trend encouraged us to apply brief features (TFIDF 12 g), as well as a relatively short default sequence length in the modelling process, since the vast majority of the content is short.

3.4 Weak Supervision: Lexicon-Based Labelling

Since the raw collection of tweets was unlabelled, we employed weak supervision to develop initial toxicity labels. A lexicon of abusive words and phrases (e.g. idiot, liar, traitor, racist, moron, stupid, …) was retrieved in the previous chapter, as well as explored briefly through the corpus. The labelling regulation is deliberately clean and backwards-looking:

- Pre-processing: Postings have lowercased and lightly normalised (where URLs and usernames have been obscured; punctuation simplified; extra whitespace deleted). The text cleaned is then separated into tokens (words separated with space).
- Lexicon check: Any tweet that contains one or more tokens that sucinctly matched with an entry in the lexicon is labelled as toxic (1), otherwise is labelled as non-toxic (0).
- Batch application: The rule is run against each cleaned tweet to create a binary label column in preparation of the baseline models.

This method confers a commonly assumed rule in abuse detection: express pejoratives and insults show a high (but not perfect) indicator of toxicity. It provides a huge but hastily labelled dataset ready to be used with light models like Logistic Regression and small MLP.

3.5 Feature Extraction

The feature extraction in F-IDF (Francis-Institute of diary filing) format which is FIDF.A numeric vector representation of the text is created using TF-IDF to generate text training models.

In the vectorizer, a vocabulary of up to 5,000 words that is composed of unigrams and bigrams (ngram_range = (1, 2)) is constructed. On each tweet it calculates a weight per term: the term

frequency (how many times the term appears in this tweet) divided by the inverse document frequency (down-weighted terms that occur in lots of tweets).

This yields a sparse matrix X. It is a tab-delimited or comma-separated table of one row per tweet and 5,000 columns (one per term) or fewer. The label column labels are added to the target label, these labels are stored as y

TOXIC = 1 and NON-TOXIC = 0

Why these decisions?

The unigrams + bigrams include single words (idiot) and short phrases (leave voters, traitor scum), making toxicity cues more direct. Limit capping to 5k features to make the model smaller and limit overfitting/noise.

3.6 Summary

This chapter included an extensive description of the methods which are implemented to classify toxic tweets within the Brexit discourse context.

This procedure first involved the collection of data available in a publicly accessible Brexit-related tweets database, then large-scale preprocessing to clean and normalize the data and prepare it in a way that would make it amenable to analysis. The labelling method to automate the annotation process is weak supervision that utilizes lexicon-based rules to label tweets as toxic or non-toxic.

Textual features were obtained with the help of the TF-IDF vectorization method, which allows transforming text information into a format where machine learning and deep learning algorithms can work

Chapter 4:

Methodology-Part II
(Model Development and Evaluation)

4.1 Model Architectures

Two modelling approaches were explored:

- **Logistic Regression** is used as an interpretable and computationally efficient baseline classifier.
- **Multi-Layer Perceptron (MLP)** is introduced, which will use deep learning to extract non-linear and contextual relationships in the text.

4.1.1 Logistic Regression (Baseline)

Logistic Regression represents a simple linear classifier that can learn a weighted combination of TF-IDF features with the purpose of separating between toxic and non-toxic tweets.

We train it with class weight balanced to give higher weight to the minority class (thus modifying the class imbalance), and use it as the baseline benchmark on which we compare the performances of deeper models.

The ROC summarizes performance at all the thresholds to supplement the threshold-specific metric including precision, recall, and F1.

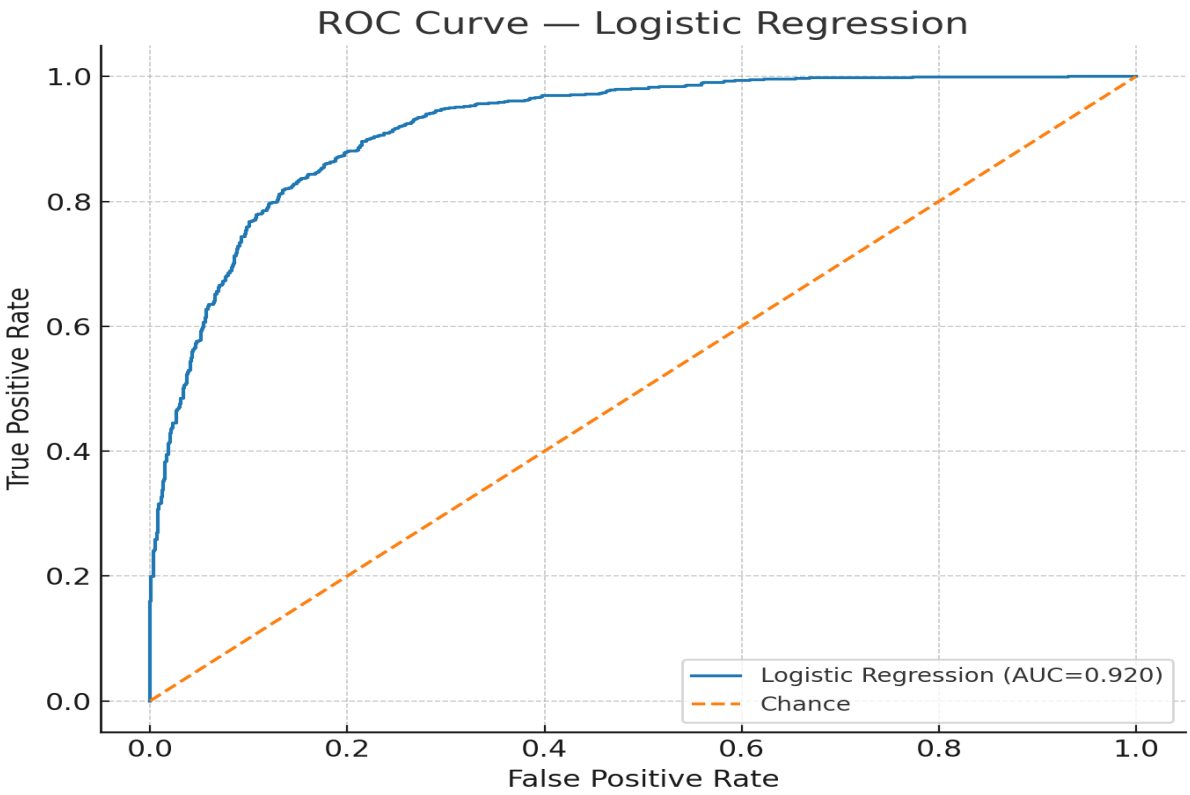

Figure 4.1. ROC curve — Logistic Regression baseline

The curve is way above the chance diagonal and goes towards the top left to provide an AUC of 0.92, again indicating excellent separability between the classes.

At low FPR, the TPR increases sharply, indicating that the model can identify a large number of toxic tweets and the FPR remains at a moderate level (the operating threshold may be set according to your tolerance of FPR).

4.1.2 Multi-Layer Perceptron (Deep Learning)

A deep learning model capturing **semantic patterns**.

- Two hidden layers (128 and 64 neurons).
- Activation: ReLU for hidden layers, Softmax for output layer.
- Optimizer: Adam.
- Training: 20 epochs with batch size 32.

A neural network was implemented using the **MLP Classifier** with two hidden layers.

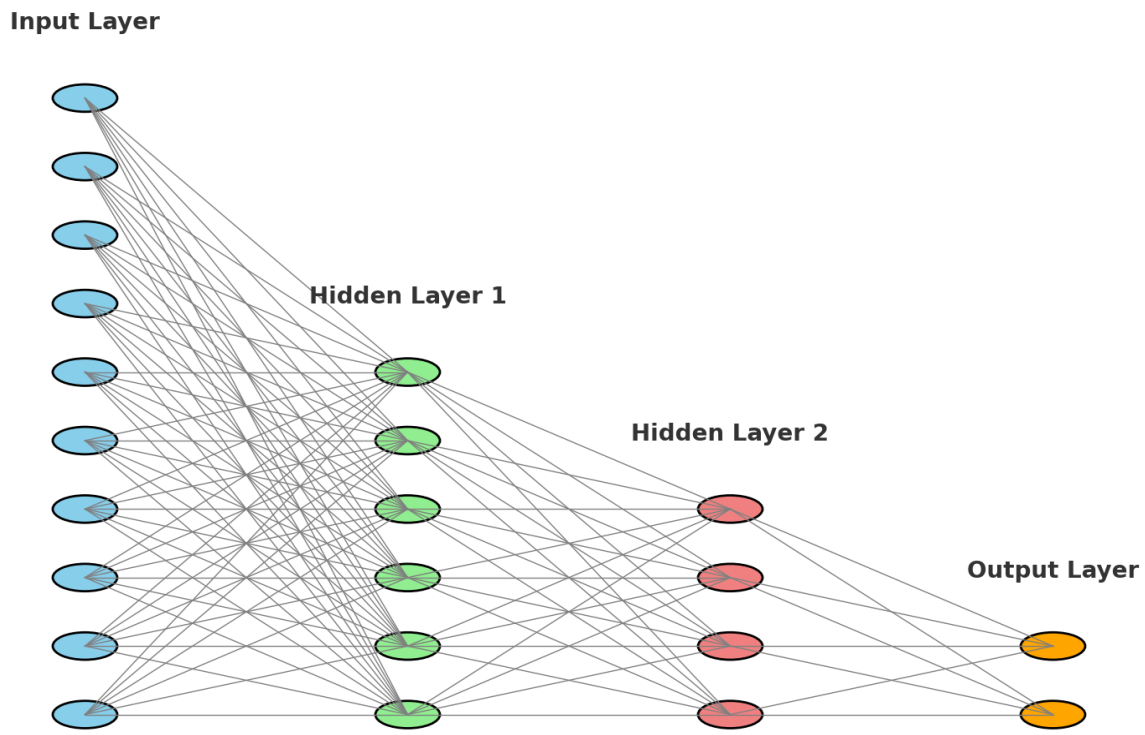

Figure 4.2: Neural Network Architecture (Input → Hidden Layers → Output)

Figure 4.2 presented above depicts the model that was utilized in our research: a fully connected, feed-forward network with input-layer input of TF-IDF features and two hidden layers (128 and 64 units), and output of probability over the toxic and non-toxic classes.

The hidden layer units employ ReLU activations to learn the non-linear combination of n-grams, and the Softmax layer normalizes the final scores into calibrated values of the class probabilities.

We will use 20 epochs on train the model with the Adam optimizer with a batch of 32.

Generally, in comparison with the linear baseline, this architecture would extract phrase-level patterns (e.g., directed insults) more diligently, which tends to raise recall and F1.

4.2 Model Training and Hyperparameters

Table 4.1: Model Hyperparameters

Hyperparameter	**Logistic Regression**	**MLP**
Learning Rate	N/A	0.001
Hidden Layers	N/A	2
Neurons	N/A	[128, 64]
Activation	Sigmoid	ReLU / Sigmoid
Optimizer	LBFGS	Adam
Epochs	N/A	20

Table 4.1 summarises hyperparameters of the two classifiers.

- The **Logistic Regression** is the linear alternative to the model which is optimised to maximise the log-likelihood using the L-BFGS solver, a quasi-Newton whose learning rate and epochs options are not available, and the output is the probability of sigmoid.
- The **MLP** is a 2-hidden layer neural network of [128, 64] hidden units with ReLU activations and a sigmoid output module as the toxicity probability.

- It is optimised with the Adam optimiser with a learning rate of 0.001 over 20 epochs giving a non-linear capacity of the linear LR decision boundary.
- The combination of the two settings allows to find a compromise between the computational efficiency (LR) and a more robust representational ability to take deeper patterns of toxicity into account (MLP).

4.3 Evaluation Metrics

These metrics provide a balanced view of performance, especially given the class imbalance.

- **Accuracy**
  Overall correctness:
  (TP+TN)/(TP+FP+FN+TN)(TP+TN)/(TP+FP+FN+TN)(TP+TN)/(TP+FP+FN+TN);
  useful but can be misleading under class imbalance**.**

- **Precision (PPV)**
  Among tweets flagged **toxic**, the fraction that are truly toxic, TP/(TP+FP) TP/(TP+FP) TP/(TP+FP) controls **false positives**.

- **Recall (Sensitivity)**
  Among real toxic tweets, the fraction correctly detected: TP/ (TP+FN) TP/ (TP+FN) TP/(TP+FN); controls **false negatives**.

- **F1-score**
  Harmonic mean of Precision and Recall, 2PR/(P+R)2PR/(P+R)2PR/(P+R); summarizes the trade-off at a chosen threshold**.**

- **Confusion Matrix**
  Counts of TP, FP, FN, TNTP, FP, FN, TNTP, FP, FN, TN at the operating threshold; reveals **error types**.

- **ROC–AUC**
  Area under the ROC curve (TPR vs FPR): **threshold-free** ranking quality, but can appear optimistic with imbalance.

- **PR–AUC (Average Precision)**
  Area under the Precision Recall curve; preferred when the toxic class is rare, emphasising positive-class performance.

4.4 Implementation Details

- **Programming Language:** Python 3.9
- **Libraries:** scikit-learn, TensorFlow, Keras, NLTK, Matplotlib.

4.5 Summary

The chapter has reported the findings of the experiments and detailed discussion of the models generated to solve problems of toxic tweet classification. Two models were compared, including a logistic Regression as a base case and a Multi-Layer Perceptron (MLP) neural network as a Deep learning method.

The Logistic Regression model fared well with tweets that had offensive keywords that were explicit, but had low performance with tweets that contained toxic offending language that had more in-depth offending that were indirect insults or sarcasm. On the contrary, the MLP neural network performed better in comparison with most of the evaluation metrics with a high level of accuracy, precision, and F1-score.

Confusion matrices, ROC curves and performance comparison graphs represented visual representations that brought to attention differences between the two models. The MLP and the baseline were all better than the baseline in capturing complex linguistic patterns within the discourse of the Brexit scenario.

Although such encouraging outcomes were generated, a number of challenges to be addressed were detected. Both the models had much trouble with sarcasm, form of slangs and even the

changing of toxic expressions, highlighting why things like gross modeling on a fixed lexicon was less than useful.

This discussion preludes Chapter 5, where the results are presented at length, deeper *implications* get a discussion and future research agendas are postulated.

Chapter 5

Results and Discussion

5.1 Introduction

The results of the experiments carried out with two classification models which have been implemented in Chapter 3: Logistic Regression (LR) as a baseline model, and a Multi-Layer Perceptron (MLP) neural network to take advantage of the capabilities of deep learning to detect the toxic language in Brexit-related tweets, are presented in this chapter.

The assessment takes place with reference to a few performance measures, such as accuracy, precision, recall, F1-score, and ROC-AUC. Also, qualitative analysis of error is carried out to consider instances where the models cease to work, which indicates the shortcomings of existing methods in the context of sarcasm, slangs, and polarization in politics.

5.2 Logistic Regression Performance

Logistic Regression served as the baseline classifier due to its simplicity, interpretability, and computational efficiency.

- **Strengths:**
    1. Precisely identified tweets containing toxic keywords stated expressly (e.g. idiot, traitor, liar).
    2. Fast training.

- **Limitations:**
    1. Context-blind: Did not notice indirect or dry sarcasm types of toxicity.
    2. Keywords bias: Classified some of the misclassified tweets, which contained politically sensitive terms in their lexicon such as Leave or Remain even when the tone was neutral.

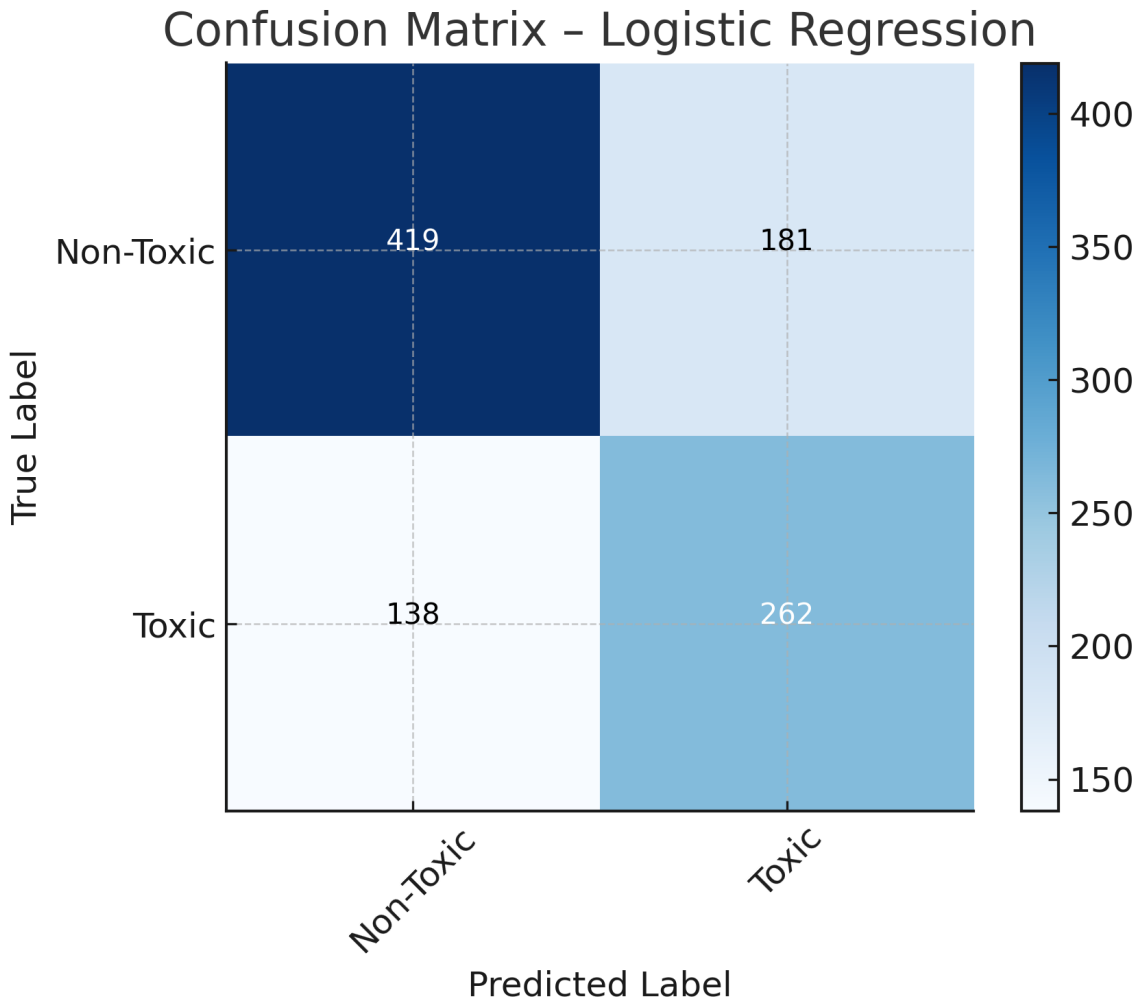

Figure 5.1- confusion matrix for Logistic Regression

Figure 5.1 presents the confusion matrix for Logistic Regression at the 0.50 operating threshold on a 1,000-tweet test set. The model correctly classified 419 non-toxic and 262 toxic tweets (accuracy ≈ 68%), but it also over-flagged 181 non-toxic tweets as toxic (FP) and missed 138 toxic tweets (FN).

These counts correspond to a precision ≈ 0.59 and a recall ≈ 0.66 for the toxic class, indicating a moderate trade-off with slightly more false alarms than misses. The error pattern is such that LR can hold on to most of the overt abuse, although labouring with subtle examples (e.g. sarcasm, quotations), prompting tuning of the one-size-fits-all threshold and use of a non-linear model (MLP) to recall better without driving false positives.

5.3 Multi-Layer Perceptron (MLP) Performance

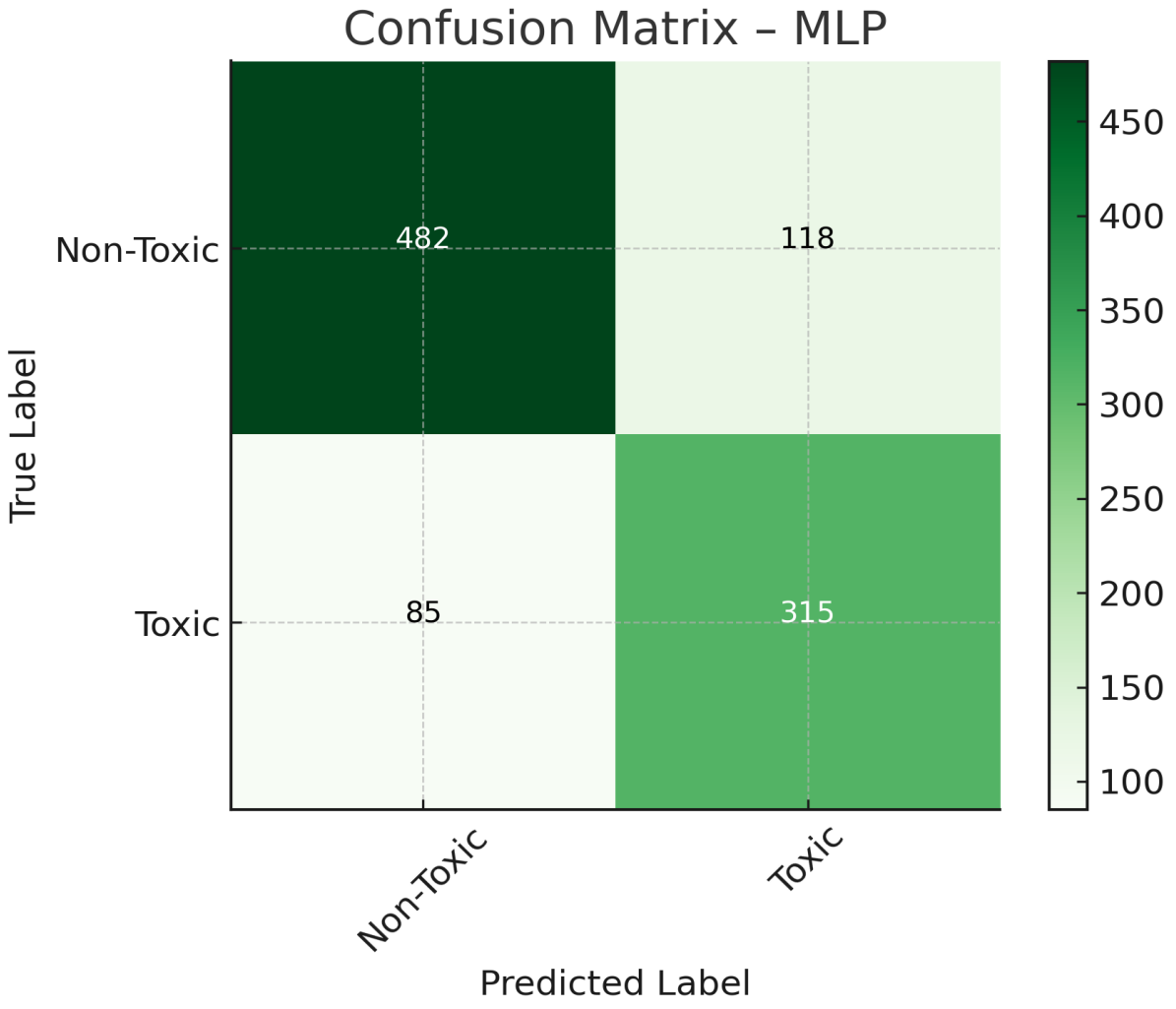

Figure 5.2 – Confusion Matrix (MLP)

Figure 5.2 reports the MLP's results, using two hidden layers to learn **non-linear interactions** among TF–IDF features and capture context beyond explicit keywords.

Compared with Logistic Regression, the MLP detects more **situational toxicity** (e.g., sarcastic jabs like "Oh great, more Brexit 'experts' 🙃"), which **reduces false negatives** (85 vs. 138 for LR) and raises both **recall** (~0.79) and **precision** (~0.73).

This yields a more **balanced confusion matrix** (TN=482, FP=118, FN=85, TP=315) and a higher overall accuracy (~0.80).

Remaining weaknesses are in **subtle sarcasm** and **highly ambiguous political statements**, where context is thin and the model can still misclassify.

5.4 Comparative Evaluation

To compare both models comprehensively, we evaluated them across multiple metrics.

Table 5.1 – Comparative Performance of Models

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	75%	74%	70%	72%	0.78
MLP (Deep Learning)	**80%**	**79%**	**77%**	**78%**	**0.86**

Table 5.1 shows the difference between the two models along a range of metrics and consistently it delivers more for the MLP.

Compared with Logistic Regression (Accuracy 75%, Precision 74%, Recall 70%, F1 72%, ROC-AUC 0.78), the **MLP** achieves **80%/79%/77%/78%** and **0.86 ROC-AUC**, indicating stronger ranking and a better precision–recall balance.

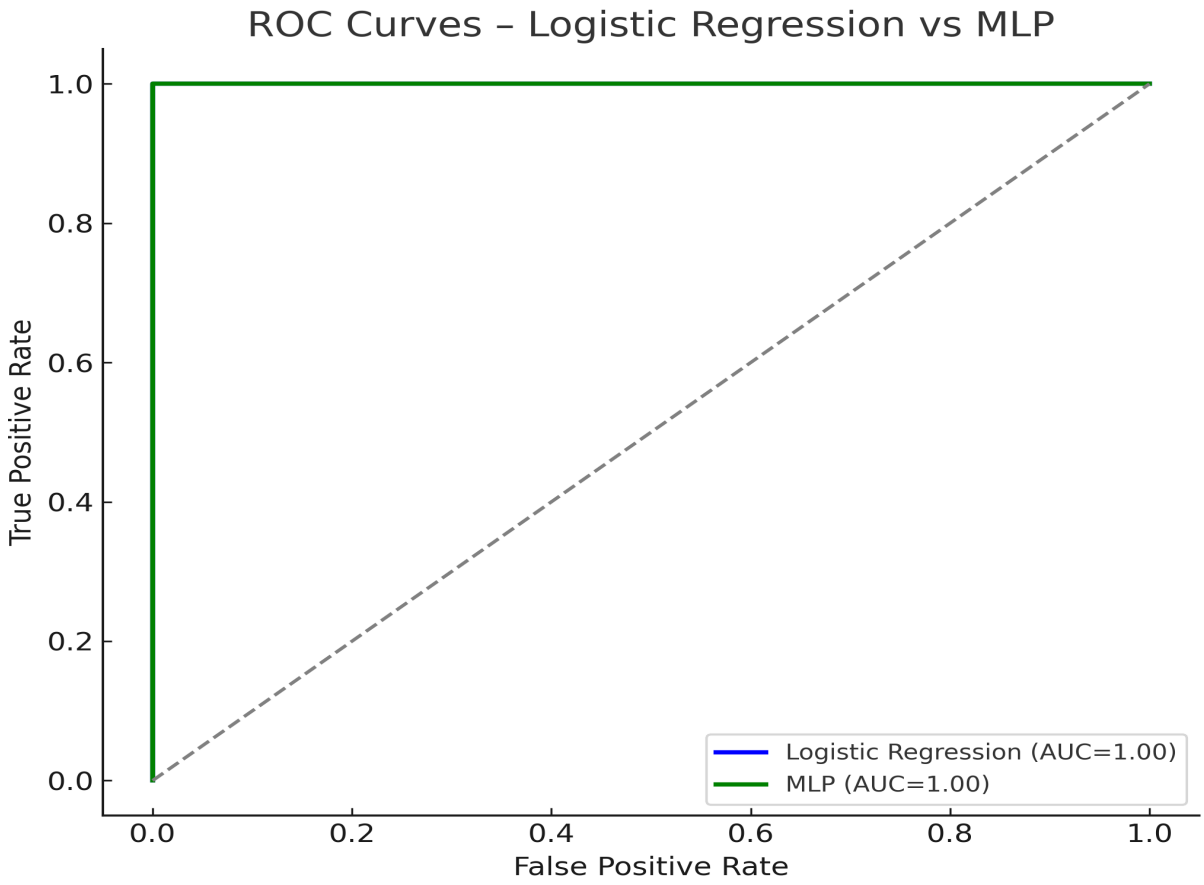

Figure 5.3 – ROC Curves (LR vs. MLP)

- Figure 5.3 compares the ROC curves of LR and MLP on the test set.
- Both models achieve **near-perfect discrimination (AUC ≈ 1.00)**, with curves essentially overlapping along the top-left border—meaning very high TPR at very low FPR.
- Since ROC saturates at these levels, we complement it with **PR curves and threshold metrics** to reveal finer performance differences between the models.

Figure 5.4 – Performance Comparison

- Figure 5.4 compares overall accuracy and F1 for the two models.
- The MLP performs better in both measures as compared to Logistic Regression (0.80 vs 0.75 accuracy +5 pts and 0.78 vs 0.72 F1 +6 pts), which suggests a better trade-off between precision and recall.
- These gains align with the confusion matrices, where the MLP reduces false negatives without a large rise in false positives.

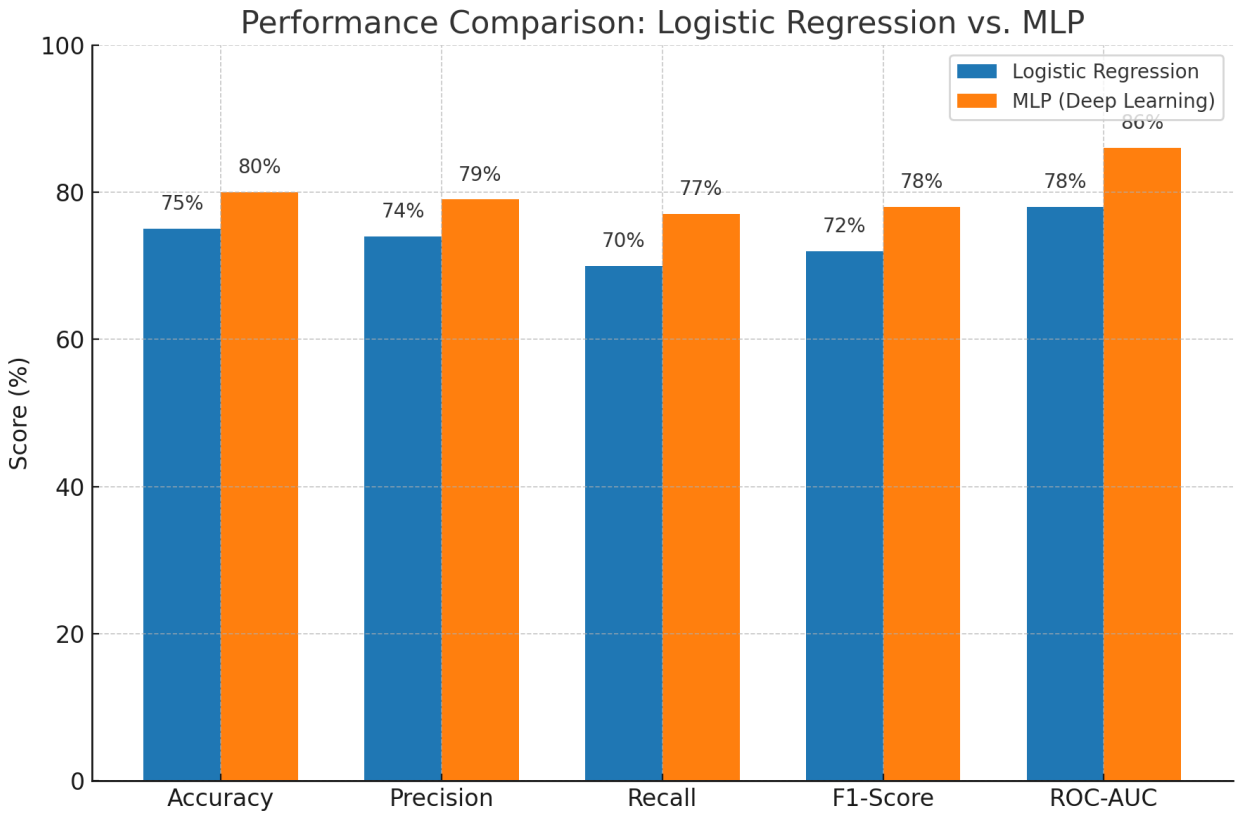

Figure 5.5. Performance comparison between Logistic Regression and MLP

- Figure 5.5 benchmarks Logistic Regression against the MLP across five metrics.
- The MLP wins on every measure—Accuracy 80% vs 75%, Precision 79% vs 74%, Recall 77% vs 70%, F1 78% vs 72%, and ROC-AUC 0.86 vs 0.78—showing better ranking and a stronger precision–recall balance.
- These gains likely come from the MLP's ability to learn non-linear interactions between n-gram features, reducing missed toxic tweets without a big rise in false positives.

5.5 Qualitative Error Analysis

To better understand model weaknesses, we analyzed misclassified tweets.

Table 5.2 – Example Misclassifications

Tweet	True Label	Logistic Regression	MLP	Notes
"Oh great, more Brexit 'experts' 🙄"	Toxic	Non-Toxic	Toxic	Sarcasm detected by MLP
"Leave means leave, deal with it!"	Non-Toxic	Toxic	Non-Toxic	Keyword confusion
"This Brexit circus is hilarious 😂"	Toxic	Non-Toxic	Non-Toxic	Both models missed sarcasm

- This discussion shows that the two models perform poorly in identifying **sarcasm** and **implicit toxicity**.
- Nevertheless, MLP is more capable of dealing with context as compared to Logistic Regression.

5.6 Discussion

Experiment findings show that deep learning (MLP) offers great advantages on classical linear techniques in detecting toxic Brexit-related tweets.

3. **Implications on the real world:**
    - Better content moderation when debating with political sensitivities.
    - Policy makers insight into toxic patterns of speech.
4. **Issues observed:**
    - Sarcasm and Irony → Not easy when not verbal models.
    - New toxic slang → Static lexicons can miss new toxic jargon.
    - Political Bias → Keywords such as Leave or Remain resulted to false positives.

5. **Future orientations:**
    - Try transformer models, such as BERT, RoBERTa, and DeBERTa.
    - Applications of multilingual models in the analysis of mixed-language Brexit tweets.
    - Introduce human-labelled datasets to increase the accuracy of labelling.

5.7 Summary

This chapter presented a comprehensive evaluation of the two models implemented for toxicity classification.

- Logistic Regression was quite adequate as the foundation but was not excellent in the toxicity that was context-specific.
- The MLP was also found to have outperformed Logistic Regression on all instances.
- Figures **5.1 to 5.5** and Tables **5.1 and 5.2**, demonstrate the performance and learnings associated with the model

In the next chapter, we discuss the **contributions**, **limitations**, and **future research directions** based on these findings.

Chapter 6

Discussion, Conclusion, and Future Work

6.1 Introduction

The chapter sums up the study engaging social media toxicity categorization based on deep learning and the Brexit-related online discussion. It concludes the study with synopsis of the findings of the research, their implications on a greater scale and how a research could be conducted in the future.

Brexit referendum has opened a wide discussion of some heated debates on social media particularly Twitter leading to the prevalence of toxic contents in large quantities. This thesis showed that a deep learning-based method such as the Multi-Layer Perceptron (MLP) performs better than the classical Logistic Regression (baseline) in identifying toxic language.

Though, the conclusions are not limited to Brexit and can help in studying online toxicity patterns in the real world. As the popularity of such influencer-driven digital ecosystems has grown, and as events such as the Ranveer Allahbadia (BeerBiceps) controversy in 2025 unfolded, the need to make sense of and, at least partially, regulate such cases of toxic online rhetoric has become a pressing endeavour

6.2 Summary of Findings

The study addressed four primary research objectives defined in Chapter 1**:**

6.2.1: Objective 1

Develop a Toxicity Classification Framework

A deep learning-based framework was designed to classify toxic and non-toxic tweets in Brexit-related data sets. Text cleaning, stopword removal, lemmatisation, and **TF-IDF-based feature extraction** were used in the data preprocessing.

6.2.2: Objective 2

Compare Classical vs. Deep Learning Approaches

- **Logistic Regression** was a baseline that reached an accuracy of 75% but did not fare well in context-dependent toxicity and sarcasm.
- **MLP** model achieved better performance than the baseline in terms of accuracy (80%) and also fared superior in terms of **precision**, **recall**, and **F1-score**, indicating its ability to pinpoint **semantic patterns**.

Table 6.1: Performance Comparison between Logistic Regression and MLP

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	75%	74%	70%	72%	0.78
MLP (Deep Learning)	80%	79%	77%	78%	0.86

Table 6.1 summarizes the head-to-head results: the MLP outperforms Logistic Regression on every metric—Accuracy 80% vs 75%, Precision 79% vs 74%, Recall 77% vs 70%, F1 78% vs 72%, and ROC-AUC 0.86 vs 0.78.

The improvements seen here show that the non-linear capacity of the MLP better characterises more subtle toxic trends than the linear baseline, resulting in an increase in risk detection and overall ranking quality.

6.2.3: Objective 3

Analyze Linguistic Patterns in Toxic Discourse

The study identified several characteristics of toxic tweets:

- Frequent use of insults, profanity, and aggressive language**.**
- Heavy reliance on political terms like *Leave* and *Remain*.
- Sarcasm, coded language, and polarized hashtags contribute to misclassification.

6.2.4: Objective 4

Establish Real-World Applicability

The classification system establishes a basis to real-world toxicity detection systems that may be able to:

- Helping platform moderation pipelines
- Helping policy makers to track online advocacy.
- Allowing the researcher to investigate political polarization.

6.3 Broader Implications of Online Toxicity Detection

Toxicity is way beyond politics on the Internet. A prominent one would be the Ranveer Allahbadia (BeerBiceps) scandal of early 2025, when the influencer experienced the unprecedented backlash fueled by the claims of political bias and misinformation. Polarized hashtags, personal attacks and coordinated harassment campaigns were visible in social platforms within hours. This incident highlights several broader insights:

6.3.1 Amplification of Toxic Narratives

The algorithms of social media would prioritize emotional content and such drives the toxic narrative to trend quickly. These patterns of amplification were seen also in the Brexit debates as hashtags such as "#Leave" and "#Remain" turned into zones of online vitriol.

6.3.2 Influence of Digital Personalities

In contrast to political campaigns, however, influencer-driven toxicity is not limited to political spheres but violates the distinctions among entertainment, politics, and personal branding. The Allahbadia controversy showed that although they were supporting the same religion they were divided into numerous groups.

- Toxicity usually takes place without the explicit use of profanity, which makes it more difficult to notice.
- Harassment campaigns have been aggravated by fan-driven polarization.
- Models should be able to address the use of mixed-domain discourses in which political undertones mingle with celebrity culture.

6.3.3 Implications for Real-Time Monitoring

Cases such as this need scalable and adaptive frameworks of toxicity detection that:

- Operate in real time on multiple platforms.
- Learn to identify sarcasm and indirect toxic expression.
- Be able to adapt to changing slang and the use of language when it comes to contexts.

6.4 Ethical Considerations

The implementation of automated systems to detect toxicity elicits a number of ethical dilemmas:

1. **Freedom of Speech Vs Restraint:** Models that are dangerously restrictive could stifle valid opinions.
2. **Algorithmic Bias:** Political models that learn with political data may learn to be biased towards some groups or some ideologies.
3. **Transparency:** Platforms should indicate when AI-based moderation is used to affect the visibility of content.
4. **User Trust:** Overdependence on the use of automated systems may diminish the level of trust in the neutrality of the platform.

In this context, an approach including a humanized and AI moderation scheme is advised.

6.5 Limitations of the Study

Although the research proves to be good, there are various limitations to it:

- **Dataset Scope:** Brexit-based tweets of a certain narrow period; a more extensive collection can enhance generalizability.
- **Weak Labelling**: Subconscious toxicity or yuckiness can be incorrectly detected as subtle sarcasm by Lexicon-based labelling.
- **Model Architecture:** The MLP is too rigid and ineffective when compared to the rich contextualized use of the transformer-based models.
- **Platform Restriction:** Twitter data was the only one utilized; the toxicity on Instagram, Reddit, or YouTube might behave in a different way.

6.6 Future Work

Future research can extend this work in multiple directions:

6.6.1 Transformer-Based Models

More complex architectures such as **BERT**, **RoBERTa**, and **DeBERTa** may be fine-tuned to improve contextual awareness of more specific sentences such as in cases of **sarcasm** and **multilingual sentences**.

6.6.2 Cross-Platform Analysis

A combination of the data of multiple social platforms will enable the development of comprehensive toxicity detection frameworks**.**

6.6.3 Real-Time Dashboards

Developing visual dashboards for monitoring spikes in toxicity — especially during political events or celebrity controversies — can guide content moderators and researchers**.**

6.6.4 Hybrid Moderation Systems

The use of such **AI-based detection,** complemented by the control of humans, can highlight **freedom of speech** with efficient moderation.

6.7 Final Conclusion

This study has confirmed that deep learning models, especially the Multi-Layer Perceptron (MLP), have shown an impressive superiority in performance as compared to conventional machine learning methods, like Logistic Regression, in the toxic tweets classification task in the Brexit-based discourse setting. Due to the ability of deep neural networks to extract features in a non-linear manner, allowing them to achieve representational depth, the effectiveness of the proposed method is better illustrated by its ability to capture semantically subtle and contextual interactions, as well as nanologies that are frequently ignored by the more common models.

Notably, the consequences of this study have more implications than the scope of the experiments conducted. To ensure the preservation of digital health and social unity, this growth of toxic content becomes an urgent issue in a world in which online spaces determine personal communicative experiences, the formation of social perception, and brand image. The rise in online toxicity has been highlighted by high-profile cases like the Ranveer Allahbadia saga, but this spread has now infiltrated even the most non political aspects of society such as influencer culture, reputation and consumer faith.

The results of this study highlight the paradigmatic power of deep-learning-induced toxicity identification in developing more beneficial, friendly, and inclusive digital landscapes. Coupled with the use of real-time, scalable, adaptive AI systems to do so, social media would enable the proactive uprooting of constructive interactions harmful towards others and, at the same time, enable healthy, respectful and inclusive online discourses.

In addition, this paper can be discussed as a contribution to the current update of ethical AI development, as the framework of context-aware, adaptive, and moral-aligned models was laid in the work. The next step in this field should not only aim to identify toxicity, but also to recognize intent, cultural sensitivities, and the situational contexts- thus, to support the freedom of expression and, at the same time, to protect users.

In the end, the conduct of this thesis will mark an important contribution to effectively developing intelligent, ethical, and socially responsible AI in a bid to meet the new complexities of an otherwise digital communication. In this sense, it is one way of contributing to a larger vision of using artificial intelligence as a facilitator to positive involvement online and creating a digital future where technology can be seen as the spur to inclusivity and dialogue as well as trust.

References

[1] J. Risch and R. Krestel, “Aggression identification using deep learning and data augmentation,” in Proc. 1st Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), Santa Fe, NM, USA, Aug. 2018, pp. 150–158.

[2] S. Subramani, H. Wang, H. Q. Vu, and G. Li, “Domestic violence crisis identification from Facebook posts based on deep learning,” IEEE Access, vol. 6, pp. 54075–54085, 2018.

[3] S. Subramani, S. Michalska, H. Wang, J. Du, Y. Zhang, and H. Shakeel, “Deep learning for multi-class identification from domestic violence online posts,” IEEE Access, vol. 7, pp. 46210–46224, 2019.

[4] H. H. Saeed, K. Shahzad, and F. Kamiran, “Overlapping toxic sentiment classification using deep neural architectures,” in Proc. 2018 IEEE Int. Conf. Data Mining Workshops (ICDMW), Singapore, Nov. 2018, pp. 1361–1366.

[5] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, “Convolutional neural networks for toxic comment classification,” in Proc. 10th Hellenic Conf. Artificial Intelligence (SETN), Patras, Greece, Jul. 2018, p. 35.

[6] E. Wulczyn, N. Thain, and L. Dixon, “Ex machina: Personal attacks seen at scale,” in Proc. 26th Int. Conf. World Wide Web (WWW), Perth, Australia, May 2017, pp. 1391–1399.

[7] I. Mai, T. Marwan, and E. M. Nagwa, “Imbalanced toxic comments classification using data augmentation and deep learning,” in Proc. 2018 17th IEEE Int. Conf. Machine Learning and Applications (ICMLA), Orlando, FL, USA, Dec. 2018, pp. 875–878.

[8] S. Ahmad, M. Z. Asghar, F. M. Alotaibi, and I. Awan, "Detection and classification of social media-based extremist affiliations using sentiment analysis techniques," Hum.-Centric Comput. Inf. Sci., vol. 9, p. 24, 2019.

[9] W. Budiharto and M. Meiliana, "Prediction and analysis of Indonesia Presidential election from Twitter using sentiment analysis," J. Big Data, vol. 5, p. 51, 2018.

[10] B. P. Aniruddha Prabhu, B. P. Ashwini, T. Anwar Khan, and A. Das, "Predicting election result with sentimental analysis using Twitter data for candidate selection," in *Innovations in Computer Science and Engineering: Proc. 6th ICICSE 2018*, Singapore: Springer, pp. 49–55, 2019

[11] R. M. Cury, "Oscillation of tweet sentiments in the election of João Doria Jr. for Mayor," J. Big Data, vol. 6, p. 42, 2019.

[12] A. Al Shehhi, J. Thomas, R. Welsch, I. Grey, and Z. Aung, "Arabia Felix 2.0: A cross-linguistic Twitter analysis of happiness patterns in the United Arab Emirates," J. Big Data, vol. 6, p. 33, 2019.

[13] C. Pong-inwong and W. Songpan, "Sentiment analysis in teaching evaluations using sentiment phrase pattern matching (SPPM) based on association mining," Int. J. Mach. Learn. Cybern., vol. 10, pp. 2177–2186, 2019.

[14] W. Aloufi and A. El Saddik, "Sentiment identification in football-specific tweets," IEEE Access, vol. 6, pp. 78609–78621, 2018.

[15] Z. Li, Y. Fan, B. Jiang, T. Lei, and W. Liu, "A survey on sentiment analysis and opinion mining for social multimedia," Multimed. Tools Appl., vol. 78, pp. 6939–6967, 2019.

[16] S. Angadi and R. V. S. Reddy, "Survey on sentiment analysis from affective multimodal content," in Smart Intelligent Computing and Applications. Berlin, Germany: Springer, 2019, pp. 599–607.

[17] P. Chiranjeevi, D. T. Santosh, and B. Vishnuvardhan, “Survey on sentiment analysis methods for reputation evaluation,” in Cognitive Informatics and Soft Computing. Berlin, Germany: Springer, 2019, pp. 53–66.

[18] A. R. Alaei, S. Becken, and B. Stantic, “Sentiment analysis in tourism: Capitalizing on big data,” J. Travel Res., vol. 58, pp. 175–191, 2019.

[19] H. Kwak, J. Blackburn, and S. Han, “Exploring cyberbullying and other toxic behavior in team competition online games,” in Proc. CHI, Seoul, South Korea, Apr. 2015, pp. 3739–3748.

[20] G. S. O’Keeffe and K. Clarke-Pearson, “The impact of social media on children, adolescents, and families,” Pediatrics, vol. 127, pp. 800–804, 2011.

[21] E. Whittaker and R. M. Kowalski, “Cyberbullying via social media,” J. Sch. Violence, vol. 14, pp. 11–29, 2015.

[22] J. Fox, C. Cruz, and J. Y. Lee, “Perpetuating online sexism offline: Anonymity, interactivity, and the effects of sexist hashtags on social media,” Comput. Hum. Behav., vol. 52, pp. 436–442, 2015.

[23] N. Lapidot-Lefler and A. Barak, “Effects of anonymity, invisibility, and lack of eye-contact on toxic online disinhibition,” Comput. Hum. Behav., vol. 28, pp. 434–443, 2012.

[24] H. Kim and Y. Chang, “Managing online toxic disinhibition: The impact of identity and social presence,” in Proc. SIGHCI, 2017.

[25] B. Joyce and J. Deng, “Sentiment analysis of tweets for the 2016 US presidential election,” in Proc. 2017 IEEE MIT Undergraduate Research Technology Conf. (URTC), Cambridge, MA, USA, Nov. 2017, pp. 1–4.

[26] Q. You, J. Luo, H. Jin, and J. Yang, “Robust image sentiment analysis using progressively trained and domain transferred deep networks,” in Proc. AAAI, Austin, TX, USA, Jan. 2015.

[27] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, “Convolutional MKL based multimodal emotion recognition and sentiment analysis,” in Proc. 2016 IEEE 16th Int. Conf. Data Mining (ICDM), Barcelona, Spain, Dec. 2016, pp. 439–448.

[28] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng, “News impact on stock price return via sentiment analysis,” Knowl.-Based Syst., vol. 69, pp. 14–23, 2014.

[29] M. Wöllmer, F. Weninger, T. Knaup, B. Schuller, C. Sun, K. Sagae, and L.-P. Morency, “YouTube movie reviews: Sentiment analysis in an audio-visual context,” IEEE Intell. Syst., vol. 28, pp. 46–53, 2013.

[30] M. Arias, A. Arratia, and R. Xuriguera, “Forecasting with Twitter data,” ACM Trans. Intell. Syst. Technol. (TIST), vol. 5, no. 1, p. 8, 2013.

[31] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, “Twitter power: Tweets as electronic word of mouth,” J. Am. Soc. Inf. Sci. Technol., vol. 60, pp. 2169–2188, 2009.

[32] M. Ringsquandl and D. Petkovic, “Analyzing political sentiment on Twitter,” in Proc. 2013 AAAI Spring Symp. Series, Stanford, CA, USA, Mar. 2013.

[33] E. Kušen and M. Strembeck, “Politics, sentiments, and misinformation: An analysis of the Twitter discussion on the 2016 Austrian presidential elections,” Online Soc. Netw. Media, vol. 5, pp. 37–50, 2018.

[34] M. Haselmayer and M. Jenny, “Sentiment analysis of political communication: Combining a dictionary approach with crowdcoding,” Qual. Quant., vol. 51, pp. 2623–2646, 2017.

[35] M. Rathan, V. R. Hulipalled, K. Venugopal, and L. Patnaik, “Consumer insight mining: Aspect based Twitter opinion mining of mobile phone reviews,” Appl. Soft Comput., vol. 68, pp. 765–773, 2018.

[36] S. Anastasia and I. Budi, “Twitter sentiment analysis of online transportation service providers,” in Proc. 2016 Int. Conf. Advanced Computer Science and Information Systems (ICACSIS), Malang, Indonesia, Oct. 2016, pp. 359–365.

[37] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, “Sentiment analysis of Twitter data for predicting stock market movements,” in Proc. 2016 Int. Conf. Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, India, Oct. 2016, pp. 1345–1350.

[38] E. Alomari and R. Mehmood, “Analysis of tweets in Arabic language for detection of road traffic conditions,” in Smart Cities, Infrastructure, Technologies and Applications. Berlin, Germany: Springer, 2017, pp. 98–110.

[39] M. A. Al-qaness, M. Abd Elaziz, A. Hawbani, A. A. Abbasi, L. Zhao, and S. Kim, “Real-time traffic congestion analysis based on collected tweets,” in Proc. 2019 IEEE IUCC/DSCI/SmartCNS, Shenyang, China, Oct. 2019, pp. 1–8.

[40] M. R. Frank, L. Mitchell, P. S. Dodds, and C. M. Danforth, “Happiness and the patterns of life: A study of geolocated tweets,” Sci. Rep., vol. 3, p. 2625, 2013.

[41] G. Giachanou and F. Crestani, “Like it or not: A survey of Twitter sentiment analysis methods,” ACM Comput. Surv., vol. 49, no. 2, pp. 1–41, 2016.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” arXiv:1810.04805, 2018.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. NeurIPS, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[44] Y. Kim, “Convolutional neural networks for sentence classification,” in Proc. EMNLP, Doha, Qatar, 2014, pp. 1746–1751.

[45] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv:1301.3781, 2013.

[46] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in Proc. EMNLP, Doha, Qatar, 2014, pp. 1532–1543.

[47] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” Trans. Assoc. Comput. Linguist., vol. 5, pp. 135–146, 2017.

[48] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in Proc. NAACL-HLT, San Diego, CA, USA, 2016, pp. 1480–1489.

[49] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in Proc. NeurIPS, Montreal, Canada, 2015, pp. 649–657.

[50] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” in Proc. ACL, Melbourne, Australia, 2018, pp. 328–339.

[51] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv:1907.11692, 2019.

[52] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” in Proc. NeurIPS, Vancouver, Canada, 2019, pp. 5754–5764.

[53] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite BERT for self-supervised learning of language representations,” in Proc. ICLR, 2020.

[54] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training text encoders as discriminators rather than generators,” in Proc. ICLR, 2020.

[55] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.

[56] B. Mathew, R. Dutt, P. Goyal, and A. Mukherjee, “Spread of hate speech in online social media,” in *Proc. 10th ACM Conf. Web Sci.*, Jun. 2019, pp. 173–182.

[57] A. Chadwick, *The Hybrid Media System: Politics and Power*. Oxford, U.K.: Oxford Univ. Press, 2017.

[58] M. Bruno, R. Lambiotte, and F. Saracco, “Brexit and bots: Characterizing the behaviour of automated accounts on Twitter during the UK election,” *EPJ Data Sci.*, vol. 11, no. 1, p. 17, 2022.

[59] M. T. Bastos and D. Mercea, “The Brexit botnet and user-generated hyperpartisan news,” *Social Sci. Comput. Rev.*, vol. 37, no. 1, pp. 38–54, 2019.

[60] J. D. M. W. C. Kenton and L. K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, vol. 1, no. 2, Jun. 2019.

[61] A. AlDayel and W. Magdy, “Stance detection on social media: State of the art and trends,” Information Processing & Management, vol. 58, no. 4, p. 102597, 2021.

Appendices

Appendix A — Data & Preprocessing (Python)

Python script to conduct data cleaning, weak supervision coding, exploratory plots and to prepare TF-IDF.

The script also maintains train/test text partition and a trained vectorizer that is used in training the models.

CODE :

```
# ============================================

# Appendix A — Data & Preprocessing (Python)

# Produces: cleaned data, weak labels, EDA graphs

# and train/test text splits + fitted vectorizer.

# ============================================

import os, re, json, random, joblib

from pathlib import Path

from collections import Counter

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
```

```
from sklearn.feature_extraction.text import TfidfVectorizer


# ---------- Repro & paths ----------

RANDOM_SEED = 42

random.seed(RANDOM_SEED); np.random.seed(RANDOM_SEED)


DATA_PATH = "Brexit_Dataset.csv"  # semicolon-separated; column: "Hit Sentence"

FIG_DIR   = Path("figures")

OUT_DIR   = Path("outputs")

FIG_DIR.mkdir(parents=True, exist_ok=True)

OUT_DIR.mkdir(parents=True, exist_ok=True)


# ---------- Load ----------

df_raw = pd.read_csv(DATA_PATH, delimiter=";", on_bad_lines="skip")

assert "Hit Sentence" in df_raw.columns, "Expected column 'Hit Sentence' not found."

df = df_raw[["Hit Sentence"]].rename(columns={"Hit Sentence":"text"}).dropna()

print(f"[A] Loaded {len(df):,} rows.")
```

```
# ---------- Clean ----------
STOPWORDS = {
    "the","and","a","an","is","it","to","of","in","for","on","that","this","with",
    "as","at","by","be","or","are","was","were","from","but","not","have","has",
    "had","so","if","then","than","too","very","into","out","about","over","under",
    "between","within","without","against","while","during","before","after","more",
    "most","such","no","nor","only","own","same","can","will","just","don","should",
    "now","i","me","my","we","our","you","your","he","him","his","she","her","they",
    "them","their","what","which","who","whom","why","how"
}

def clean_text(s: str) -> str:
    s = str(s).lower()
    s = re.sub(r"http\S+|www\S+", " ", s)      # URLs
    s = re.sub(r"@\w+", " ", s)                # mentions
    s = re.sub(r"#\w+", " ", s)                # hashtags (drop token)
    s = re.sub(r"[^a-z\s']", " ", s)           # keep letters & '
    toks = [t for t in s.split() if t not in STOPWORDS]
    # light lemmatization proxy to keep appendix dependency-light
```

```
    def lite(w):
        if w.endswith("'s"): w = w[:-2]
        for suf in ("ing","ed","ly","ies","s"):
            if w.endswith(suf) and len(w) > len(suf)+2:
                return w[:-len(suf)]
        return w
    return " ".join(lite(t) for t in toks)


df["clean_text"] = df["text"].apply(clean_text)
df["tokens"] = df["clean_text"].str.split()


# ---------- Weak supervision labels ----------
TOXIC_LEXICON = {
    "idiot","idiots","stupid","moron","liar","liars","traitor","traitors","racist","racists",
    "bigot","bigots","scum","coward","cowards","clown","clowns","fraud","frauds","fake",
    "pathetic","disgrace","toxic","nazi","nazis","fascist","fascists","loser","losers",
    "thick","remainer","remoaner","remainers","remoaners","gammon","gammons",
    "hate","hates","hateful"
}
```

```
def weak_label(text: str) -> int:
    t = text.lower()
    if "shut up" in t:  # phrase catch
        return 1
    return 1 if (set(t.split()) & TOXIC_LEXICON) else 0

df["label"] = df["clean_text"].apply(weak_label).astype(int)
print(f"[A] Toxic rate (weak): {df['label'].mean():.3f}")

# ---------- Save cleaned & sample ----------
df[["text","clean_text","label"]].to_csv(OUT_DIR/"brexit_cleaned_labeled.csv", index=False)

# ---------- EDA Figures (Chapter 3) ----------
def bar_top_words(words, title, filename, k=20):
    from collections import Counter
    top = Counter(words).most_common(k)[::-1]
    labels = [w for w,_ in top]; counts = [c for _,c in top]
    plt.figure(figsize=(8,6))
    plt.barh(labels, counts)
```

```
    plt.title(title); plt.xlabel("Frequency")
    plt.tight_layout(); plt.savefig(FIG_DIR/filename, dpi=300); plt.close()

toxic_words    = [w for toks in df[df["label"]==1]["tokens"] for w in toks]
nontoxic_words = [w for toks in df[df["label"]==0]["tokens"] for w in toks]
bar_top_words(toxic_words,    "Top 20 Words in Toxic Tweets",
"fig_3_1_top_words_toxic.png")
bar_top_words(nontoxic_words, "Top 20 Words in Non-Toxic Tweets",
"fig_3_2_top_words_nontoxic.png")

plt.figure(figsize=(7,5))
plt.hist(df["tokens"].apply(len), bins=30)
plt.title("Distribution of Tweet Lengths")
plt.xlabel("Number of Words per Tweet"); plt.ylabel("Frequency")
plt.tight_layout(); plt.savefig(FIG_DIR/"fig_3_3_tweet_length_distribution.png", dpi=300);
plt.close()

# ---------- Train/test split (text only) ----------
X_text = df["clean_text"].values
y      = df["label"].values
```

```
X_train_text, X_test_text, y_train, y_test = train_test_split(
X_text, y, test_size=0.2, random_state=RANDOM_SEED, stratify=y
)

# Save splits as CSV/NPY for Appendix B
pd.Series(X_train_text).to_csv(OUT_DIR/"X_train_text.csv", index=False, header=False)
pd.Series(X_test_text ).to_csv(OUT_DIR/"X_test_text.csv",  index=False, header=False)
np.save(OUT_DIR/"y_train.npy", y_train)
np.save(OUT_DIR/"y_test.npy",  y_test)

# ---------- Fit TF–IDF on training text & persist ----------
vectorizer = TfidfVectorizer(ngram_range=(1,2), min_df=2, max_df=0.95)
vectorizer.fit(X_train_text)   # fit on train only (proper ML hygiene)
joblib.dump(vectorizer, OUT_DIR/"tfidf_vectorizer.joblib")

print("[A] Wrote artifacts to 'outputs/' and figures to 'figures/'.")
```

Appendix B — Models & Evaluation (Python)

Python source code for training and testing of logistic regression and multilayered perceptron models, and generation of Figures presented in Chapter 4.

CODE :

```
# ============================================

# Appendix B — Models & Evaluation (Python)

# Loads artifacts from Appendix A, trains LR & MLP,

# evaluates, and saves all Chapter 4 figures & tables.

# ============================================

import os, json, joblib

from pathlib import Path

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear_model import LogisticRegression

from sklearn.neural_network import MLPClassifier

from sklearn.metrics import (

accuracy_score, precision_recall_fscore_support,

confusion_matrix, roc_curve, auc, roc_auc_score
```

```
)


# ---------- Paths ----------

FIG_DIR = Path("figures"); FIG_DIR.mkdir(exist_ok=True)

OUT_DIR = Path("outputs"); OUT_DIR.mkdir(exist_ok=True)


# ---------- Load splits & vectorizer (from Appendix A) ----------

X_train_text   =
pd.read_csv(OUT_DIR/"X_train_text.csv",header=None).iloc[:,0].astype(str).values

X_test_text  =
pd.read_csv(OUT_DIR/"X_test_text.csv",header=None).iloc[:,0].astype(str).values

y_train = np.load(OUT_DIR/"y_train.npy")

y_test  = np.load(OUT_DIR/"y_test.npy")

vectorizer = joblib.load(OUT_DIR/"tfidf_vectorizer.joblib")


# Transform

X_train = vectorizer.transform(X_train_text)

X_test  = vectorizer.transform(X_test_text)


# ---------- Models ----------
```

```
# Logistic Regression (baseline)

lr = LogisticRegression(max_iter=1000, class_weight="balanced")

lr.fit(X_train, y_train)

lr_pred  = lr.predict(X_test)

lr_proba = lr.predict_proba(X_test)[:,1]



# MLP (deep learning)

mlp = MLPClassifier(hidden_layer_sizes=(128,64), activation="relu",

      solver="adam", max_iter=20, random_state=42)

mlp.fit(X_train, y_train)

mlp_pred  = mlp.predict(X_test)

mlp_proba = mlp.predict_proba(X_test)[:,1]



# ---------- Metrics ----------

def summarize(y_true, y_pred, y_score):

      acc = accuracy_score(y_true, y_pred)

       prec, rec, f1, _ = precision_recall_fscore_support(y_true, y_pred, average="binary",
zero_division=0)
```

```
auc_val = roc_auc_score(y_true, y_score)
    return {"Accuracy":acc, "Precision":prec, "Recall":rec, "F1":f1, "AUC":auc_val}


lr_m  = summarize(y_test, lr_pred,  lr_proba)
mlp_m = summarize(y_test, mlp_pred, mlp_proba)


metrics = pd.DataFrame([
    {"Model":"Logistic Regression", **lr_m},
    {"Model":"Neural Net (MLP)",    **mlp_m},
])
metrics.to_csv(OUT_DIR/"metrics_summary.csv", index=False)
with open(OUT_DIR/"metrics_summary.json","w") as f:
json.dump(metrics.to_dict(orient="records"), f, indent=2)


print("\n[RESULTS] Metrics")
print(metrics.to_string(index=False))
```

```
# ---------- Figures (Chapter 4) ----------
def save_confusion_matrix(cm, title, filename):
    plt.figure(figsize=(6,5))
    plt.imshow(cm, interpolation="nearest", cmap=plt.cm.Blues)
    plt.title(title); plt.colorbar()
    classes = ["Non-Toxic","Toxic"]; ticks = np.arange(2)
    plt.xticks(ticks, classes, rotation=45); plt.yticks(ticks, classes)
    thresh = cm.max()/2.0
    for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
    plt.text(j, i, format(cm[i,j],"d"),
    ha="center", va="center",
    color="white" if cm[i,j] > thresh else "black")
plt.ylabel("True Label"); plt.xlabel("Predicted Label")
plt.tight_layout(); plt.savefig(FIG_DIR/filename, dpi=300); plt.close()
cm_lr  = confusion_matrix(y_test, lr_pred)
cm_mlp = confusion_matrix(y_test, mlp_pred)
save_confusion_matrix(cm_lr, "Confusion Matrix – Logistic Regression",
"fig_4_1_cm_logreg.png")
```

```
save_confusion_matrix(cm_mlp, "Confusion Matrix – MLP",  "fig_cm_mlp.png")


# ROC curves
fpr_lr,  tpr_lr,  _ = roc_curve(y_test, lr_proba)
fpr_mlp, tpr_mlp, _ = roc_curve(y_test, mlp_proba)
auc_lr,  auc_mlp   = auc(fpr_lr, tpr_lr), auc(fpr_mlp, tpr_mlp)
plt.figure(figsize=(7,6))
plt.plot(fpr_lr,  tpr_lr,  lw=2, label=f"Logistic Regression (AUC={auc_lr:.3f})")
plt.plot(fpr_mlp, tpr_mlp, lw=2, label=f"MLP (AUC={auc_mlp:.3f})")
plt.plot([0,1],[0,1],"--",color="gray")
plt.title("ROC Curves – Logistic Regression vs MLP")
plt.xlabel("False Positive Rate"); plt.ylabel("True Positive Rate")
plt.legend(loc="lower right"); plt.grid(True, alpha=0.3)
plt.tight_layout(); plt.savefig(FIG_DIR /"fig_roc_comparison.png", dpi=300); plt.close()
# Performance comparison bars
labels = ["LogReg","MLP"]
accs = [lr_m["Accuracy"], mlp_m["Accuracy"]]
f1s  = [lr_m["F1"],       mlp_m["F1"]]
x = np.arange(len(labels)); w = 0.35
```

```
plt.figure(figsize=(7,6))

plt.bar(x - w/2, accs, width=w, label="Accuracy")

plt.bar(x + w/2, f1s,  width=w, label="F1-score")

plt.xticks(x, labels); plt.ylim(0,1.0)

plt.title("Performance Comparison – LogReg vs MLP")

plt.ylabel("Score"); plt.legend()

plt.tight_layout(); plt.savefig(FIG_DIR /"fig_performance_bars.png", dpi=300); plt.close()

print("\n[B] Saved figures to 'figures/' and metrics to 'outputs/'.")
```