CRITICAL COMPARISON OF

NATURAL LANGUAGE PROCESSING

(NLP) APPROACHES ON BIG DATA

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF EDINBURGH NAPIER UNIVERSITY FOR THE DEGREE OF BSC (HONS) CYBER SECURITY

SCHOOL OF COMPUTING

April 2023

Authorship Declaration

I, Keeleigh T, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed:

Date: 13/04/2023

Matriculation no:

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

# ABSTRACT

As sources of data around the world continue to grow and develop, the need for up-to-date and specialised data analysis increases. Within intelligence, data must be processed and analysed quickly in order to assess cyber threat and malicious intent. One way of triaging big data is Natural Language Processing (NLP). This project looked to find the most accurate Python library that would correctly classify email content within the Enron dataset. The three libraries compared in this project were NLTK, SpaCy and Gensim.

Secondary research was conducted by critically reviewing existing literature and studies; this review outlined the current state of big data analysis and relevant NLP theory as well as existing studies that compared critical NLP infrastructure. Similar studies comparing NLP libraries were also reviewed to understand current theories and opinions. From this research a method was designed to implement three models, one for each library. Each model used relevant NLP pre-processing, vectorization, text feature extraction and model training through a machine learning classifier.

Evaluative methods included analysing performance metrics and confusion matrices gathered after model training. The results were critically analysed and compared with a focus on why each model performed the way it did when comparing pre-processing pipelines to performance metrics. A manual evaluation of extracted features for each model was carried out to further understand how each model classified text and whether these features could be used to determine importance of an email.

The results from this project showed NLTK to be the best performing library scoring highest in terms of model accuracy and precision, showing NLTK to have the most effective preprocessing pipeline.

# Table of Contents

[ABSTRACT 3](#_Toc45875)

[Table of Contents 4](#_Toc45876)

[List of Tables 5](#_Toc45877)

[List of Figures 6](#_Toc45878)

[1 INTRODUCTION 7](#_Toc45879)

[1.1 Background and Motivation 7](#_Toc45880)

[1.2 Aim 8](#_Toc45881)

[1.3 Objectives 8](#_Toc45882)

[1.4 Project Timeline 9](#_Toc45883)

[2 LITERATURE REVIEW 10](#_Toc45884)

[2.1 Introduction 10](#_Toc45885)

[2.2 The Background and Theory of Natural Language Processing 10](#_Toc45886)

[2.3 Natural Language Processing Components 12](#_Toc45887)

[2.4 NLP Python Libraries 15](#_Toc45888)

[2.5 Big Data and Analysis Challenges 16](#_Toc45889)

[2.6 Conclusion 17](#_Toc45890)

[3 METHODOLOGY 18](#_Toc45891)

[3.1 Research Methods 18](#_Toc45892)

[3.2 Project Management 19](#_Toc45893)

[3.3 Domain-specific Methods 19](#_Toc45894)

[3.4 Experimental Methods 20](#_Toc45895)

[3.5 Evaluation Methods 22](#_Toc45896)

[4 IMPLEMENTATION AND RESULTS 24](#_Toc45897)

[4.1 Data Formatting 24](#_Toc45898)

[4.2 Data Pre-Processing 24](#_Toc45899)

[4.3 Model Evaluation 27](#_Toc45900)

[5 DISCUSSION 30](#_Toc45901)

[5.1 Performance Metrics 30](#_Toc45902)

[5.2 Determination of Importance 33](#_Toc45903)

[5.3 Summary of Findings 34](#_Toc45904)

[6 CONCLUSION 35](#_Toc45905)

[6.1 Project Summary 35](#_Toc45906)

[6.2 Bias and Error 35](#_Toc45907)

[6.3 Future Work and Development 36](#_Toc45908)

[6.4 Self-appraisal 36](#_Toc45909)

[7 REFERENCES 38](#_Toc45910)

# List of Tables

Table 1 - Precise Training/Test Data Sample Sizes For Model Evaluation .......................... 21

Table 2 - Metrics Used to Evaluate Model Performance and Effectiveness (Bountakas et al.,

2021) & (Hasan et al., 2021) ............................................................................................... 23

Table 3 - A table displaying example email content parsed through each library's pre-

processing pipeline ............................................................................................................. 26 Table 4 - Summary of model performance metrics for all models ........................................ 30

# List of Figures

Figure 1 - Gantt chart Timeline of Project until Submission ................................................... 9

Figure 2 - Recent Developments Within NLP Since 2000 (Khurana, 2022) ......................... 10

Figure 3 - Methodology and Implementation Project Management Chart ............................ 19

Figure 4 - Method Flow Chart Outlining the Steps For each Library Execution .................... 21

Figure 5 - Code snippet displaying data formatting by splitting content types ...................... 24

Figure 6 – Top 5 formatted data to be parsed through NLP technology .............................. 24

Figure 7 - NTLK Pre-Processing Steps ............................................................................... 25

Figure 8 - Gensim Pre-Processing Steps ............................................................................ 25

Figure 9 - SpaCy Pre-Processing Steps .............................................................................. 25

Figure 10 - Code snippet showing vectorization, feature extraction and model training ....... 27

Figure 11 - NLTK Confusion Matrix ..................................................................................... 27

Figure 12 - NLTK Model Performance Metrics .................................................................... 27

Figure 13 - Gensim Confusion Matrix .................................................................................. 28

Figure 14 - Gensim Model Performance Metrics ................................................................. 28

Figure 15 - SpaCy Confusion Matrix ................................................................................... 29

Figure 16 - SpaCy Model Performance Metrics ................................................................... 29

Figure 17 - NLTK Top 10 Weighted Features ...................................................................... 33

Figure 18 - Gensim Top 10 Weighted Features................................................................... 33

Figure 19 - SpaCy Top 10 Weighted Features .................................................................... 33

# INTRODUCTION

## Background and Motivation

According to a National Statistics survey in 2021, internet users in the UK made up 92% of the adult population (ONS, 2021), meaning 92% of adults were creating an online footprint and generating data, some of which would end up being personal and sensitive information protected by data protection laws. The volume of data collected by different organisations, including the intelligence community (IC) continues to grow as the number of data sources increases, for example, social media, internet of things (IoT) and marketing sources. Due to this increase in data quantity, the term big data refers to datasets that exceed the capabilities of traditional software tools designed to manage and analyse data (Al-Sai et al., 2019).

Standard perceptions of big data revolve around various “V” characteristics – volume, variety, velocity, value and veracity (Landon-Murray, 2016); if data has these characteristics, it is perceived as big.

The new characteristics of big data bring new challenges when it is used in the analysis community. Due to the volume and variety of big data, current hardware and software is not able to cope with the vast amount of data. Common future trends of big data include data analysis and mining and within the IC, this can mean building capable software that could reveal unknown relationships or intel that previous data analytics could not uncover (LandonMurray, 2016). If more data can be processed, more analysis can be conducted meaning a faster approach at dissolving conflict and protecting assets from threats such as counterterrorism and criminal activities on home soil.

Natural Language Processing (NLP) is a subset of Artificial Intelligence (AI) and was developed to make software understand the human language and meaning of speech in both written and spoken forms (Khurana, 2022). It is difficult for a computer to understand natural language because they are complex and voluminous, leading to infinite sentences and semantics that can have multiple meanings (Chowdhary, 2020). If configured correctly for a dataset, an NLP tool can be effective when analysing data quickly and accurately, giving a data overview to an analyst within minutes. Data classification using NLP is increasing in demand not only to determine online behaviour but also to gather knowledge on millions of users worldwide and analyse media and social network information to improve user performance and remove unwanted behaviour (Pranckevičius & Marcinkevičius, 2017). Classification of text in this modern research area is important when determining sentiment or polarity that could indicate opinion and emotion which can be analysed in many industries such as, healthcare, political campaigns or financial institutions (Pranckevičius & Marcinkevičius, 2017).

During the investigation into Enron’s accounting and financial scandal in 2001, the company’s email dataset was made public and posted online by the Federal Energy Regulatory Commission (FERC) not only for investigative transparency, but for historic and research purposes, mainly to help with spam detection (Hasan et al., 2021). This corpus of data is widely known as the largest public domain database and is used in many research projects because it holds real-world examples of datatypes such as names, addresses, dates, and times which are similar types of data that are analysed by the business area. Due to operational restrictions within the workplace, finding unsensitised and unclassified data would be difficult, therefore the Enron data set is a perfect example of a similar corpus of emails that can be analysed using NLP.

The motivation of this project is to find an NLP approach that is suitable for the type of data found within the Enron dataset. As the Enron dataset is publicly available, it is easier and unrestricted to analyse so the results can be transferred to private and sensitive datasets within the workplace. The incentive of this report is to build on current literature by comparing NLP approaches to find one that is best suited to the data types found within the Enron dataset. The idea being that an NLP tool can draw conclusions based on sentiment and context to support knowledge gathering and decision making (Pranckevičius & Marcinkevičius, 2017). If an analyst can understand organisational practice from emails, predictions can be made about organisational activities and persona profiles.

## Aim

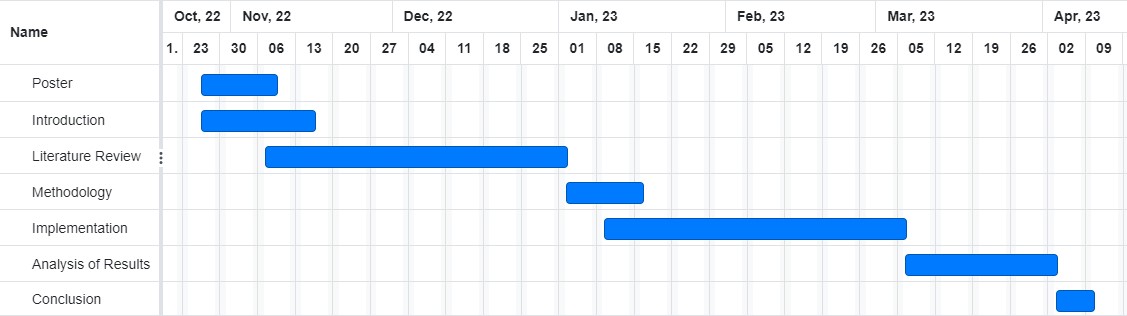
To critically evaluate and compare NLP libraries to identify the best analysis method for email datasets for transference to sensitive data in the workplace.

## Objectives

1. Critically review literature surrounding NLP and big data analysis to supply a theoretical underpinning of current methodologies and existing works.
2. Develop a method from candidate processes and toolkits identified from literature.
3. Execute the primary research and implement chosen methodologies to achieve the project aim.
4. Critically analyse statistical results in a discussion comparing performance metrics and data output taken from the experiment results.
5. Conclude with a summary of findings and a reflective synthesis of project development.

## Project Timeline

To achieve the project aim, the timeline shown in Figure 1 will be followed as closely as possible to find a project outcome and publish results.



*Figure 1 - Gantt chart Timeline of Project until Submission*

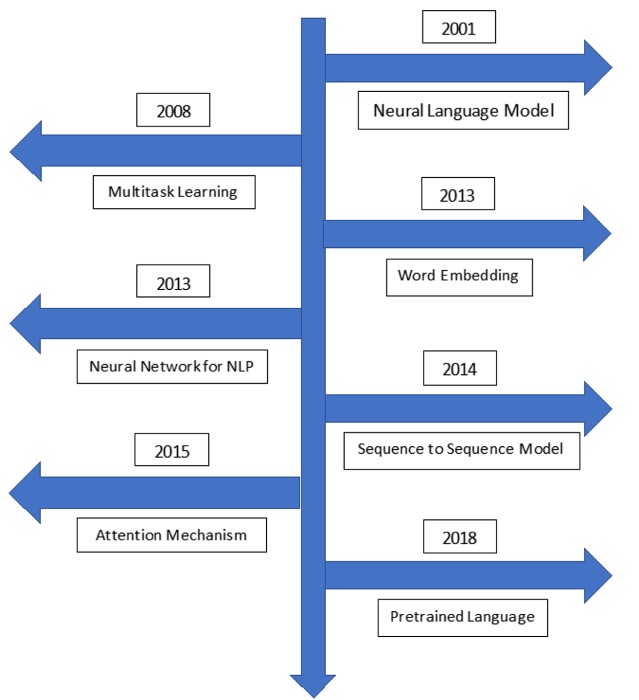
# LITERATURE REVIEW

## Introduction

The goal of this literature review is to supply an understanding of NLP and its toolkits and how this technique can be used in big data analysis. This review will explore the problem, relevant theory, and existing work in the form of a thematic literature review. It will review the current literature surrounding the theory and current state of the art of NLP as well as looking into why big data analysis is a challenge. The closing section will critically review NLP techniques and toolkits, their purposes, and existing applications.

## The Background and Theory of Natural Language Processing

NLP dates back to the 1940s with the first computer-based language application being Machine Translation (MT) in World War II with the translating and breaking of enemy codes. (S. R. Joseph et al., 2016). MT work quickly became a basis for Artificial Intelligence (AI) in the 1960s with signature work surrounding speech recognition and pattern analysis followed by grammar theory and user profiling in the 1980s (Khurana, 2022). The 1990’s led way for computer-generated, linguistically coherent speech that could communicate with the natural language; it was at this stage, the term NLP was coined and given its own branch of research within AI (Khurana, 2022). Because of the variety of possibilities and objectives that could potentially be achieved in NLP, the research field has received much attention in the last two decades; the growth of the NLP field since 2000 can be seen in Figure 2 with the most recent development of pre-trained language using neural networks (Khurana, 2022). Joseph et al. (2016) credit the growth of NLP to the maturation of technologies including the internet, increased memory size and the availability of large volumes of text online.



*Figure 2 - Recent Developments Within NLP Since 2000 (Khurana, 2022)*

The term deep learning refers to the use of neural networks of different types for machine learning, and recently, deep learning methodologies have been applied to NLP in different use cases, most commonly with image classification (Wu et al., 2016). Yin et al. (2017) builds on this theory by explaining that deep learning NLP techniques have been developed because of the high performance of deep neural networks (DNN), which can be further divided into two categories, convolutional neural networks (CNN) and recurrent neural networks (RNN). DNN’s consist of three layers: the input layer, the hidden layers and the output layers, with the CNN’s moving through these layers hierarchically and RNN’s moving through them sequentially (Zainab et al., 2019).

Yin et al. (2017) completed a study comparing CNN and RNN for simple NLP tasks, comparing the same dataset in each type of network with zero pre-trained data. They found that RNNs performed better especially in phrase recognition and sentiment detection, but they did conclude that varying batch size varies the performance of a DNN, suggesting that a CNN can perform just as well as an RNN if network parameters are fine-tuned and optimised. Zainab et al. (2019) agrees with this conclusion, adding that CNNs have proved better at text summarisation and reading checks for substantial amounts of data. Whilst Zainab et al. (2019) did not use NLP in their study, they used a big data set of text-based images to train their networks on, and so their findings are still relevant in the comparison of CNN and RNN.

Wu et al. (2016) argue the challenges surrounding deep learning with NLP and believe that such techniques are ineffective when inferring and making decisions based on training data, although they argue this is more of a general challenge of NLP, not just when combined with deep learning. They argue that if a dataset is small, a neural network will lose its efficiency because they are data hungry models that will not produce accurate results if not trained properly. Because the variety of vocabulary can vary per dataset, Wu et al. (2016) argue that no matter how much data there is, there will always be grammatical and lexical knowledge that the training data cannot cover, but Khurana (2022) suggest that over time as models are exposed to different dialects on a regular basis, they improve with broader use, therefore gaining in accuracy. Khurana (2022) suggests the main challenges of NLP root from the variety of natural language, explaining that words and phrases can have multiple meanings, some sentences may contain sarcasm and irony, but Joseph et al. (2016) explain that handcrafting these types of data to feed into a model is time-consuming and therefore may halt certain research areas.

## Natural Language Processing Components

### Pre-Processing Techniques

A fundamental component of NLP is pre-processing; which is used to clean the data by removing irrelevant and redundant data from the corpus (Bountakas et al., 2021). Raw formats of data are not normalised or structured therefore by applying standard processing techniques, a structured version of the data can then be used for feature extraction and classification stages based on the need of the application (Sarkar, 2019). Also known as data wrangling, pre-processing raw data can optimise the data-set for the purpose of the model, increasing accuracy and reproduction in later research (Hasan et al., 2021).

Hasan et al. (2021) use three steps as part of their pre-processing stages; these include normalisation (removing stop words, words under three letters and punctuation), tokenisation and lemmatization. Tokenisation involves splitting the raw text into a list of words that can each be referred to as a token; as a word can be used in varying forms of its base word, the process of lemmatization follows where these variations are removed to get back to the base form found in the dictionary allowing for optimised feature extraction and classification (Hasan et al., 2021). Bountakas et al. (2021) agree with Hasan et al. (2021) by using the same three step process as part of their proposed approach before the textual feature extraction phase.

Both Hasan et al. (2021) and Bountakas et al. (2021) analyse machine learning techniques involving NLP for the Enron dataset, the first for spam classification, the second for phishing detection. Although both methodologies focused on machine learning techniques involving neural networks and classifiers, their pre-processing of text using standard NLP functionalities is still applicable to this project and are examples of successful pre-processing on the Enron dataset.

Sun et al. (2017) suggests using several pre-processing steps of cleaning and normalising data before the extraction phase, however their review compares different NLP techniques for opinion mining, also known as sentiment analysis, where an opinion of the text is drawn from the model. For example, the model may decide if a positive or negative opinion is displayed in the text; for this project, a simple classification model is applicable, so opinions do not need to be drawn based on sentiment. Although Sun et al. (2017) use tokenisation and lemmatization, they also suggest Part of Speech (PoS) tagging and parsing. PoS tagging is used to identify word labels, such as adjective, noun, verb which can be particularly useful when identifying opinions or the main focus of an email per say (Sun et al., 2017). Parsing is a technique which produces a tree of grammatical structures used in each sentence to give an indication of the type of sentence or question in a text block (Sun et al., 2017).

### Text Feature Extraction Methods

After normalising the dataset into a structured format using pre-processing techniques, textual feature extraction methods can be applied so that the data can be passed into ML algorithms and classifiers, with the three most common methods being TF\_IDF, Word2Vec and BERT (Hasan et al., 2021). The TF-IDF method uses two terms: Term Frequency (TF) which depicts the number of times a word appears in the text taking into consideration bias for larger texts and Inverse Documents Frequency (IDF) which depicts a word’s importance using a ratio of the number of texts to the number of texts containing the given word (Kasri et al., 2019). (Bountakas et al., 2021) explains that TF-IDF works by calculating an overall weight from a series of mathematical probability calculations, indicating the importance of a word in a collection of documents using the formulas shown below:

𝑓𝑡, 𝑒

𝑇𝐹(𝑡, 𝑒) =

∑𝑡 ′𝜀 𝑒 𝑓𝑡′, 𝑒

𝑛

𝐼𝐷𝐹(𝑡, 𝐶) = 𝑙𝑜𝑔

|𝑒𝜀𝐶 ∶ 𝑡𝜀𝑒 |

𝑇𝐹 − 𝐼𝐷𝐹(𝑡, 𝑒, 𝐶) = 𝑇𝐹(𝑡, 𝑒) ∗ 𝐼𝐷𝐹(𝑡, 𝐶)

Word2Vec outputs a vector representation of the context of words within the text itself, taking into consideration the semantic similarity to the other words using a neural network to learn connections (Bountakas et al., 2021). BERT takes these techniques one step further by reading texts from both directions in order to understand the semantic of a word based on its previous and next words; although BERT is relatively new, this model facilitates a new area of predictive NLP to create new text based on the input (Bountakas et al., 2021).

Cahyani & Patasik (2021) compared both TF-IDF and Word2Vec models for emotive language classification on tweet data and found that TF-IDF produced better accuracy and precision scores compared to the vectorisation method previously discussed. Although their study classifies text based on polarity and sentiment, the results give a clear indication that the best feature extraction method to use for text classification applications is TF-IDF. Kasri et al. (2019) completed a study on feature extraction methods for Arabic sentiment analysis comparing TF-IDF with other Arabic based techniques. These methods would not apply to this project due to the dialect differences, but regardless of technique, Kasri et al. (2019) found that the TF-IDF method still achieved the best results when the data was put through classifiers after feature extraction.

### Types of Classifiers

To gain the sentiment of text within a dataset, the text must first be classified in order to differentiate the text based on certain features; this is done using machine learning models known as classifiers (Pranckevičius & Marcinkevičius, 2017). Ma et al. (2020) argue that the Naïve Bayes (NB) classifier is a fast and highly accurate model with easy implementation and because of this, it has been known to outperform other classification methods. Ma et al. (2020) explain that the Bayes algorithm uses a graphical model to represent probability relations between random features; the formula is shown below:

𝑃(𝑥|𝑐)𝑃(𝑐)

𝑃(𝑐|𝑥) =

𝑃(𝑥)

A Decision Tree can also be used to create all the possible solutions based on conditionals therefore when testing data is applied, predictive analysis can be applied by following the training decisions; a Random Forest (RF) builds on this classifier by merging multiple decision trees to produce increasingly stable predictions (Smitha & Bharath, 2020). The Support Vector Machine (SVM) and Logistic Regression (LR) classifiers are both used for linear and binary classification, but Smitha and Bharath (2020) argue that LR is better at removing outliers while SVM methods are considered more effective by maximising margin points.

In a study completed in 2017, Pranckevičius & Marcinkevičius compared the NB, RF, Decision Tree, SVM, and LR classifiers on multi-class texts and concluded that the LR method achieved the highest accuracy with the Decision Tree receiving the lowest accuracy values. Tusar and Islam (2021) agree with these results through their classifier comparison on tweet text classification; although the data types are different, similar NLP methods are compared but with varying ML classifiers.

Smitha and Bharath (2020) disagree with this conclusion and found that the SVM algorithm worked better, because it was the fastest method meaning there was less computation time to train the data. Miao et al. (2021) compared different library pre-processing techniques with varying classifiers and concluded that the best performing classifier was RF and when compared to Naïve Bayes, was much slower, but produced similar performance scores of around 87%. Miao et al. (2021) also stated that SVM took too long to process (taking an average of 250 minutes to train one model), which contradicts the theory of Smith and Bharath (2020), suggesting that SVM would be impractical and infeasible to train and classify data.

## NLP Python Libraries

Nagpal & Gabrani (2019) completed a paper surrounding the advantages and disadvantages of using Python for data analytics, arguing the main benefits were, high portability and flexibility, a fine balance between high and low level programming as well as the wide variety of open source libraries that are free to use. Zahidi et al. (2019) agree with these advantages adding that Python would be better used than Java because of its lower complexity, useful NLP libraries and larger corpus of DL libraries that can assist NLP coding. Although comparing Python with Java and individual libraries, Zahidi et al. (2019) concludes that every language and library will have its own strengths and weaknesses so the preferred choice would depend on the need of the coder and the requirements of the application. As well as the many advantages that Python can give a developer, Nagpal & Gabrani (2019) explain that Python is a relatively slow language compared to other compiled languages and because it is dynamically typed, more runtime errors occur meaning applications require more testing thereby increasing development time.

NLTK is known as the “leading platform for NLP” and is used to teach NLP as it provides a variety of functionality in a self-contained library that can achieve multiple NLP tasks (AlAmrani et al., 2021). TextBlob was developed to process textual data in sentiment analysis and classification, specifically focusing on subjectivity and polarity of text in order to make decisions based on the opinion and positivity or negativity of the textual data (Al-Amrani et al., 2021). Gensim, another Python library, is a vector-based solution that can be used to process massive datasets which, unlike NLTK, uses statistical machine learning; Gensim uses implementations of the Word2Vec algorithm explained earlier (Al-Amrani et al., 2021).

Zahidi et al. (2019) argues that NLTK offers a simple but vast variety of functionality that is easy to use and because it is based on object-oriented programming, offers fast and literate programming; however, the above argues that unlike NLTK, Gensim can be seen as the preferred option for processing large datasets because it is fast and memory efficient, but Gensim was not built as an all-purpose library and has limited functionality compared to NLTK. Nagpal & Gabrani (2019) argue that NLTK is a highly popular Python library because it was developed solely for information retrieval and text classification. (Zahidi et al., 2019) argues that CoreNLP, although widely used, is only preferred by some data scientists because it is API based and therefore accessibility is preferred over accuracy and technicality.

Al Omran & Treude (2017) argue that choosing the right and correct NLP library is a critical choice in any process that is used to analyse software artefacts, specifically because the language is technical and not understood by most people. By assessing over 1350 conference papers and comparing four main Python libraries (NLTK, SpaCy, SyntaxNet and CoreNLP) in an NLP software application, Al Omran & Treude (2017) concluded that spaCy achieved the best accuracy but with vast room for improvement.

Smelyakov et al. (2020) compared the pre-processing algorithms on NLP applications by analysing the NLTK and SpaCy libraries effectiveness of stop word removal, stemming and lemmatization; by evaluating run time and accuracy scores, it was concluded that SpaCy was most effective and ran faster during the lemmatization phase. They did find that NLTK performed faster at stemming but left lower quality results in comparison to SpaCy. Contradicting to these findings, Miao et al. (2021) compared different Python libraries with different machine learning classifiers and concluded that NLTK and SpaCy performed similarly for all types of classifier with NLTK leading in performance by small margins for each; suggesting the type of classifier will not affect the performance of the pre-processing libraries.

## Big Data and Analysis Challenges

In 2014, big data was characterised by three V’s: variety, velocity and volume, with each characteristic measuring a dimension of big data that traditional architectures and data storage methods are unable to cope with (Kwon et al., 2014). Sandhu (2022) disagrees with this theory explaining that because of a new generation of architecture, the three V’s may be outdated and can be extended to five V’s: variety, velocity, volume, value, and veracity. Lee (2017) believes that the primary cause of this latest big data expansion, is due to the IoT applications that create diverse types of data ranging from images to audio and video. Because IoT devices generate data over the internet without human intervention, it is predicted that this form of data generation will overtake social media and e-commerce websites as the sole sources of big data (Lee, 2017).

Analysis of big data is difficult to perform using standard data analytics because the accuracy and effectiveness of these methods are significantly reduced due to the big data characteristics outlined previously (Hariri et al., 2019). A key factor when conducting data analysis is speed and timeliness. Due to the volume of data, high computing power is needed to deliver timely results repeatedly, therefore innovative algorithms and methods are required to keep up with demand that the human workforce cannot match (Mohanty et al., 2015). Within the intelligence community, data analysis plays a vital role in combating threats and dissolving conflict; the faster data analysis can be achieved, the quicker the decision-making process of vital, sometimes life-saving future actions.

Hariri et al. (2019) argue that advanced data analysis techniques have the capability to reshape big data into smart data meaning critical information is easier to obtain in large datasets, therefore improving speed, accuracy, and effectiveness of analysis; by having the capability to analyse large datasets, organisations can improve their post-analysis business decisions. Mohanty et al. (2015) agrees with Hariri et al. (2019) but add that analysis techniques are only beneficial when tailored to the dataset or data type and this lack of accuracy only increases with size, suggesting that generic methods are not applicable or effective on each organisation’s dataset and that tailored techniques will have a greater degree of accuracy. The downside to this suggestion is that each organisation and business will have to build their own, unique analysis algorithm for each dataset they want to analyse; this is an impractical and huge expense.

## Conclusion

The critical analysis of literature explored in this section reviewed NLP techniques and processes to understand how these can be best applied to big data analysis. The theory and applications of NLP were discussed to understand how this AI technique could be used in data analysis within the intelligence community. Different pre-processing techniques and classifiers were discussed with earlier works to understand the strengths and applications of different functionalities to find the best suited methodologies for this project.

By exploring factual data and earlier works, this literature review was able to conclusively identify certain methods that can be used in this project to achieve the aim set out in chapter one. This literature review found that using machine learning classifiers to train models is best suited solution for text classification and analysis problems which is the most applicable scenario to the datatypes seen in the Enron dataset. It was also concluded that a simple three step pre-processing technique is needed to clean and structure the data, consisting of normalisation, tokenisation, and lemmatization.

After pre-processing, the next stage of the NLP method is the feature extraction stage; this review analysed both TF-IDF and Word2Vec alongside the BERT algorithm and found that TF-IDF produced better accuracy and precisions scores in the existing works that were reviewed. Another important stage of NLP machine learning is classifying data based on its textual features and this review concluded that although Logistic Regression and SVM achieved better accuracy in existing works, the Naïve Bayes classifier works just as well, but with a much simpler implementation.

When comparing different libraries, this review concluded that most studies found SpaCy to be the best performing library with only one reviewed study suggesting NLTK led the way in terms of model performance. Although literature supplied an underpinning of theory and current work, this report aims to find related results, but for a specific data type and set for intelligence analysis purposes.

# METHODOLOGY

This section will outline and document the methods used to achieve the project aim. The methodology has been informed by the concepts discussed in the literature review which looked at the current state of big data and the present analysis challenges, the theory of NLP and a review of the ML techniques that are used in the training process. The following will outline the research methods used, the technologies and tools that will be employed, the main method with step-by-step explanation along with the evaluative and performance metrics that will be used to analyse results. There will also be an acknowledgement of bias, ethical concerns and legal requirements surrounding data collection and processing.

## Research Methods

### Literature Collection Strategy

The thematic literature review supplied a theoretical underpinning of current theory and methodologies surrounding the themes of this project. To understand all earlier works and findings to support the theory of this project, a range of mediums were used such as peer reviewed journals, conference papers, academic books, journal articles and websites. The main source used for finding this literature was Google Scholar, which is an approved and favoured source of published work for academics to find content relevant to their work. The sources of literature used in this review came from reputable and approved sources, such as well-known publishers (IEEE, Springer and ACM) and statistical data came from the Office for National Statistics which is a recognised national UK institute.

By searching for key terms related to NLP and big data, recently published sources with a strong theoretical understanding of the topic were selected as well as those papers that conducted experimentation on the relevant techniques and methodologies that will be used in this project. For example, there were many sources that compared pre-processing techniques, so only those that were relevant to text and sentiment analysis were considered. In addition to Google Scholar, factual pieces of information in the review came from published books found from published books found in Google Books, for example, Sarkar’s book *Text Analytics in Python* (2019), discusses the fundamentals and theory of NLP, with a focus on the Pythonic syntax needed to program applications.

## Project Management

The time management of this project is closely following the Gantt chart displayed in Figure 1 and at this stage of the project, both the introduction and literature review have been completed leading to the development of the methodology which is inspired by the findings of the literature review. The next stage of the project should be completed in eight weeks and will be split into three sections, the development of the experimental methods, the implementation of methods and finally, the evaluation stage where results are gathered; a breakdown of this time is displayed in Figure 3.

**Development of**

**Methodology (2 weeks)**

•

Domain

-

specific methods

•

Experimental Methods

•

Evaluative Methods

**Implementation (6 weeks)**

•

Testbed Setup

•

Data Collection

•

Script Development

**Evaluation (1 week)**

•

Testing

•

Results gathering

*Figure 3 - Methodology and Implementation Project Management Chart*

## Domain-specific Methods

### Development Environment

There are many popular choices that software engineers use to develop code, some of which include, Jupyter Notebooks, Visual Studio Code, Google Collab, and AWS. For the boundaries and limitations of this project, a free and easy to access training environment must be used to provide the correct programming tools and techniques to carry out natural language processing, as well as handle the large volume of data needed to train models. Because of the requirements listed above, the commonly used Jupyter Notebooks platform will be used to develop the experimental methods and carry out evaluation techniques. Jupyter Notebook is an open source, interactive application that is easy to share and edit also allowing an easy display of visual representations of data (Silaparasetty, 2020).

### Data Collection and Processing

For the constraints of this project, the Enron email corpus will be downloaded from the data science company, Kaggle; this website offers a Jupyter Notebooks environment with a free repository of over 50,000 public datasets. Other sources of data on the internet claim to provide an authentic copy of the corpus, however these sources cannot be validated therefore the data should come from a reputable and reliable source that has been licensed. Some sources of data also hold the raw formats, including over a hundred folders holding individual files of emails which would need to be significantly pre-processed and formatted to extract relevant data, therefore finding a version of the data which is in ready state is crucial as to not waste vital time extracting information.

Kaggle is the chosen domain for developing the experiments for this project because it can provide both a free Jupyter Notebooks training environment and the Enron dataset in a freely available spreadsheet format for easy importation. After collecting the data, but before the experiments can be conducted, the data must be processed into a suitable format that can be easy manipulated and extracted by a Python script. For this reason, functionality will be included to transform the spreadsheet of data into a Python Pandas data frame to extract the specific fields such as sender, receiver, email subject and content.

### Bias, Ethics and Legalities

A key part of data handling and management is the ethical and legal considerations when sourcing data and processing it, especially if data is disseminated outside of the experiment. Collections of data, like that of Enron, are normally bound by privacy and restrictive measures because of the high volume of extremely sensitive personal information pertaining to realworld examples. However, before the Enron corpus was made public, employee information was removed so that the data could be used as part of research studies. Ethically, it is important to note that this experiment will not directly engage with any of the named members of Enron as the focus is solely on the email content.

As with any investigation, there can be bias from both the researcher, the supporting literature, and the reader of the report. There has been effort to reduce bias when choosing relevant literature, aiming to provide a wide range of sources from different academic institutions and range of perspectives on the topics discussed. A significant effort will also be made to reduce bias during the experiment by remaining neutral and choosing an encompassed word set to capture the true outcome of the models used. Finally, by using statistical data, the risk of bias when discussing results will be significantly decreased as there will be factual data to analyse reducing the need for opinion based results.

## Experimental Methods

### Training/Test Data

In total, the Enron dataset holds just over 517,000 unique records, each a different email. Because this is a voluminous dataset, 5% of the data will be extracted randomly using the sample function in the Python Pandas library. The random sample and will be further split into training and test data which is a common approach to validating a machine learning model; the models are fitted on the training set and validated using the testing data allowing different models to be evaluated and compared based on testing data performance (V. R. Joseph, 2022). This project will split the sample data into an 80:20 ratio with 80% of the data being used to train the models and 20% of the data being used as testing data to measure model performance. Table 1 shows the sample size that will be used in the project and how this sample data will be split into 80:20 ratio for model evaluation. *Table 1 - Precise Training/Test Data Sample Sizes For Model Evaluation*

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset Size** | **Sample Size** | **Training Data** | **Testing Data** |
| 517,401 | 51,740 | 41,392 | 10,348 |

### Method Execution

The practical method for this project has been summarised and outlined in the project flowchart which has been visualised in Figure 4 with each stage of the flow chart being explained below. The first two stages are separate to the experiment itself but are important stages that will be conducted primarily which involve sourcing and formatting data. The chosen libraries to be compared are NLTK (Model 1), Gensim (Model 2) and SpaCy (Model 3) with each execution following the same flow chart; these libraries were discussed in the literature review and were found to be simple yet effective in the field of NLP to allow for a more direct comparison.

Data

Sourcing

Data

Formatting

Data Pre

-

processing

Text Feature

Extraction

Data

Splitting

Model

Training

Model

Evaluation

*Figure 4 - Method Flow Chart Outlining the Steps For each Library Execution*

1. **Data Sourcing** – the Enron dataset will be sourced on Kaggle and will be transferred as a spreadsheet from the Kaggle dataset repository to the Jupyter notebook development environment within Kaggle itself. Using a Python library, 10,000 random emails will be taken from the corpus as sample data to train the model.
2. **Data Formatting** – once the data is sampled, it will be formatted using a series of Python libraries and functions to create a data frame. Irrelevant metadata will be removed (message ID, date, time) to extract the email content for training each model.
3. **Data Pre-Processing** – the formatted data will be put through a list of pre-processing techniques that were discussed in the literature review. The three stages that will be implemented at this point will be normalisation, tokenisation, and lemmatization, using each library’s standard pre-processing techniques. By carrying out these processes, the data can be filtered and cleaned ready for text feature extraction.
4. **Text Feature Extraction** – the TF-IDF text feature extraction method will be applied which will calculate the importance of each word based on mathematical probabilities outlined in section three. TF-IDF was chosen because of its improved performance in accuracy and precision scores over Word2Vec and BERT in earlier studies.
5. **Data Splitting** – In order to train and validate the models, data will be split into training and test data at an 80:20 ratio for reasons previously detailed.
6. **Model Training** – to allow sentiment to be learned by the machine, classifiers must be applied to the training data and research showed minor difference in classifiers therefore, for simplicity, the Multinomial Naïve Bayes classifier has been chosen and will be used as a constant in all three models. Using the same classifier in each model will allow for a direct comparison of each library without the influence of changing external variables.
7. **Model Evaluation** – A critical analysis of results gathered after model training will be used to compare the effectiveness and accuracy of each model.

## Evaluation Methods

At this stage of the project, the project flow chart shown in Figure 4 - Method Flow Chart

Outlining the Steps For each Library Execution should be at the final step called ‘Model Evaluation’. To understand the accuracy of the models created for each Python library, a series of performance metrics will be collected after each model is trained on a classifier. Three performance metrics will be collected along with confusion matrices which provide a visualisation the accuracy of the model classifier by differentiating between actual and predicted results.

A confusion matrix will be made of true Positive (TP), false positive (FP), true negative (TN) and false negative (FN) results and from this visualisation, other metrics can be calculated to give a statistical overview of the performance of each model. Each of these metrics can be seen in Table 2 which explains each metric, the mathematical formula used in calculations and how the metric result can be interpreted as a measure of performance.

These results will be compared and analysed to answer the research questions outlined in section two and to check if the project aim has been achieved. A further in-depth inspection of the words considered most important by the TF-IDF vectorizer will also be discussed to see which words class an email as important according to each model.

*Table 2 - Metrics Used to Evaluate Model Performance and Effectiveness (Bountakas et al., 2021) & (Hasan et al., 2021)*

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Definition*** | ***Formula*** | ***Metric***  ***Interpretation*** |
| **Accuracy** | The number of correctly classified important emails among the total sample size. |  | The higher this metric, the better the performance of the model. |
| **Precision** | The ratio of correctly classified important emails compared to the total number of emails that were classed as important. | 𝑇𝑃    (𝑇𝑃 + 𝐹𝑃) | The higher the precision score, the better the model predicts low FP and high TP results. |
| **Recall** | The ratio of correctly classified important emails compared to the number of actual important emails. | 𝑇𝑃    (𝐹𝑃 + 𝐹𝑁) | The higher this number is, the higher the model can classify emails. |
| **F1 Score** | This metric is used to balance precision and recall and calculates a weighted average of the two. | 𝑇𝑃    𝑇𝑃 +(𝐹𝑃 + 𝐹𝑁) | The best performing models achieve an  F1 score close to 1. |

# IMPLEMENTATION AND RESULTS

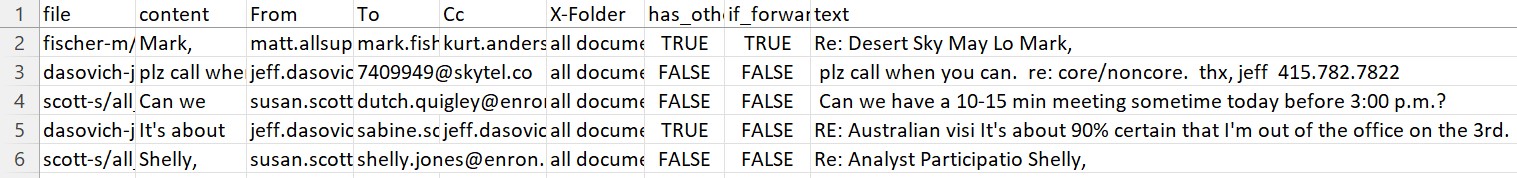
The following section aims to outline the implementation of the domain specific and experimental methods. The differences in processing by each library will be explained with a further outline of feature extraction and model training. Example code will be presented in the following sub-sections to demonstrate model experimentation. A full version of the script may be made available upon request. The final sub-section will display the results gathered after training each model to evaluate performance and accuracy; this will include, statistical metrics, confusion matrices and a list of words deemed important by each model.

## Data Formatting

The first step of the experiment involved taking a raw dataset and formatting data characteristics to fit the specification of each model. This meant converting a spreadsheet into a filtered Pandas data frame with email content split into separate columns. Most importantly, the email content had to be separated out to be parsed through pre-processing and classifier functions. An example of one step in the formatting process is shown in Figure 5, in which different content types are split into separate columns, e.g., forwarded emails are separated into two. The top five rows of the resulting data frame after all formatting processes have been completed is shown in Figure 6 in the form of a spreadsheet; the ‘text’ column contains all stripped email content to be passed through NLP functions.



*Figure 5 - Code snippet displaying data formatting by splitting content types*



*Figure 6 – Top 5 formatted data to be parsed through NLP technology*

## Data Pre-Processing

### Pre-Processing Pipelines

The focus of this experiment is to compare different Python libraries in their execution of processing natural language of email content to classify emails based on importance in machine learning models. Therefore, an examination of the pre-processing techniques must be carried out to understand the differences that may lead to differing model performance.

For each model, the pre-processing steps have been outlined through flow diagrams in Figures 7, 8 and 9, all of which have been created by the author. The NLTK and SpaCy pipelines follow similar steps in the same order with one difference. SpaCy uses a built in pipeline function called ‘nlp()’ which automatically pre-processes text using a series of known techniques. Comparing this to NLTK, Spacy uses two more steps, Part-of-Speech (POS) tagging and Named Entity Recognition (NER). These two techniques involve considering word labels such as adjective or noun and named entities such as names, companies, or quantitative values to better understand the text structure and entity relationships. By using extra techniques, this could filter the text for better model learning, or this could mean the text is processed and stripped too far, leading to lower precision when training the model. When comparing these two processes to Gensim, the steps are reversed by removing characteristics before creating tokens and stemming text; meaning words may be removed without considering the root token first.

Tokenisation

Stemming

Lemmatization

Normalisation

•

Stop Word Removal

•

Punctuation

Removal

*Figure 7 - NTLK Pre-Processing Steps (Author, 2023)*

Normalisation

•

Stop Word Removal

•

Punctuation Removal

Tokenisation

Lemmatization

*Figure 8 - Gensim Pre-Processing Steps (Author, 2023)*

NLP Pipeline

•

Tokenisation

•

Part

-

of

-

speech (POS)

Tagging

•

Lemmatization

•

Name Entity Recognition

(

NER

)

Stop Word Removal

Punctuation

Removal

*Figure 9 - SpaCy Pre-Processing Steps (Author, 2023)*

4.2.2 Pre-Processing Outputs

*Table 3 - A table displaying example email content parsed through each library's pre-processing pipeline*

### **MODEL PRE-PROCESSING EXAMPLE**

|  |  |
| --- | --- |
| **NLTK** | fw presentatio thi ha revis ... plea see =20 =09fw present present tuesday novemb 20th 3:30 - 5:30 pm t= `` congest transmiss system eastern interconnect '' confer room determin monday 19th thank =09present tj pl invit trader mid-market originators.=20 present tuesday novemb 21st 3:30-5:30 pm o= n `` congest transmiss system eastern interconnect '' confer room determin monday thank Stephen |
| **GENSIM** | FW Presentatio This revised please below FW Presentation The presentation Tuesday November th pm t Congestion transmission Eastern Interconnect The conference room determined Monday th Thanks Presentation TJ Can pls invite trader mid market originator  There presentation Tuesday November st pm o n Congestion transmission Eastern  Interconnect The conference room determined Monday Thanks Stephen |
| **SPACY** | fw Presentatio \n\n revise ... \n\n = 20 \n\n\n = 09fw presentation \n\n presentation Tuesday November 20th 3:30 - 5:30 pm t= \n " congestion transmission system  Eastern \n interconnect " \n\n conference room determine Monday 19th \n\n thank \n\n = 09presentation \n\n TJ \n\n pls invite trader mid - market originators.=20 \n\n\n presentation Tuesday November 21st 3:30 - 5:30 pm o= \n n " congestion transmission system Eastern Interconnect " \n\n conference room determine Monday \n\n\n thank \n\n Stephen |

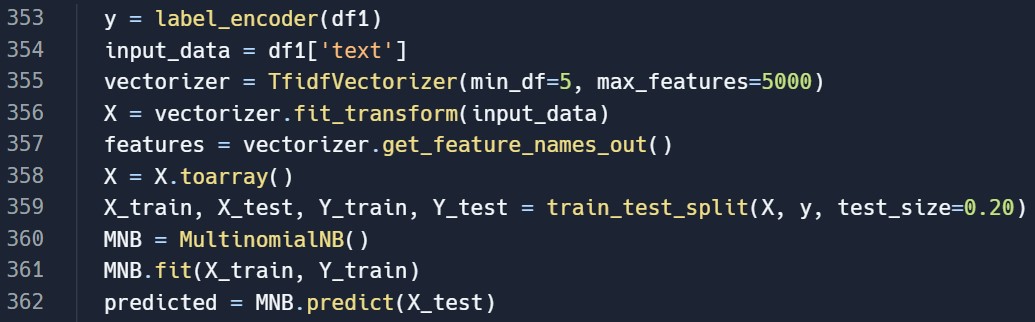
To understand how each pre-processing pipeline affects the text, an example of email content has been extracted after it has been put through each pipeline and is displayed in Table 3. Comparing these examples, NTLK and SpaCy do not remove numbers and some forms of punctuation, meaning these entities will be considered as tokens to train the classifier on. This could lead to low confidence in accuracy scores due to the test data matching on numbers and punctuation rather than the text itself. Gensim and SpaCy also do not convert all text to lowercase meaning that a classifier might consider the uppercase and lowercase version of a word as two different entities so some words may be missed because of the capitalisation of the word.

Another important distinction to be made is the lack of stemming involved in both the Gensim and SpaCy models compared to that of NLTK. The word ‘presentation’ is rooted to ‘present’ in NLTK but kept at full length in the other two models. The classifier may work better when the root of the word is considered compared to the variant, for example, the root word ‘present’ will match to ‘presenting’, ‘presenter’ or ‘presentation’, but ‘presentation’ will only match to

‘presentation’, leading to a lower model accuracy. It should be noted that the limitations of each library discussed in this section, are characteristics of the NLP pipelines and are findings of the report, not limitations of the experiment itself.

## Model Evaluation

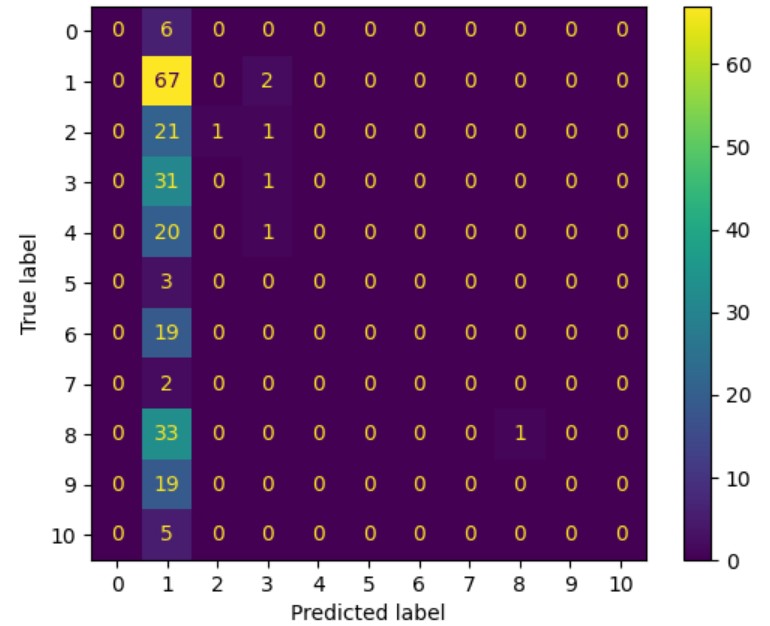
After pre-processing the data using NLP techniques, it must then be put through a vectorizer and split into training and test data before it is trained using the Naïve Bayes classifier; Figure 10 shows an example of the code needed for one model to carry out these processes.



*Figure 10 - Code snippet showing vectorization, feature extraction and model training*

A TF-IDF vectorizer is created on line 355 with the text being transformed to weighted values on line 356. The 10 words with the highest weighting are then extracted to show which words are thought most important by each model; the output from this will be discussed later in the report. Once weightings have been calculated, input data can be split into training and test data at an 80:20 ratio, shown on line 359. A Multinomial Naïve Bayes (MNB) is then fitted on training data and evaluated against test data. Multinomial classification has been used as the standard classifier in this experiment due to the previous studies analysed in literature where MNB was used most consistently.

### NLTK

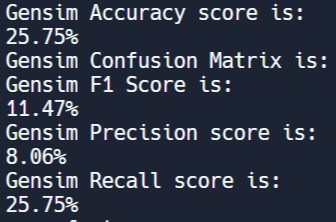
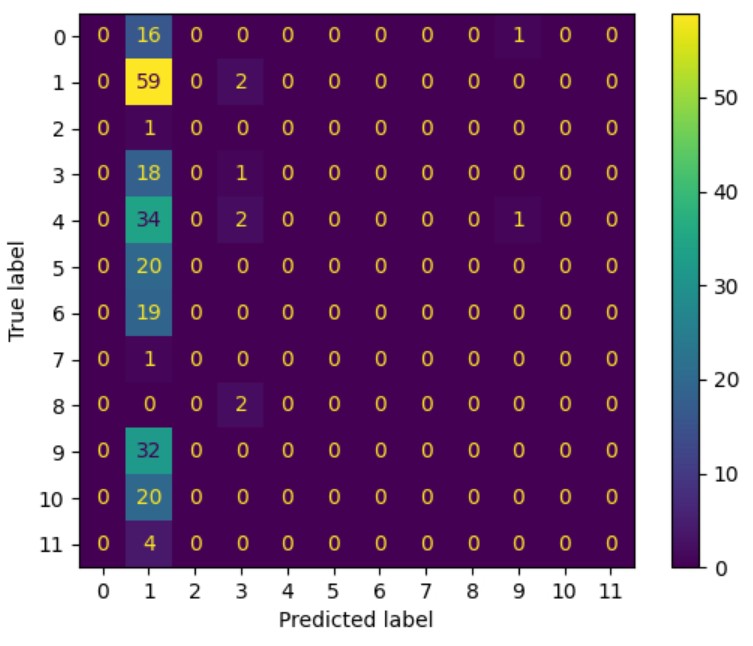


*Figure 12 - NLTK Model Performance Metrics*

*Figure 11 - NLTK Confusion Matrix*

The first model to be trained used NLTK NLP processes and the results of this model training can be seen by the confusion matrix in Figure 11 and the performance metrics displayed in Figure 12. The confusion matrix shows an uneven distribution with the model predicting most values as class 1, only 67 of these being correct, however, most model predictions do not match the true classes of the text, showing a poor model performance. Looking at the metrics, the NLTK model achieved a low accuracy percentage of only 30.04%, which is expected when compared with the matrix. This model has a precision score of 35.99% suggesting it isn’t efficient at predicting low false positives and high true positives; this is reflected by a similar recall score of 30% (which displays the proportion of emails that were classified correctly overall). A low recall score of 15% suggests this model is not effective at classifying emails, when an acceptable score would be close to 100% for best model performance.

### Gensim

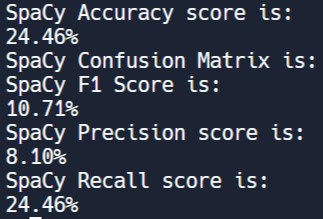
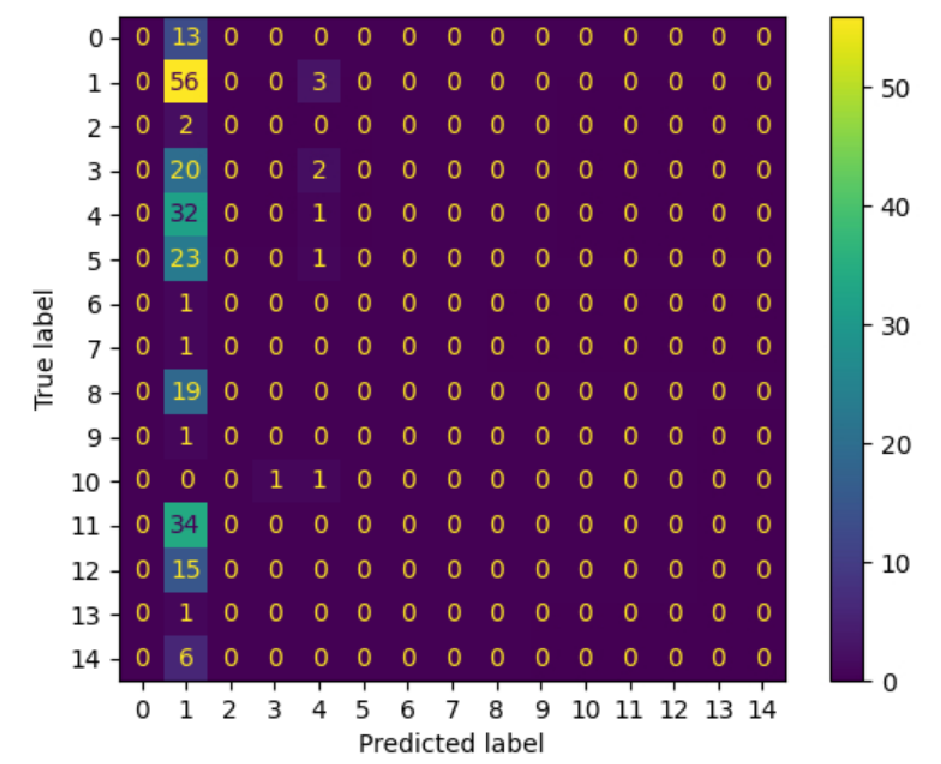


*Figure 14 - Gensim Model Performance Metrics*

*Figure 13 - Gensim Confusion Matrix*

Figure 13 shows the confusion matrix that is produced as a result of Gensim model training. Showing a similar pattern to Figure 11, it can be deduced that this model performs similarly to that of NLTK, predicting most values as class 1, but only 59 of these being correct. The majority were predicted as class 1 but again, not for the correct true labels. Figure 14 displays the Gensim performance metrics which show only a quarter of all predications being true. With 75% of prediction being incorrectly classed, it can be determined that the Gensim model, as is, performs unsuccessfully and cannot be relied upon to classify this data. A very low recall score is married with a lesser precision score, indicating that the Gensim model predicts high false positives and very few true positives, leading to minimal model performance.

### SpaCy



*Figure 16 - SpaCy Model Performance Metrics*

*Figure 15 - SpaCy Confusion Matrix*

The results from the SpaCy model are displayed in a confusion matrix in Figure 15 and as metrics in Figure 16. The matrix shows an uneven distribution, similar to that of NLTK and Gensim, where most class 1 labels are predicted incorrectly leading to a low model accuracy and recall score of only 24%. This is in conjunction with a low F1 score indicating a reduced model performance, highlighted by the precision score of 8%. It should also be noted that the SpaCy model has created the greatest number of classes to fit the data into, therefore increasing the chance of misrepresentation because there are more options to fit data to.

# DISCUSSION

This element of the report aims to discuss the statistical results of the experiment in further depth and critically compare the performance of each model to achieve the project aim. A comparison of findings to reviewed literature will highlight similar theories and differing opinions. There will also be a discussion surrounding the selection of words that are deemed important according to each model and what this means per the project aim.

## Performance Metrics

The performance metrics gathered in the experiment have been summarised in Table 4 for a direct comparison and an easy visualisation.

*Table 4 - Summary of model performance metrics for all models*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1 Score (%)** |
| **1 – NLTK** | 30.04 | 35.99 | 30.04 | 15.85 |
| **2 – Gensim** | 25.75 | 8.06 | 25.75 | 11.47 |
| **3 – SpaCy** | 24.46 | 8.10 | 24.46 | 10.71 |

In terms of model accuracy, NLTK performs at a higher rate, although only a 5% difference to the other libraries, it is still an increase in accuracy. Subsequently, it can be inferred that the NLTK pipeline does offer an advantage when classifying the Enron dataset in terms of model accuracy. SpaCy and Gensim perform at similar levels with only a 1.29% difference. This indicates that the pre-processing pipelines offer the same results and neither one offers an advantage over the other.

Recall is a determination of how many times the model was able to predict a certain class so when this metric is recorded in a multiclass model, this score will be close if not the same as the accuracy score as they will both be measuring the same output in terms of correctly predicted classes. Because each model predicts multiple classes, the recall score does not provide alternative metrics so does not offer any help with analysing these results.

When comparing precision scores, which determines a model’s ability to predict low false positives and high true positives, it can be determined that NLTK far outperforms the other two libraries with a difference of 27.9%. This shows that the NLTK pipeline is far better at predicting true positives therefore increasing the reliability on these results. With only 8% as a precision score, Spacy and Gensim pipelines cannot be relied upon to predict a high number of true positives, implying these models predict high numbers of false positives. Because both Spacy and Gensim have a similar precision scores, it can be assumed that both models perform similarly and again, neither offers an advantage in terms of model precision.

The F1 score gives a weighted balance of precision and recall considering false negatives and false positives as a higher priority than what accuracy will consider. F1 scores are more reliable for overall model performance when there is an uneven class distribution, which for this experiment is true. There may be more emails in one class than another therefore the F1 score is a more representative indicator of performance based on all result types. The results in Table 4 show the F1 percentages follow a similar pattern to the accuracy scores, with NLTK scoring highest and SpaCy and Gensim scoring similarly at a lower rate.

Each F1 score comes out at an average of half of the corresponding accuracy percentage which indicates that model performance overall, although accurate in predicting true positives and true negatives, is ineffective at predicting false negatives and false positives. Although the F1 scores are not high in terms of general model performance, the results do give an indication of hierarchy when comparing the three libraries and their relevant NLP pipelines.

The results of this experiment determined that NLTK was the best performing library, however this contradicts many of the studies reviewed in literature with both Al Omran & Treude (2017) and Smelyakov et al. (2020) finding that SpaCy performed better than others at text classification problems. Gensim was not found to be the best performing library in any reviewed study and this theory was reflected in the results of this experiment; a probable reason being that because Gensim is vector-based, it works better with the Word2Vec algorithm which is theorised by Al-Amrani et al. (2021). Miao et al. (2021) agreed with the outcome of this report, also finding that NLTK outperformed other libraries, but only by small margins.

### Pre-Processing

When looking back to the pre-processing pipelines in Figures 7, 8 and 9, NLTK was the only library to performing stemming as an official step in its pipeline, which would suggest that stemming words to their root increases the model’s chance at classifying an email correctly, if it can match more text based on root tokens. Although SpaCy performed more pre-processing steps by including POS and NER, this did not appear to have an impact on model performance as SpaCy performed worst out of the three but had similar results to Gensim (which had a very simple pre-processing pipeline that did not include POS and NER).

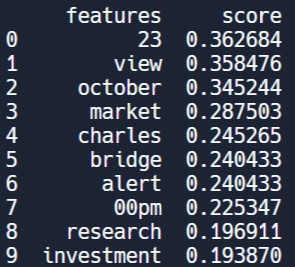
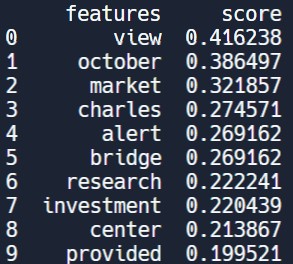
Hasan et al. (2021) and Bountakas et al. (2021) both concluded that a three step process of normalisation, tokenisation and lemmatization is just as effective to clean data for text classification. The pipeline closest to this theory was Gensim (Figure 8) and this report found Gensim to be the least accurate model therefore disagreeing with both authors. This report also found that introducing extra steps such as POS and NER in the SpaCy pipeline (Figure 9) were detrimental to the model accuracy as seen by the model metrics. This is disputed by Sun et al. (2017) who found that by adding extra pre-processing allowed for cleaner and more filtered text for their model; although Sun et al. (2017) assessed data for opinion mining, their pre-processing techniques would still pull out the same features as those needed for text classification, so their study is comparable to this report in terms of pre-processing outcomes. Other Variables

It must be noted that this experiment relies upon many variables, not just NLP techniques, but vectorizers and machine learning classifiers. From literature, it was deduced that TF-IDF was proven to be the preferred vectorizer with Cahyani et al. (2021) and Kasri et al.(2019) providing evidential studies to support this theory; for this reason, TF-IDF was used but the experiment could be tested with alternative vectorizers to check if model performance is impacted by this variable.

It was clear from research carried out by Ma et al. (2020) and Miao et al. (2021) that Naïve Bayes classifiers have an easy implementation and that such models using Naïve Bayes were fast and required little computation power. For this reason and considering the limited resources available for this project, Multinomial Naïve Bayes was chosen as the model classifier.

Literature showed a varied recommendation of classifiers, suggesting different classifiers work differently for differing datasets and types. Pranckevičius & Marcinkevičius (2017) and Tusar & Islam (2021) both found LR classifiers proved most accurate for text classification models whereas Smitha & Bharath (2020) found SVM algorithms to be most effective. Contradicting these studies, Miao et al. (2021) found that RF and Naïve Bayes performed similarly, with RF leading by 1.79% in accuracy. Because of this variance, the models could be re-trained using different classifiers to check the impact on model performance and ensure that this variable is set at the best possible option for Enron.

## Determination of Importance



*Figure 18 - Gensim Top 10*

*Figure 17 - NLTK Top 10 Figure 19 - SpaCy Top 10*

*Weighted Features*

*Weighted Features Weighted Features*

The main goal for finding the best performing library is to determine the importance of emails within the Enron dataset ; for this reason, a comparison of the top ten features chosen by each model’s vectorizer must be analysed. Figures 17, 18 and 19 display each library’s most important features with a corresponding weighting; the higher the score, the more important the model will consider the word.

NLTK (Figure 17) has given significant weighting to a number which is feasible considering the pre-processing example shown in Table 3 where numbers were not removed as part of the NLTK pipeline. When comparing the NLTK weightings to Spacy, it can also be seen that the same number has been placed at the top of the list in Figure 19, but with a higher score indicating Spacy has considered ‘23’ to be of more importance than NLTK. The words, ‘view’, ‘market’ and variations of ‘October’ have all been placed at the top of each list with varying weightings. This could suggest many emails consist of content surrounding the stock market which would be deemed important content; however, the word ‘view’ is very vague and does not hint at a particular theme that could suggest emails of interest, therefore this would be an inaccurate weighting and would decrease the validity of results.

NLTK and SpaCy have given similar weighting to the token ‘00pm’ which could indicate the end of a meeting time which would not be considered important and would reduce the validity of the results, if times are considered tokens. On the other hand, both Gensim and SpaCy have given weighting to the word ‘investment’ which would hint at important emails to be analysed, therefore increasing the validity of these models’ performances.

Overall, these lists show that each model differs by giving different weightings to different tokens, but similar tokens in each list suggest they perform similarly which is reflected in the results shown in Table 4. With the accuracy percentages in a similar range, it can be expected that the weighted feature lists would contain resemblances.

## Summary of Findings

Summarising the analysis above, it can be concluded that NLTK is the best performing model, producing the highest scoring metrics and far outperforming the other two libraries in terms of precision. Literature disagreed with these findings, suggesting instead, SpaCy, to be the best performing library, with only one reviewed study finding NLTK to be the best solution. Each pre-processing pipeline produced different results and it was found that adding more steps to a pipeline does not improve model accuracy overall. When looking at determination of importance, each model picked similar features with similar weightings with each list consisting of range of tokens that would and would not be considered important, therefore reasoning that neither library possesses an advantage over the other.

# CONCLUSION

The closing section will conclude the findings of the report with a project summary focussing on the extent at which the project aim and objectives have been met. There will also be a discussion surrounding project limitations and how these could be developed into future work, should the project be taken forward. Finally, a self-appraisal where the author will reflect on the development of the project.

## Project Summary

The aim of this project was to critically compare NLP libraries to identify the most effective analysis method for email content using the Enron dataset. This was achieved by conducting secondary research, developing a methodology, implementing such methods, and critically comparing models based on performance metrics and data output. Firstly, an in-depth literature review was conducted to provide necessary NLP theory as well as existing studies comparing NLP techniques and processes surrounding big data analysis. Literature determined a series of techniques to be used to develop training models and established three libraries to compare, giving three models for experimentation; these were NLTK, SpaCy and Gensim. A detailed section outlined the implementation of such models along with results gathered throughout.

Evaluative methods included gathering performance metrics and confusion matrices to give statistical overview of model performance where a discussion critically compared each model using metrics and pre-processing pipelines for each library. This comparison also included a discussion of extracted features to review the basis on which a model was determining text classification and whether this revealed the importance of an email. All project objectives were achieved and documented in this report, producing successful deliverables. This report conclusively finds that NLTK was the best performing library by achieving the highest performance metrics therefore showing the project aim has been achieved.

## Bias and Error

Previously discussed in the methodology section, every effort was made to reduce bias and error and remain legally compliant when handling sensitive data. To remain neutral, email content was not analysed pre-experimentation and was randomly sampled using computational techniques to include a wide variety of emails. When conducting secondary research, only works published post-2016 were considered when researching NLP techniques and studies to understand the most current theories. Publications were chosen from a wide range of peer-reviewed sources using a wide range of mediums. As statistical data was gathered for model evaluation, there was minimal need for qualitative data leading to opinion based results.

## Future Work and Development

As noted throughout this report, the process of training an NLP model relies on multiple variables: pre-processing, vectorization, feature extraction and machine learning classification. Although this project has the sole focus of comparing library performance, much of the experimentation can be manipulated to improve model performance, such as changing the vectorizer or using different classifiers. A comparison of different vectorizers and classifiers was conducted in the literature review to find the best suited choice for this experiment given the limited time and resources available.

If this project were to be taken forward, the models could be experimented with different vectorizers and classifiers to assess the impact this could have on model performance. Many of the classifiers discussed previously involved a greater implementation with higher computational need, therefore this would require extra time and research, hence why a simple Naïve Bayes classifier was chosen for this experiment. If other variables are adjusted for optimal training, then a fully comprehensive analytic could be developed to achieve much higher accuracy scores nearing 100%.

## Self-appraisal

Due to the achievement of the aim and successful completion of each objective, the project in its entirety was successful. Planning a time management strategy early on using a Gantt chart, ensured the project stayed on track and each stage was given a suitable time allocation. Secondary research, although a lengthy process, covered all the necessary theory with the author building on their own knowledge and learning throughout. Given the time constraints, an appropriate amount of research was conducted that led to a detailed methodology being developed. One strength of this report is the clear goals of each method with a step by step process that allowed for a smooth implementation of code. Had the methodology not been as detailed, it would have taken much longer and required deeper research to develop a successful working script. The smooth transition between chapters shows the well thought out flow of the project which led to very few setbacks.

To ensure I stayed within time and resource constraints, I opted to choose only three models to implement on only a sample of the initial dataset. To improve the validity and reliability of the experiment, the sample size could be increased, the more data to train models on, the higher the chance of correct text classification. In terms of evaluative techniques, all planned results were gathered and led to an in-depth critical analysis and comparison leading to a definitive conclusion; showing the quality of results can be considered a key strength of this project.

Where the project weakens is the consideration of other variables. Discussed previously, the option of experimenting with different vectorizers and classifiers would greatly improve the model accuracy, therefore, if the project were to be conducted again, these variables would be incorporated as part of the study, not only comparing libraries, but vectorizers and classifiers in the same experiment. This would take much longer and be a much larger experiment, but the results would have profound impact in producing a much more accurate analytic for data analysis. Another alternative approach would be to use a neural network (which was discussed in the literature review) instead of a classifier and use the Keras Python library to tailor make a neural network for the specifics of the Enron dataset.

Over the time this project has been carried out, the author has managed to effectively manage their time, completing this project amongst other commitments. Being the biggest project undertaken by the author, the project has allowed the author to develop their research and report writing skills as well as their application of knowledge when implementing code into a field of study related to the author’s interests. Upon completion of the experiment itself, it was noted by the author how low model performance was in general and so, the author feels that other variables would have had an impact on model training. As previously mentioned, experimenting with vectorizers and classifiers means the author could develop the project and improve model performance creating as best an analytic as possible.

Since the project proposal in August 2022, the publication of ChatGPT in 2023 changed the current view on artificial intelligence and has shown a significant development in the field, being labelled the “finest AI chatbot ever” (Haleem et al., 2022). ChatGPT, a generative pretrained transformer, is a machine learning tool which has been trained through supervised learning from human specialists on a substantial amount of data that “can respond to practically any question” (Haleem et al., 2022). On reflection, this new software would have been a vital part of the secondary research stage with its model training theory having considerable influence on the experiment methodology.

# REFERENCES

Al Omran, F. N. A., & Treude, C. (2017). Choosing an NLP Library for Analyzing Software

Documentation: A Systematic Literature Review and a Series of Experiments. *2017*

*IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*,

187–197. https://doi.org/10.1109/MSR.2017.42

Al-Amrani, Y., Zahidi, Y., & El Younoussi, Y. (2021). Different valuable tools for Arabic sentiment analysis: A comparative evaluation. *International Journal of Electrical and*

*Computer Engineering (IJECE)*, *11*(1), 753. https://doi.org/10.11591/ijece.v11i1.pp753-762

Al-Sai, Z. A., Abdullah, R., & husin, M. heikal. (2019). Big Data Impacts and Challenges: A Review. *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 150–155.

https://doi.org/10.1109/JEEIT.2019.8717484

Bountakas, P., Koutroumpouchos, K., & Xenakis, C. (2021). A Comparison of Natural

Language Processing and Machine Learning Methods for Phishing Email Detection.

*The 16th International Conference on Availability, Reliability and Security*, 1–12. https://doi.org/10.1145/3465481.3469205

Cahyani, D. E., & Patasik, I. (2021). Performance comparison of TF-IDF and Word2Vec models for emotion text classification. *Bulletin of Electrical Engineering and*

*Informatics*, *10*(5), 2780–2788. https://doi.org/10.11591/eei.v10i5.3157

Chowdhary, K. R. (2020). *Fundamentals of Artificial Intelligence*. Springer India.

https://doi.org/10.1007/978-81-322-3972-7

Haleem, A., Javaid, M., & Singh, R. P. (2022). An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges. *BenchCouncil*

*Transactions on Benchmarks, Standards and Evaluations*, *2*(4), 100089.

https://doi.org/10.1016/j.tbench.2023.100089

Hariri, R. H., Fredericks, E. M., & Bowers, K. M. (2019). Uncertainty in big data analytics:

Survey, opportunities, and challenges. *Journal of Big Data*, *6*(1), 44.

https://doi.org/10.1186/s40537-019-0206-3

Hasan, Md. M., Zaman, S. M., Talukdar, Md. A., Siddika, A., & Rabiul Alam, Md. G. (2021).

An Analysis of Machine Learning Algorithms and Deep Neural Networks for Email

Spam Classification using Natural Language Processing. *2021 IEEE International*

*Conference on Service Operations and Logistics, and Informatics (SOLI)*, 1–6.

https://doi.org/10.1109/SOLI54607.2021.9672398

Joseph, S. R., Hlomani, H., Letsholo, K., Kaniwa, F., & Sedimo, K. (2016). Natural Language

Processing: A Review. *Applied Sciences*, *6*(3), 10.

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining:*

*The ASA Data Science Journal*, *15*(4), 531–538. https://doi.org/10.1002/sam.11583

Kasri, M., Birjali, M., & Beni-Hssane, A. (2019). A comparison of features extraction methods for Arabic sentiment analysis. *Proceedings of the 4th International Conference on Big*

*Data and Internet of Things*, 1–6. https://doi.org/10.1145/3372938.3372998

Khurana, D. (2022). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, 32.

Kwon, O., Lee, N., & Shin, B. (2014). Data quality management, data usage experience and acquisition intention of big data analytics. *International Journal of Information*

*Management*, *34*(3), 387–394. https://doi.org/10.1016/j.ijinfomgt.2014.02.002

Landon-Murray, M. (2016). Big Data and Intelligence: Applications, Human Capital, and

Education. *Journal of Strategic Security*, *9*(2), 94–123. https://doi.org/10.5038/1944-

0472.9.2.1514

Lee, I. (2017). Big data: Dimensions, evolution, impacts, and challenges. *Business Horizons*,

*60*(3), 293–303. https://doi.org/10.1016/j.bushor.2017.01.004

Ma, T. M., Yamamori, K., & Thida, A. (2020). *A Comparative Approach to Naive Bayes*

*Classifier and Support Vector Machine for Email Spam Classification*.

Miao, Y., Jin, Z., Zhang, Y., Chen, Y., & Lai, J. (2021). Compare Machine Learning Models in Text Classification Using Steam User Reviews. *2021 3rd International Conference on Software Engineering and Development (ICSED)*, 40–45.

https://doi.org/10.1145/3507473.3507480

Mohanty, H., Bhuyan, P., & Chenthati, D. (Eds.). (2015). *Big Data: A Primer* (Vol. 11).

Springer India. https://doi.org/10.1007/978-81-322-2494-5

Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical

Applications. *2019 Amity International Conference on Artificial Intelligence (AICAI)*,

140–145. https://doi.org/10.1109/AICAI.2019.8701341

ONS, O. for N. S. (2021). *Internet users, UK: 2020*. 3.

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification. *Baltic Journal of Modern Computing*, *5*(2), 221–232. https://doi.org/10.22364

Sandhu, A. K. (2022). Big data with cloud computing: Discussions and challenges. *Big Data*

*Mining and Analytics*, *5*(1), 32–40. https://doi.org/10.26599/BDMA.2021.9020016

Sarkar, D. (2019). *Text Analytics with Python: A Practitioner’s Guide to Natural Language*

*Processing*. Apress. https://doi.org/10.1007/978-1-4842-4354-1

Silaparasetty, N. (2020). *Machine Learning Concepts with Python and the Jupyter Notebook*

*Environment: Using Tensorflow 2.0*. Apress. https://doi.org/10.1007/978-1-4842-

5967-2

Smelyakov, K., Karachevtsev, D., Kulemza, D., Samoilenko, Y., Patlan, O., & Chupryna, A. (2020). *Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications*. 5.

Smitha, N., & Bharath, R. (2020). Performance Comparison of Machine Learning Classifiers for Fake News Detection. *2020 Second International Conference on Inventive*

*Research in Computing Applications (ICIRCA)*, 696–700.

https://doi.org/10.1109/ICIRCA48905.2020.9183072

Sun, S., Luo, C., & Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information Fusion*, *36*, 10–25.

https://doi.org/10.1016/j.inffus.2016.10.004

Tusar, T. H. K., & Islam, T. (2021). *A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y.,

Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł.,

Gouws, S., Kato, Y., Kudo, T., Kazawa, H., … Dean, J. (2016). *Google’s Neural*

*Machine Translation System: Bridging the Gap between Human and Machine*

*Translation* (arXiv:1609.08144). arXiv. http://arxiv.org/abs/1609.08144

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). *Comparative Study of CNN and RNN for*

*Natural Language Processing* (arXiv:1702.01923). arXiv.

http://arxiv.org/abs/1702.01923

Zahidi, Y., Azroumahli, C., & El Younoussi, Y. (2019). Comparative Study of the Most Useful Arabic-supporting Natural Language Processing and Deep Learning Libraries. *2019*

*5th International Conference on Optimization and Applications (ICOA)*, 1–10.

https://doi.org/10.1109/ICOA.2019.8727617

Zainab, M., Usmani, A. R., Mehrban, S., & Hussain, M. (2019). FPGA Based

Implementations of RNN and CNN: A Brief Analysis. *2019 International Conference on Innovative Computing (ICIC)*, 1–8.

https://doi.org/10.1109/ICIC48496.2019.8966676