AVE Trends in Intelligent Computing Research

# Sentiment Analysis and Review Summarisation Using Machine Learning and Transformer Models: A Comparative Study

**Prem Kumar Mehta[1,*], RejwanBin Sulaiman[2]**

[1,2]Department of Computer Science and Technology, University of the West of Scotland, Paisley, Scotland, United Kingdom. b01794742@studentmail.uws.ac.uk[1], rejwan.binsulaiman@gmail.com[2]

*Corresponding author

**Abstract:** This study examines how machine learning, deep learning, and transformer-based techniques can be applied to classify sentiment in customer reviews and generate short, abstractive summaries from them. The research uses a cleaned dataset of genuine customer feedback, including both brief comments and longer narrative entries. Classical machine-learning models were first applied to create a baseline, followed by neural-network models, and finally transformer architectures, such as BERT for sentiment analysis and PEGASUS for summarisation. All models were trained and tested on the same processed dataset to maintain fairness in comparison. The results show that classical methods handle simple, straightforward reviews well, while deep learning models handle longer, more descriptive text more effectively. Across both tasks, the transformer models produced the strongest and most consistent performance. BERT achieved the highest accuracy in sentiment classification, and PEGASUS generated clear summaries that captured the main ideas of the reviews. Taken together, the findings suggest that transformer-based models are better suited to analyzing informal customer feedback that varies in structure and often carries mixed or complex emotions.

**Keywords:** Deep Learning (DL); Transformer Models; Text Summarisation; Customer Reviews; Bert and Pegasus; Sentiment Analysis; Abstractive Summarisation; Complex Emotions.

**Cite as:** P. K. Mehta and R. Sulaiman, "Sentiment Analysis and Review Summarisation Using Machine Learning and Transformer Models: A Comparative Study," *AVE Trends in Intelligent Computing Research*, vol. 1, no. 1, pp. 35–64, 2026.

**Journal Homepage:** https://avepubs.com/user/journals/details/ATICR

**Received on:** 28/07/2025, **Revised on:** 19/09/2025, **Accepted on:** 07/10/2025, **Published on:** 05/03/2026

**DOI:** https://doi.org/10.64091/ATICR.2026.000309

## 1. Introduction

In today's data-driven world, online customer reviews play a crucial role in influencing consumer decisions and shaping product perception. E-commerce platforms such as Amazon, Yelp, and TripAdvisor collect millions of textual reviews daily, forming a vast repository of consumer sentiment and product feedback [5]. While these reviews are valuable sources of information, their sheer volume presents significant challenges. Consumers often struggle to extract useful insights from repetitive, verbose, and unstructured text. Businesses likewise struggle to manually process large-scale feedback to guide product development or marketing strategies [28]. This situation has spurred research in Natural Language Processing (NLP), a subfield of Artificial Intelligence (AI), which focuses on enabling machines to understand and interpret human language. Within NLP, two important applications are sentiment analysis, which identifies emotional polarity (positive, negative, or neutral), and text summarisation, which generates concise representations of lengthy textual data. Despite extensive research in these individual areas, there is

limited exploration of combining the two tasks into a single automated system that can both summarise and classify sentiments efficiently [24]. The proposed paper seeks to fill this gap by developing a unified NLP-based framework that performs both review summarisation and sentiment classification simultaneously. The paper will employ a comparative approach using Machine Learning (ML) and Deep Learning (DL) methods to determine which offers superior accuracy, interpretability, and scalability for this dual task [7].

### 1.1. Research Aim and Question

The primary aim of this research is to design, implement, and evaluate an automated system capable of summarising customer reviews and classifying their sentiments through the combined application of NLP and Deep Learning techniques.

### 1.2. Objectives

To achieve the aim and address the research question, the paper will pursue the following objectives:

- Conduct a comprehensive literature review and explore publicly available review datasets.
- Preprocess the text data by cleaning, tokenizing, removing stop words, and lemmatizing.
- Train and evaluate baseline ML models, including Logistic Regression, Naïve Bayes, and Random Forest classifiers.
- Develop and fine-tune DL architectures, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTMs), and transformer-based models (BERT, PEGASUS).
- Evaluate model performance using metrics including precision, recall, F1-score (for classification), and ROUGE/BLEU (for summarisation).
- Compare ML and DL models in terms of accuracy, scalability, speed, and interpretability.
- Develop a functional prototype capable of summarising and classifying new reviews.
- Document findings and provide a comparative analysis of all methodologies.

### 1.3. Scope of the Study

The paper focuses exclusively on publicly available product reviews. It will minimize ethical issues by avoiding the use of private company data or human volunteers [9]. A comparison of several techniques and architectures, ranging from transformer-based DL models to conventional ML, will be part of the study. However, only evaluations written in English will be examined owing to time and computer limitations.

### 1.4. Research Motivation and Context

The growth of digital marketplaces has led to an enormous increase in customer-written reviews. People now depend heavily on these online comments to judge products and services, but the volume of text available is often far more than anyone can read. As a result, a clear gap exists between the vast amount of helpful information available online and the ability to quickly comprehend it. Businesses also face this problem, as they must examine thousands of reviews across various platforms to glean valuable insights. These issues are addressed by automated techniques, such as sentiment analysis and text summarisation, which minimize manual labour and enable quick interpretation of customer opinions [15]. The growing adoption of NLP tools in industry underscores the importance of this research. As companies turn to AI-supported decision-making, they need reliable systems to analyze and summarize customer feedback at scale. For this reason, the present study aligns well with industry needs for automated, accurate review analysis [27].

### 1.5. Significance of the Study

This research is essential in both academic and practical settings. From an educational perspective, it adds value by examining machine learning, deep learning, and transformer-based models within a single integrated pipeline for sentiment analysis and text summarisation, an approach rarely studied in combination. From a practical point of view, it demonstrates how automated systems can help businesses enhance customer satisfaction, track product-related opinions, and identify new trends in user behavior. The results can also assist online platforms by providing clear summaries of extensive review collections, reducing the burden of excessive information, and enabling faster, more effective decision-making [6].

### 1.6. Research Contributions

The key contributions of this study can be outlined as follows:

- First, it presents a complete end-to-end NLP workflow that includes data preprocessing, sentiment classification, and abstractive text summarisation.
- Second, it provides a comparative analysis of several models, such as Logistic Regression, Naive Bayes, Random Forest, CNN, LSTM, DistilBERT, and PEGASUS, using the same dataset.
- Third, it examines how zero-shot transformer-based summarisation performs when applied to customer review texts.
- Fourth, it offers insights into which model groups perform best for sentiment prediction and which require task-specific fine-tuning.
- Ultimately, the study presents a functional system that generates summaries and sentiment outputs for new review inputs.

## 2. Literature Review

### 2.1. Overview of Existing Literature

Over the past few decades, the field of automatic text summarisation has advanced quickly due to the expansion of digital data and the growing demand for effective text processing [21]. This overview of the literature examines the development, approaches, and challenges of text summarisation, with a focus on machine learning (ML) and natural language processing (NLP). The study provides a foundation for understanding key trends, identifying limitations, and guiding the development of an efficient automated review summarisation system by examining extractive, abstractive, and hybrid approaches, as well as current evaluation techniques and datasets.

#### 2.1.1. Key Studies Reviewed

Mridha et al. [20] discuss the entire progress and challenges of automatic text summarisation, ranging from traditional extractive methods to more recent deep learning-based abstractive methods. They claim that conventional extractive methods primarily rely on simple statistics, such as term frequency, sentence position, and certain cue words, to identify key sentences. These descriptions, however, are frequently succinct and can lack consistency or a natural flow of context. Then emerged deep learning models such as RNNs and LSTMs, which improved machines' comprehension of context and sequence in large text collections. However, transformers like BERT, GPT-2, and PEGASUS, which use attention mechanisms to analyze long-term text relationships and produce summaries that are notably more readable and human-like, were the ones that truly made a difference. The authors also mention that these models perform exceptionally well on datasets such as CNN/Daily Mail, Gigaword, and Newsroom, producing summaries that are both informative and fluent. Still, there are problems, such as the need for extensive computing power, insufficient domain-specific data at times, factual errors in summaries, and the difficulty of assessing quality using metrics like ROUGE or BLEU. Overall, this paper demonstrates that text summarisation research is shifting toward combining semantic understanding, context reasoning, and transformer architectures, which is particularly useful for automated review summarisation. These ideas provide a solid foundation for creating models that can effectively handle user reviews, ensuring summaries are clear, relevant, and logical for informed decision-making or analysis.

Hemalatha et al. [11] begin by stating that automated textbook summarisation has become increasingly important due to the vast amount of digital data available today, which makes it challenging for people to read and extract key information quickly. Consequently, NLP, as a component of AI, proves very useful in addressing this issue. Abstractive summarisation, on the other hand, attempts to generate new sentences like a human would, utilizing neural networks and deep learning models, including seq2seq and attention mechanisms. However, it requires large, annotated datasets and substantial computational resources. Both extractive and abstractive methods have greatly improved with the advent of transformers and pre-trained language models, such as BERT and GPT. GPT is especially good at producing fluent summaries, while BERT is excellent at context-aware extraction. Some drawbacks of purely abstractive models have been mitigated by hybrid approaches that combine extractive filtering and abstractive generation to produce succinct yet logical summaries. Recent work has greatly improved these models and yielded encouraging results by fine-tuning them on datasets such as XSum, CNN/Daily Mail, and biological corpora. However, challenges remain, including handling long texts, domain-specific vocabulary, and complex sentence structures. All things considered, the literature demonstrates that combining modern NLP models with hybrid summarisation strategies can produce high-quality summaries; nevertheless, machines continue to struggle to understand human-like context, sarcasm, and contextual nuance, which suggests that more research and development in automated summarisation is needed.

Gowsic et al. [10] discuss how vast amounts of digital text surround people today. Every day, new blogs, papers, and reports are added online, and reading them all takes significant time and effort. They explain that this growing flow of information has created a need for systems that can condense long documents while clearly preserving the main ideas. The study describes text summarisation as part of Natural Language Processing, in which the goal is to condense a large piece of text into a shorter version without losing meaning. The authors note that recent advances in Machine Learning and deep learning, especially models such as Long Short-Term Memory (LSTM), have made it easier for machines to understand the flow and context of

text. Their paper combines two methods: extractive summarisation, which selects key lines from the original, and abstractive summarisation, which rewrites them in a natural and readable manner. They also utilize models such as BERT, BART, and TF-IDF to construct more effective, natural-sounding summaries. The system was evaluated using metrics such as ROUGE and BLEU to assess its quality and accuracy. To make it simple for anyone to use, the authors built a web interface with Gradio that lets users upload text, PDF, audio, or video files. Overall, their work demonstrates how AI can save time and help people read more effectively, not harder. Kumar et al. [14] begin by stating that companies nowadays need to understand what people think about their products and services. Still, there are so many reviews online every day that reading them all manually is just impossible and very time-consuming. That's why sentiment analysis has become increasingly important, as it leverages NLP and ML to quickly and cost-effectively determine people's emotions from text. In earlier studies, models such as Random Forests, Logistic Regression, and SVMs have been effective at classifying reviews as positive, negative, or neutral.

However, there are still problems, such as sarcasm, mixed feelings, or unclear words, which can confuse machines and lead to incorrect predictions. They also discuss how NLP steps like tokenization, stemming, and vectorization help make raw text readable for computers, but even then, it's not perfect. Sentiment analysis is not just for companies; it can also be used for social media analysis, checking public opinion on policies, or even for news purposes. In this paper, the authors used Amazon Alexa reviews to test different ML models and compare their accuracy at predicting customer sentiment. Overall, the literature shows that combining NLP and ML is a good way to deal with large amounts of text and get helpful info fast, even though machines still can’t fully understand human emotions and sometimes makes wrong decisions and this is why human oversight is still essential and also shows that more improvements in algorithms are needed to handle tricky things like irony or humor or mixed sentiments which is very common in online reviews and public opinion. Still, there is considerable room for research and development in this area. Patel and Tabrizi [4] highlight the development and significance of Automatic Text Summarisation (ATS). They clarify that ATS automatically condenses large amounts of material into summaries to address information overload. Abstractive and extractive are the two main categories into which ATS approaches are frequently separated. In extractive summarisation, key lines or phrases are selected from the source material, which can lead to grammatical errors. The coherence and human-likeness of the summary are enhanced by abstractive summarisation, which is more challenging to employ because it generates new sentences or phrases.

In addition, the study disambiguates between single-document and multi-document summarisation. It examines purpose-based summarisation, including generic, domain-specific, and query-oriented summarisation, as well as various input types. They discuss current research trends, noting that most studies focus on extractive methods due to their relative simplicity. Deep learning is gradually emphasizing reactive approaches through sequence-to-sequence models and attention mechanisms. They emphasize the use of datasets such as DUC and CNN for training and evaluation, with ROUGE metrics serving as the standard for assessing summary quality. Personalized ATS services range widely, from news headlines and student notes to medical records and personalized summaries for question-answering systems. All things considered, the study shows that ATS is still developing by combining NLP, ML, and AI approaches, offering a solid basis for further investigation in automated review summarisation. Because it is still difficult to generate accurate, logical, and context-relevant summaries, this is a major difficulty. Nagarhalli et al. [22] explain how computers' engagement with human language has changed dramatically through machine learning. Because language is never static or straightforward, human-made rules were the mainstay of NLP systems before all of this. These rules were slow and not always correct. When machine learning was first introduced, systems began learning from real-world examples rather than just following established rules. The author explores various learning methods, including supervised and unsupervised learning, and explains models such as Naïve Bayes, SVM, and neural networks, which are now commonly used in NLP. He discusses how these models are helpful for tasks like voice recognition, automatic translation, and even determining an individual's emotional state from their speech.

The study also examines how more modern deep learning models, such as RNNs and transformers, can better understand context and longer words than previous models. Still, Nagarhalli et al. [22] make it clear that there are significant problems, including the need for extensive training data, powerful computers, and the inherent bias in the data itself. Even with all that, he says that machine learning has made NLP far more capable and closer to how humans process language. In the end, the article offers a clear explanation of how technology has become much wiser and better suited to managing human interactions because of the transition from rule-based to data-driven systems. Pisote et al. [23] in their paper on sentiment analysis in Marathi, focus on how machine learning and natural language processing can be utilized to enhance the comprehension of regional-language content, particularly on social media. Their study builds on earlier research that has sought to address language diversity and the challenges of analyzing emotions in low-resource languages. Some researchers noted that India has more than 100 spoken languages, but only a few have adequate NLP tools, making it difficult to process local-language text. Other studies have focused on collecting, cleaning, and building datasets for languages such as Kannada, Bengali, Urdu, and Arabic to improve sentiment classification accuracy. Deep learning models, such as LSTMs, CNNs, and XLM-R, have been applied in various works to address problems such as mixed-language use, inconsistent grammar, and small datasets. Researchers have also developed sentiment lexicons and employed hybrid methods that combine lexicon-based and machine-learning models, achieving better accuracy than single-approach methods.

Works employing Lex Rank and Text Rank algorithms for text summarisation, in conjunction with rule-based stemming in Marathi and related regional languages, have shown that combining conventional techniques with contemporary algorithms improves outcomes. By creating a dataset centered on popular perceptions of Chhatrapati Shivaji Maharaj and implementing a correction mechanism to enhance sentiment identification, Deshpande et al.'s study helps address this. All things considered, the analysis of these studies shows a clear trend toward developing NLP systems for underrepresented languages, such as Marathi, that are more precise, adaptable, and culturally sensitive. Although considerable previous work has examined different approaches to text summarisation, sentiment analysis has become an equally important area within NLP, especially as online reviews have become more informal and inconsistent. Earlier studies using Logistic Regression, Naïve Bayes, and Random Forests often report good results when the data is clean and well-structured. Still, several papers note that these models tend to lose accuracy when the text becomes messy, emotional, or unstructured. Deep learning models, particularly LSTMs, were developed in part to overcome these weaknesses, and many researchers have shown that they capture sequential information more effectively. Even so, the literature also notes that their performance can fluctuate when the writing style changes from one review to another or when the dataset contains slang, abbreviations, and spelling variations. More recent research highlights the advantages of transformer-based sentiment models, especially BERT variants, which analyse text in both directions and retain context more reliably than earlier methods.

However, several studies also emphasize that the success of these models depends on having fine-tuning data that closely matches the target domain; otherwise, their performance may drop unexpectedly. By examining summarisation and sentiment classification together in a single workflow, this dissertation explores a relationship that most studies treat separately. This integrated view helps show how the two tasks influence each other and addresses a gap acknowledged by the existing literature but rarely examined in depth. Online communication has grown so rapidly that enormous volumes of text now require automated handling. Researchers initially used straightforward, rule-based methods to handle the growing volume of data as users began posting reviews, comments, and other personal viewpoints on various websites. These early summarisation techniques relied on techniques such as checking where a sentence appears in a paragraph or counting repeated words. Approaches like these were simple to implement and worked reasonably well on structured writing, but they did not understand the actual message being communicated. They treated text as separate, unrelated words rather than as something that carries meaning through its structure. Due to this limitation, the summaries produced often lacked nuance, especially as online writing shifted toward a more informal and emotional style. A similar pattern can be observed in the early stages of sentiment analysis. Initially, most systems relied on sentiment dictionaries, such as SentiWordNet, which labeled individual words as positive, negative, or neutral. This worked to a point, but it quickly became clear that language does not always behave literally. A phrase such as “not bad” is generally meant positively, yet early systems interpreted it word by word and often arrived at the opposite sentiment.

Challenges like this encouraged researchers to move towards machine learning methods that could learn how sentiment is expressed from real examples rather than relying on literal word meanings. Instead of relying on fixed word lists that never change, machine learning models learn how people express sentiment by studying labeled examples. This shift enabled the models to adapt to their context and handle more complex forms of expression. As machine learning advanced, deep learning methods became central tools for both summarisation and sentiment analysis. CNNs, for example, helped spot repeated phrase patterns, while RNNs and, primarily, LSTMs handled longer pieces of text where the sequence of words plays a crucial role. These models were able to track the flow of a sentence or an entire review far better than the earlier, more rigid techniques ever could. This made them better suited for the varied, messy writing styles found on platforms such as Amazon, TripAdvisor, and Yelp, where reviews can range from brief, emotional reactions to lengthy narratives. The field changed even more dramatically with the arrival of transformer-based models. Transformers use self-attention, which enables them to consider relationships across an entire piece of text simultaneously rather than reading it sequentially. This design has proved highly effective, and transformers now dominate many NLP tasks. Their performance improves further when they are trained on data specific to a particular domain. Because they are so flexible and accurate, researchers and industry teams are now exploring systems that combine summarisation and sentiment analysis into a single workflow, enabling organizations to gain a more precise, practical understanding of customer experiences.

### 2.1.2. Critical Analysis of Existing Literature

A closer examination of the existing literature reveals that, although many studies highlight the strengths of the methods they propose, only a small number take the additional step of investigating why specific techniques are effective in some situations and fail in others. Much of the published work on sentiment analysis and summarisation focuses on reporting performance gains without exploring the reasons behind model behavior. Extractive approaches, for instance, are often described as straightforward and reliable; however, several empirical studies have shown that these methods can repeat emotional or irrelevant sections of text, particularly in customer reviews. Abstractive models, on the other hand, are known for producing smoother summaries. Still, researchers have also noted factual errors and problems with domain transfer when these models are not trained on review-specific material. In sentiment analysis, classical machine learning models are commonly recognised

for their speed and simplicity. Still, many studies give limited attention to their inability to handle contextual shifts, sarcasm, or opinions spread across multiple sentences. Deep learning models, such as CNNs and LSTMs, are often presented as stronger alternatives, though their limitations are not always discussed. CNNs, for example, struggle with long-range dependencies, while LSTMs tend to lose reliability when the text is noisy, informal, or inconsistent. Even transformer-based models, which are widely considered state-of-the-art, do not always perform consistently across domains when used without fine-tuning, an issue that only a few papers mention in detail.

This lack of deeper comparison highlights the need for research that evaluates several model families using the same dataset, under the same conditions, and within a single, unified workflow. The present study was designed to address this gap directly. A common issue highlighted in the literature is that many studies test their models on tidy, curated datasets rather than on genuinely messy, real-world text. Customer reviews often contain spelling errors, shorthand, informal punctuation, and mixed tones; however, popular datasets like IMDb and SST-2, as well as other preprocessed corpora, remove most of this noise. As a result, the reported accuracy of many models may not reflect how they behave outside controlled environments. Only a small number of studies examine how models cope with unstructured or noisy input, leaving uncertainty about the reliability of these systems when used in practical settings. Another concern relates to problems that appear in the evaluation process itself. Class imbalance is common in sentiment datasets—positive reviews often dominate—yet many papers still report accuracy without showing how well the model handles less frequent categories such as negative or neutral sentiment. This can give a misleading impression of performance, especially for businesses that rely on detecting dissatisfaction. In addition, reproducibility remains a challenge, as studies often differ in their preprocessing choices, hyperparameters, or training setups; however, these details are not always clearly documented. The computational demands of deep models are rarely discussed, even though real-world deployments may lack access to high-end hardware. Together, these gaps make it challenging to compare findings across studies and raise questions about the extent to which academic results can be applied to real-world systems.

### 2.1.3. Critical Comparison of Extractive and Abstractive Summarisation

Extractive and abstractive summarisation have both received considerable attention in previous research; however, many papers describe extractive methods as straightforward and portray abstractive methods as the superior option. Upon closer examination of the literature, this assumption is not always valid. Extractive summaries usually remain factually correct because they rely on sentences taken directly from the original text. Even so, they often repeat unnecessary or highly emotional remarks, which is a common issue in customer reviews where people tend to express strong opinions or restate the same point. Abstractive models produce smoother, more natural-sounding summaries, but they also pose a different challenge. Since they rewrite information rather than copy it, they can produce lines that are not actually present in the source review. This becomes a problem in areas where accuracy cannot be compromised. These differences demonstrate that, although abstractive methods appear more advanced, they are only effective when they have strong training data and are appropriately adapted to the domain. More detailed research is needed to understand how both approaches perform on genuine customer reviews, particularly because many existing studies do not fully address these limitations. Summarisation methods do not perform equally across all document types, and the differences between extractive and abstractive techniques become clear when examining their practical use.

Extractive methods usually work well in fields where every word matters-legal writing, clinical documentation, or any setting where the original phrasing carries significant weight. Because this approach copies statements directly from the source, the meaning is preserved exactly as written. However, the same strength becomes a drawback when the text comes from customer review platforms. Reviews tend to be informal and often include emotional language, repetition, and conversational remarks. Abstractive methods were introduced partly to overcome these limitations, as they allow the system to rephrase information rather than lift it word-for-word. Yet this freedom brings its own set of concerns. Because the model constructs new sentences, it may drift away from the original meaning, especially if it has not been trained on data that reflects the vocabulary or style of a particular domain. This issue is even more noticeable in situations where precision is essential. The usefulness of an abstractive summary is closely tied to the data the model has been trained on. If the training data does not reflect the vocabulary or writing style typical of the target domain, the model may produce summaries that feel generic and lack meaningful detail, failing to capture the finer nuances of the original text. To balance both accuracy and readability, hybrid methods have been proposed-these systems first identify important parts of the source text using extractive techniques and then refine them using an abstractive layer. Although this combination appears promising and may help reduce computational load, research on how well hybrid approaches handle noisy, real-world customer reviews is still limited.

### 2.1.4. Critical Evaluation of Transformer-based Approaches

Although models such as BERT and PEGASUS are widely discussed in recent work, several studies have noted that their effectiveness is strongly influenced by the data used to train them. Many commonly used benchmark datasets are heavily cleaned and structured, meaning they do not reflect the informal, repetitive, and emotionally inconsistent writing typical of genuine customer reviews. Some researchers note that transformer-based summarisation models achieve strong results only

when fine-tuned on large amounts of domain-specific data, which is not always possible in smaller studies or specialized industries. Others report that pre-trained transformers can become overly tied to their training data, producing summaries that sound generic and fail to capture the specific issues customers raise. These observations challenge the notion that transformer models consistently outperform other methods and highlight the importance of evaluating a range of model architectures in more realistic and diverse settings [19]. Over the past few years, numerous transformer models have been introduced, each with slightly different strengths, tailored to specific tasks. Models such as BART and T5 are now widely used for summarisation because their encoder–decoder architecture enables them to understand the original text and generate a simplified version.

Although they perform well on standard benchmark datasets, several studies have noted that their accuracy can drop when the text originates from a particular domain or contains informal or messy writing. This problem was not always highlighted in early research; however, newer studies repeatedly show that some level of domain adaptation or additional training is typically required. Alongside these large models, lighter versions such as DistilBERT and ALBERT have been developed to reduce the computational load. They tend to work well for tasks such as classification, yet evidence for their ability to handle more demanding generative tasks remains limited. Another area that has not received enough attention concerns the bias that pre-trained transformer models may carry, as they are trained on vast collections of online text. Only a small number of studies have examined how such bias might influence sentiment predictions or the tone of summaries produced for different customer groups. A further challenge is the difficulty transformers face when working with very long pieces of text due to fixed token limits. Newer architectures, such as Longformer and BigBird, attempt to overcome this issue. However, there is still insufficient research demonstrating how well they handle genuine customer reviews, which vary significantly in length and content. For these reasons, selecting the most suitable transformer model for analysing customer feedback remains an active area of study.

### 2.1.5. Issues with Evaluation Metrics and Interpretation

The literature suggests that standard evaluation metrics, such as ROUGE and BLEU, do not provide a comprehensive picture of a generated summary's effectiveness. Although these metrics provide numeric scores based on shared phrases, many studies have pointed out that they do not account for clarity, flow, or how helpful the summary is to a real reader. Similar difficulties can be masked by accuracy alone in sentiment categorization, especially when the model struggles with smaller or less frequent classes. Strong results on well-balanced datasets do not always translate to real-world scenarios, where impartial assessments are often far fewer in number, several academics say. These limitations indicate that evaluation results should be interpreted with caution and supplemented with qualitative analysis to gain a more accurate understanding of how the models behave [5]. Recent studies have highlighted several weaknesses in using ROUGE and BLEU for evaluating abstractive summaries, particularly because these n-gram–based metrics often struggle to determine whether a generated summary accurately reflects the meaning of the original text.

Two summaries can communicate the same idea while sharing almost no overlapping words. Yet, ROUGE would still score them poorly because it focuses on surface-level token matches rather than deeper meaning. These limitations have encouraged the adoption of newer evaluation methods, such as BERTScore and BLEURT, which rely on contextual embeddings to compare summaries and thus provide a more meaningful indication of semantic similarity. Although these approaches represent an improvement, they are not without their own issues, as they rely on pre-trained language models and may inherit any biases or inaccuracies present within those models. For this reason, human judgment remains essential for evaluating qualities such as clarity, coherence, and overall usefulness. However, many studies rely solely on automated metrics because they are faster and cheaper, and this dependence can undermine the reliability of their conclusions, especially when evaluating summarisation systems intended for real-world applications.

### 2.1.6. Inconsistencies and Contradictions in Existing Research

A closer examination of the literature reveals that many of the findings do not align with one another. Some papers claim that CNN models outperform LSTMs for sentiment analysis, while others report the opposite, making it difficult to determine which approach is genuinely stronger. A similar disagreement also appears in summarisation research. Extractive methods are described in some studies as reliable because they preserve the original facts; however, other researchers argue that the summaries they produce often appear rough or disconnected.

Abstractive techniques usually read more smoothly, but they sometimes introduce details that are not present in the original text, especially when the model has not been trained for that specific domain. Transformer-based models also show mixed results. Several studies treat them as top performers, but others note that their accuracy can drop noticeably when they are applied to new datasets without fine-tuning. Taken together, these inconsistencies suggest that no single model performs well across all situations, and that outcomes depend strongly on the dataset and the task type. This highlights the need for studies that compare different models under the same conditions, which remains uncommon in the current literature.

### 2.1.7. Research Gaps Identified in Prior Studies

Across existing research, one issue that repeatedly arises is the lack of systems that perform sentiment classification and summarisation within a single workflow. Most papers study these tasks independently, which makes it difficult to understand how the two processes might complement each other when combined. Another noticeable gap is the limited number of studies that test different model families on the same dataset, particularly when comparing traditional machine learning methods with deep learning and transformer-based approaches. In addition, only a small amount of research examines how transformer models perform on domain-specific data when fine-tuned. These gaps form the basis for the current study, which aims to compare a wide range of models and assess their suitability for building an integrated system that analyzes and summarizes customer reviews in a single pipeline [16]. Although several recent studies have examined the strengths of individual transformer models, there is still limited research on how well these technologies function when combined into a complete review-analysis pipeline. Most existing systems treat sentiment classification and summarisation as separate components, resulting in duplicated preprocessing and preventing deeper insights when both sentiment and key themes are interpreted together. Another noticeable gap concerns multilingual and cross-cultural differences in customer reviews.

A large portion of current work is centered on English-language data, leaving reviews from low-resource languages mostly unexplored. As online markets become increasingly global, it is essential to examine how these models accommodate different languages and cultural styles of expression. This reviewed the primary research on sentiment analysis and review summarisation with a critical focus. It discussed how the field has developed, highlighted the strengths and limitations of the major modelling approaches, and showed that many studies reach conflicting conclusions. It also highlighted several areas that have not been sufficiently explored, especially the challenges posed by domain-specific data and the lack of frameworks that combine both tasks in a single system. When examining the existing research, NLP has made significant progress, but many of the same problems persist once these models are tested on real customer reviews. Models often struggle to remain consistent, understand the broader meaning of what is written, navigate different domains, and produce evaluations that can be trusted, suggesting there is still considerable room for improvement. These gaps underscore the importance of well-designed comparative studies that assess multiple modelling strategies under uniform conditions, particularly for developing integrated tools to support practical applications in review analysis.

## 3. Methodology

This presents the methodological framework adopted for this study. It explains the research approach, design, data collection strategy, data preparation techniques, model development procedures, evaluation methods, tools, and ethical considerations. It integrates the methodology developed in the interim report and extends it to provide a complete, academically rigorous foundation for the experimental analysis presented in the next section.

### 3.1. Research Approach

Using quantitative, experimental methodology, this study primarily aims to develop and evaluate several deep learning models for sentiment classification and text summarisation by analyzing textual data (product reviews). Using a data-driven pipeline, the research combines Exploratory Data Analysis (EDA), Natural Language Processing (NLP), and Transformer-based architectures. The approach involves collecting review data, preprocessing it, and experimenting with several deep learning techniques, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Transformer models such as BERT for classification and PEGASUS for summarisation. This approach is suitable because sentiment analysis and summarisation require systematic experimentation, repeatable evaluation, and quantitative measurement. A data-driven, model-focused approach enables the study to measure performance differences across multiple architectures objectively. This aligns directly with the research objectives, which aim to identify the most effective combination of models for analyzing customer reviews [17].

### 3.2. Research Design

A computational experimental design was adopted to systematically train, validate, and compare multiple models on the same dataset. The process includes the following stages:

- **Data Acquisition:** The study utilizes a publicly available dataset of consumer reviews to ensure reproducibility and transparency.
- **Data Pre-processing and Cleaning:** The raw dataset was cleaned by removing missing entries, filtering irrelevant records, and normalizing text content.
- **Feature Engineering and Transformation:** The textual data were tokenised, encoded, and vectorised into numerical representations suitable for model training.

- **Model Development:** Multiple architectures, including CNN, LSTM, BERT, and PEGASUS, were implemented and fine-tuned for classification and summarisation tasks.
- **Model Evaluation:** Each model was assessed using well-established evaluation metrics to measure predictive accuracy and generalization performance.

This design facilitates an iterative process of model refinement and comparative analysis to identify the most effective architecture for the intended tasks (Figure 1).

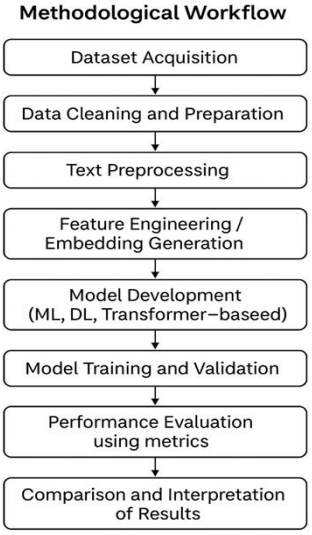


**Figure 1:** Methodology workflow

### 3.3. Data Collection and Preparation

The dataset was extracted from a publicly accessible GitHub repository, including reviews of consumer clothing. The review title, descriptive language, clothing class, and numerical ratings for material quality, construction, colour, finishing, and durability are among its qualities. The data was downloaded into CSV format and then imported into Microsoft Excel for analysis and preprocessing. During acquisition, it was observed that some rating fields contained binary values (0 or 1), indicating the presence or absence of specific product quality aspects. The following steps were performed:

#### 3.3.1. Data Cleaning

Only items with valid integer ratings were kept after missing values were eliminated from the Review and Cons_rating columns. To ensure accuracy and consistency, the dataset was filtered to exclude null and aberrant observations.

#### 3.3.2. Exploratory Data Analysis (EDA)

Initial statistical exploration was performed using Pandas, Matplotlib, and Seaborn. Visual analyses, including distribution plots, histograms of review lengths, and word frequency counts, were generated to understand the underlying data characteristics. EDA revealed distributions of sentiment and potential class imbalances across rating levels.

#### 3.3.3. Sentiment Categorisation

The numerical ratings were mapped to categorical sentiment labels to facilitate supervised learning:

- **Ratings (1-2):** Negative
- **Rating 3:** Neutral
- **Ratings (4-5):** Positive

These categories were subsequently encoded into numerical labels (negative = 0, neutral = 1, positive = 2) to enable compatibility with classification algorithms.

### 3.3.4. Text pre-processing

The Natural Language Toolkit (NLTK) was used to normalize the textual data through a series of transformations, including lowercase conversion, removal of punctuation, removal of non-alphanumeric characters, and removal of English stop words. This preserved semantically significant tokens while reducing superfluous linguistic noise.

### 3.3.5. Dataset Partitioning

The processed dataset was divided into subgroups for testing (about 15%), validation (approximately 12%), and training (approximately 73%) to encourage thorough model evaluation and avoid overfitting. Stratified sampling was employed to guarantee that sentiment classes were fairly represented across all subgroups. The dataset is well-suited for this research because it provides both written reviews and numerical ratings, which are useful for supervised learning. The reviews exhibit a diverse range of emotional expressions, offering suitable material for sentiment analysis. It also contains a large enough number of samples to train and evaluate deep learning and transformer models with confidence. Since the reviews come from real customers, the language is informal and sometimes messy, allowing the models to be tested in realistic conditions rather than clean, artificial text. Together, these features make the dataset a strong choice for fairly comparing different NLP models [26].

## 3.4. Model Development

The research employs three categories of models: Machine Learning, Deep Learning, and Transformer-based models, each designed to process textual data for sentiment classification and summarisation.

### 3.4.1. Classical Machine Learning Models

The classical models served as the baseline against which the later, more advanced approaches were compared. For these models, the review text was converted into numerical features using a TF–IDF vectoriser, configured to include up to 15,000 features and capture both single words and simple two-word combinations. The following models were trained:

- Logistic Regression (max_iter = 1000)
- Multinomial Naïve Bayes
- Random Forest (150 Estimators)

These models were trained on the TF–IDF vectors and evaluated using accuracy, precision, recall, and F1-score. They provide insight into how well traditional feature-based approaches handle noisy, user-generated review text [13].

### 3.4.2. Deep Learning Models

Two deep-learning models were implemented for this study using TensorFlow and Keras: A Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network [18]. Both models were trained under the same experimental conditions to allow a fair comparison of their behaviour:

- **Text Vectorisation:** The review text was first converted into integer sequences using a Keras Tokeniser. The configuration included a vocabulary limit of 20,000 words, a maximum sequence length of 120 tokens, post-padding, and an embedding dimension of 64. This approach preserved word order, allowing the models to learn patterns and relationships that naturally occur within the reviews.
- **CNN Architecture (as Implemented):** The CNN was designed to identify short-range patterns and frequently repeated expressions. Its architecture included a 64-dimensional embedding layer, followed by a 1D convolutional layer with 64 filters and a kernel size of 5. A global max-pooling layer was used to reduce the output, which was then passed through a dense layer with 32 units and ReLU activation. The final layer used SoftMax to predict the three sentiment classes. The model was trained for 3 epochs with a batch size of 64, a 10% validation split, the Adam optimiser, and categorical cross-entropy loss.
- **LSTM Architecture (as Implemented):** The LSTM model was designed to handle longer text sequences and capture dependencies that unfold across a review. Its structure consisted of a 64-dimensional embedding layer, an LSTM layer with 64 units, a dense layer with 32 units and ReLU activation, and a final SoftMax output layer for the sentiment classes. The training settings matched those used for the CNN: 3 epochs, batch size of 64, validation split of 0.1, Adam as the optimizer, and categorical cross-entropy as the loss function.

Both deep learning models were evaluated using test accuracy and standard classification metrics to assess how effectively they learned sentiment patterns from the dataset.

### 3.4.3. Transformer Models

Transformer-based models were implemented using the Hugging Face Transformers library in PyTorch. Two architectures were used in the study: DistilBERT for sentiment classification and PEGASUS for abstractive summarisation.

#### 3.4.3.1. DistilBERT for Sentiment Classification

The sentiment model was based on the pre-trained distilbert-base-uncased-finetuned-sst-2-english checkpoint. It was applied in inference-only mode, meaning no additional fine-tuning was performed on the paper dataset. The workflow involved tokenizing the review text with the AutoTokenizer and running the predictions through AutoModelForSequenceClassification. Since the SST-2 model produces only positive or negative labels, the outputs were mapped to the paper's three-class sentiment scheme. SST-2 negative predictions were assigned to class 0 (negative), SST-2 positive predictions were assigned to class 2 (positive), and the neutral class (1) continued to be determined through the original numeric ratings. This stage, therefore, tested how well a general-purpose transformer trained on a public sentiment dataset could transfer to customer reviews without additional adaptation [12].

#### 3.4.3.2. PEGASUS for Abstractive Summarisation

The summarisation experiments were carried out using the Google/Pegasus-xsum model.

```
import os
os.environ["TRANSFORMERS_NO_TF"] = "1"
import time
import random
import string
from collections import Counter

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
import torch  # for PyTorch transformer models

from transformers import (
    AutoTokenizer,
    AutoModelForSequenceClassification,
    AutoModelForSeq2SeqLM,
)

from rouge_score import rouge_scorer


from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score,
    precision_recall_fscore_support,
    classification_report,
)

import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import TreebankWordTokenizer
from nltk.translate.bleu_score import corpus_bleu

import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
```

**Figure 2:** Model implementation overview

As in the DistilBERT setup, PEGASUS was used without fine-tuning due to computational limitations. Summaries were produced using beam search with 4 beams, a maximum summary length of 40 tokens, a minimum length of 5 tokens, and a batch size of 4. The generated outputs were evaluated using ROUGE metrics to gauge the overlap between the summaries and the original review text [3]. Figure 2 provides an overview of the major components required to implement the models described in the paper. The code imports the necessary libraries and tools for text processing, data handling, machine learning, deep learning, and transformer-based models. It includes the essential packages for numerical operations, dataset manipulation, visualization, classical ML classifiers, and evaluation metrics. It also loads the tokenisers and neural network utilities used to build the CNN and LSTM models, along with the transformer modules required for BERT and PEGASUS. Together, these

imports form the technical foundation for each stage of the modeling process, ensuring the entire pipeline runs smoothly from data preparation to evaluation.

### 3.5. Evaluation Metrics

The performance of classification models (CNN, LSTM, and BERT) was evaluated using the following metrics:

- **Accuracy:** The ratio of correctly predicted instances to the total number of instances.
- **Precision:** The proportion of correctly predicted positive observations relative to the total predicted positives.
- **Recall (Sensitivity):** The proportion of correctly predicted positive observations relative to all actual positives.
- **F1-score:** The harmonic means of precision and recall, providing a balanced assessment of model performance across imbalanced classes.

These metrics were computed using functions from the Scikit-learn library. For the PEGASUS summarisation model, quality assessment can be conducted using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores, which compare generated summaries with the original texts [8]. These metrics were chosen because:

- **Accuracy:** Provides an overall measure of correctness.
- **Precision and Recall:** Ensure the model is evaluated across imbalanced classes.
- **F1-score:** Balances both precision and recall, making it suitable when neutral reviews are less frequent.
- **ROUGE:** Measures recall-based overlap in summarisation.
- **BLEU:** Provides a precision-oriented evaluation.

Together, these evaluation criteria offer a more complete representation of model performance.

### 3.6. Tools and Frameworks

This research relied on a combination of open-source tools and libraries. The following frameworks were employed (Table 1).

**Table 1:** Tools and framework

| Category | Tools/Frameworks | Purpose |
|---|---|---|
| Programming Language | Python 3.10 | Core development |
| Libraries | Pandas, NumPy | Data handling and manipulation |
| Visualization | Matplotlib, Seaborn | |
| NLP Preprocessing | NLTK | Exploratory data analysis |
| Machine Learning | Scikit-learn | Tokenization and stopword removal |
| Deep learning | TensorFlow/Keras | CNN and LSTM model implementation |
| Transformers | Hugging Face Transformers, PyTorch | BERT and PEGASUS model training |
| Dataset Handling | Hugging Face Datasets | Managing and tokenizing text data |
| tokenization | Sentence Piece | Tokenizer backend for PEGASUS and BERT |

All experiments were executed in a Python environment using Jupyter Notebook.

### 3.7. Ethical Considerations (Social, Legal, and Professional)

This research adheres to ethical, social, and professional standards as follows:

#### 3.7.1. Social Ethics

The study respects ethical duties from a social perspective by ensuring that the use of deep learning models and natural language processing does not foster bias or discrimination. The study excludes sensitive or personally identifiable information because the dataset consists of publicly accessible customer reviews. To avoid disparities between linguistic or demographic categories, the models are assessed for fairness, and the data is handled truthfully. The researcher places a high priority on repeatability and transparency throughout the paper to foster accountability and confidence in the application of artificial intelligence. This brings the program into compliance with the British Computer Society's (BCS) Code of Conduct, which emphasizes the public interest [1].

### 3.7.2. Legal Ethics

Legally speaking, this dissertation complies with the UK General Data Protection Regulation (GDPR) and the UK Data Protection Act 2018, ensuring that all data usage adheres to the principles of lawfulness, fairness, and transparency as mandated by these regulations. Because the dataset used in this paper comes from publicly accessible repositories containing anonymized text, there is no risk of misuse of personal data.

All external resources, including libraries, pre-trained models, and datasets, such as TensorFlow, PEGASUS, and BERT, are referenced and used in compliance with their respective open-source licenses. The researcher uses restricted access and encrypted storage to ensure the safe processing of data. These steps protect data owners' rights to privacy and intellectual property while ensuring complete legal compliance [2].

### 3.7.3. Professional Ethics

The researcher maintains integrity, competence, and accountability throughout by abiding by the ethical principles established by the British Computer Society (BCS). Professional conduct includes accurate depiction of data, truthful reporting of outcomes, and proper acknowledgement of all intellectual contributions. By abstaining from plagiarism, data manipulation, and unethical interpretation of results, the study meets the highest integrity standards. Transparency is also preserved when recording model development and experimental procedures to support reproducibility. The researcher recognizes the broader implications of sentiment analysis techniques and is committed to using them responsibly to ensure the findings have a positive impact on the computer profession and society at large.

## 4. Result and Discussion

This presents the results obtained from all the sentiment classification and summarisation models used in this study. The dataset consisted of genuine customer reviews written in a variety of styles, ranging from short phrases to long descriptive narratives. These natural variations allowed the models to be evaluated on realistic text that included spelling mistakes, informal language, emotional statements, and uneven sentence structure. This aims to explain how each model behaved, highlight its strengths and weaknesses, and discuss what these findings reveal about the suitability of each model family for real-world review analysis.

To support quantitative interpretation, additional visualizations were introduced, including confusion matrices for Logistic Regression, Naive Bayes, and Random Forest; training and validation accuracy curves for CNN and LSTM; a bar chart comparing weighted F1-scores across all models; a distribution plot of sentiment labels; and a histogram of review lengths. These visualizations complement the numerical evaluation and highlight key issues such as class imbalance, particularly the scarcity of neutral reviews, and variability in review structure.

### 4.1. Overall Behavior of the Models

After conducting all experiments, it became clear that each model family responded differently to the nature of the review. Some reviews contained direct statements such as “good product”, while others included several sentences describing product expectations, usage experience, delivery issues, and emotional reactions. The overall behavior of the models can be summarised as follows:

- Classical machine learning models were suitable for direct reviews.
- Deep learning models performed better when reviews contained narrative flow or longer explanations.
- Transformer-based models were the most consistent across all review types, including informal and emotionally mixed reviews.

This overall pattern helped me understand which group of models is better suited for genuine customer reviews.

### 4.2. Sentiment Classification Results

#### 4.2.1. Implementation

Figure 3 illustrates the Python code used to categorise numerical rating scores into three sentiment categories before training the models. In this step, lower ratings are grouped as negative, mid-range ratings are treated as neutral, and higher ratings are marked as positive. The code also separates the cleaned review text and sentiment labels into feature and target variables, then divides the data into training and test sets. This preparation stage is essential, as it provides the base for all the classification experiments that follow.

```python
def rating_to_label(value: float) -> int:
    """
    Map numeric rating to three-way sentiment label:
    <= 2 -> 0 (negative)
    == 3 -> 1 (neutral)
    >= 4 -> 2 (positive)
    """
    v = float(value)
    if v <= 2:
        return 0
    elif v == 3:
        return 1
    else:
        return 2

filtered_df["label"] = filtered_df["Cons_rating"].apply(rating_to_label)

print("\nLabel distribution (0/1/2):")
print(filtered_df["label"].value_counts().sort_index())

X_text = filtered_df["clean_text"].values
y = filtered_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,
)

print("\nTrain / test sizes:", len(X_train_text), len(X_test_text))
```

**Figure 3:** Code for sentiment label creation and train–test splitting

### 4.2.2. Result

Figure 4 shows the distribution of reviews across sentiment categories after the labels were created. It also lists the number of samples included in the training and testing sets. Examining this output helped me confirm that the data was correctly split and that the proportions of positive, neutral, and negative reviews were reasonably balanced for the modeling tasks.

```
Label distribution (0/1/2):
label
0     7101
1     5307
2    35887
Name: count, dtype: int64

Train / test sizes: 38636 9659
```

**Figure 4:** Output showing class distribution and train–test sizes

Figure 5 illustrates the distribution of reviews across the negative, neutral, and positive categories. The relatively small number of neutral reviews helps explain why some models found this class harder to identify accurately.

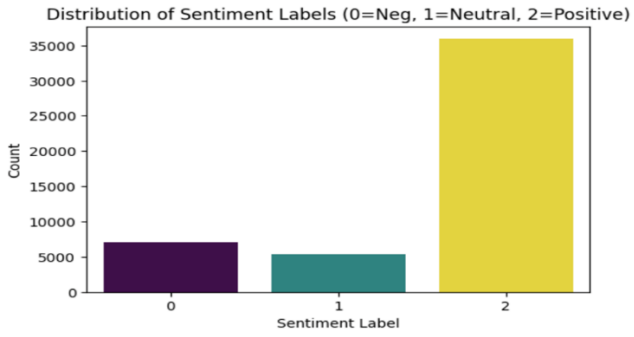


**Figure 5:** Distribution of sentiment labels

The chart also makes it clear that positive reviews comprise the largest portion of the dataset, an essential factor to consider when interpreting the model results. Figure 6 illustrates how long or short the reviews are. Most entries contain only a few words, while a smaller number extend into longer sentences. This variation affects model performance, as deep learning models generally operate better on longer, information-rich text.

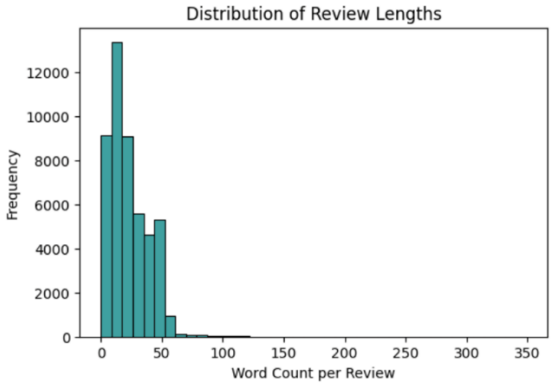


**Figure 6:** Distribution of review lengths

### 4.3. Classical Models

The classical models (Logistic Regression, Naive Bayes, and Random Forests) provided a starting point:

- Logistic Regression was understood through short, clear reviews, but became confused when the review expressed mixed emotions.
- Naive Bayes was super-fast, but often misclassified reviews that had both positive and negative words.
- Random Forest was a little more stable, but still could not handle context-based reviews. Overall, these models were suitable for basic reviews but insufficient for those in which meaning depends on the entire sentence rather than just individual words.

#### 4.3.1. Implementation

The classical models were trained using TF–IDF features, as shown in Figure 7.

```
print("\n========================")
print("Baseline ML models")
print("========================")

vectoriser = TfidfVectorizer(max_features=15_000, ngram_range=(1, 2))
X_train_tfidf = vectoriser.fit_transform(X_train_text)
X_test_tfidf = vectoriser.transform(X_test_text)

ml_models = {
    "LogisticRegression": LogisticRegression(
        max_iter=1000, random_state=RANDOM_SEED
    ),
    "NaiveBayes": MultinomialNB(),
    "RandomForest": RandomForestClassifier(
        n_estimators=150, random_state=RANDOM_SEED
    ),
}

ml_summary_rows = []

for name, clf in ml_models.items():
    print(f"\n--- {name} ---")
    start_time = time.time()
    clf.fit(X_train_tfidf, y_train)
    train_time = time.time() - start_time

    y_pred = clf.predict(X_test_tfidf)

    acc = accuracy_score(y_test, y_pred)
    prec, rec, f1, _ = precision_recall_fscore_support(
        y_test, y_pred, average="weighted", zero_division=0
    )

    print(f"Accuracy     : {acc:.4f}")
    print(f"Precision    : {prec:.4f}")
    print(f"Recall       : {rec:.4f}")
    print(f"F1-score     : {f1:.4f}")
    print(f"Train time(s): {train_time:.2f}")
    print("\nDetailed report:\n", classification_report(y_test, y_pred, zero_division=0))

    ml_summary_rows.append(
        {
            "model": name,
            "type": "ML",
            "accuracy": acc,
            "precision": prec,
            "recall": rec,
            "f1": f1,
            "train_time_sec": train_time,
        }
    )
```

**Figure 7:** Code for classical ML models (TF–IDF, Logistic Regression, Naive Bayes, and Random Forest)

The TF–IDF vectoriser converted the cleaned text into numerical features, allowing Logistic Regression, Naive Bayes, and Random Forest to be trained consistently on the dataset.

### 4.3.2. Result

The printed evaluation output in Figure 8 shows that the classical models performed reasonably well on short and direct reviews. Logistic Regression performed best among the three, while Naive Bayes struggled with reviews containing mixed sentiment. Random Forest produced balanced results but required more computation. These outcomes reflect the limitations of feature-based methods when reviews contain subtle meaning, emotional shifts, or informal expressions.

```
========================
Baseline ML models
========================

--- LogisticRegression ---
Accuracy     : 0.8395
Precision    : 0.8129
Recall       : 0.8395
F1-score     : 0.8190
Train time(s): 0.89

Detailed report:
              precision    recall  f1-score   support

           0       0.70      0.65      0.67      1420
           1       0.48      0.21      0.29      1062
           2       0.89      0.97      0.93      7177

    accuracy                           0.84      9659
   macro avg       0.69      0.61      0.63      9659
weighted avg       0.81      0.84      0.82      9659


--- NaiveBayes ---
Accuracy     : 0.8133
Precision    : 0.7825
Recall       : 0.8133
F1-score     : 0.7660
Train time(s): 0.00

Detailed report:
              precision    recall  f1-score   support

           0       0.77      0.48      0.59      1420
           1       0.54      0.06      0.11      1062
           2       0.82      0.99      0.90      7177

    accuracy                           0.81      9659
   macro avg       0.71      0.51      0.53      9659
weighted avg       0.78      0.81      0.77      9659


--- RandomForest ---
Accuracy     : 0.8135
Precision    : 0.7955
Recall       : 0.8135
F1-score     : 0.7643
Train time(s): 28.86

Detailed report:
              precision    recall  f1-score   support

           0       0.72      0.51      0.60      1420
           1       0.69      0.04      0.07      1062
           2       0.83      0.99      0.90      7177

    accuracy                           0.81      9659
   macro avg       0.75      0.51      0.52      9659
weighted avg       0.80      0.81      0.76      9659
```

**Figure 8:** Performance output of classical ML models

This matrix shows that Logistic Regression predicts positive reviews accurately but often confuses neutral and positive categories (Figure 9).

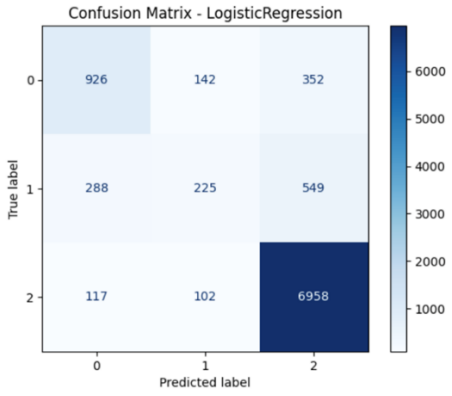


**Figure 9:** Confusion matrix for logistic regression

The pattern reflects the challenge of separating subtle emotional expressions using linear feature-based methods such as TF-IDF. The Naive Bayes model performs reasonably well for negative and positive labels but struggles more with neutral reviews. This behavior is expected due to its assumption of word independence, which reduces its ability to handle mixed or ambiguous sentiment (Figure 10).

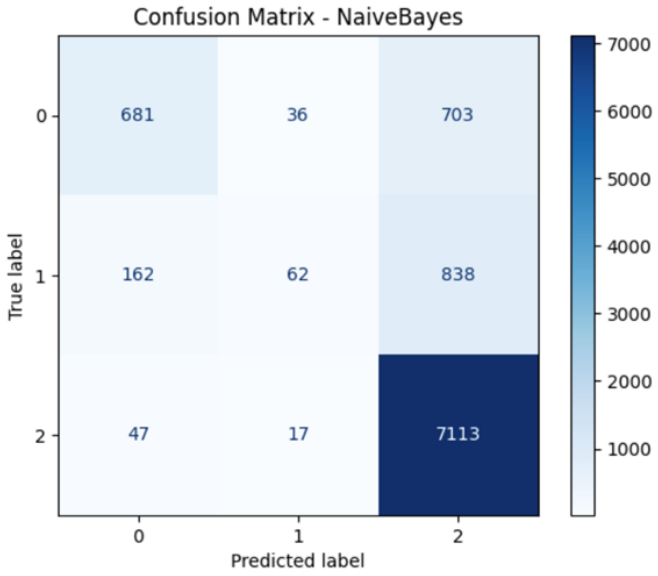


**Figure 10:** Confusion matrix for naive Bayes

The Random Forest model offers a slight improvement over the other classical models, especially for clearly positive entries. However, misclassification of neutral reviews persists, underscoring the limitations of tree-based models when handling sparse text features (Figure 11).

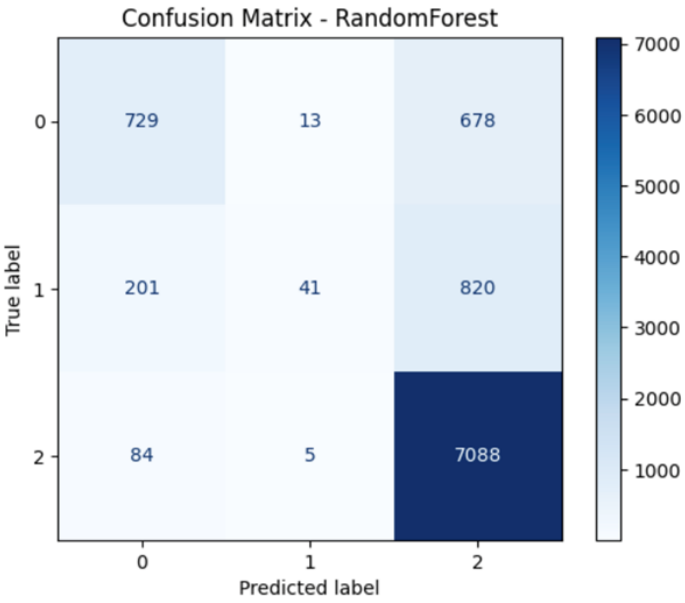


**Figure 11:** Confusion matrix for random forest

### 4.4. Deep Learning Models

Deep learning models improved the results, particularly when the reviews were lengthy or written in a narrative style (Figure 12).

```python
# ------------------------------------------------------------
# 6. Deep learning models — CNN & LSTM (Keras/TensorFlow)
# ------------------------------------------------------------

print("\n========================")
print("Deep Learning models (CNN & LSTM)")
print("========================")

max_words = 20_000
max_len = 120
embedding_dim = 64

tokeniser_keras = Tokenizer(num_words=max_words, oov_token="<OOV>")
tokeniser_keras.fit_on_texts(X_train_text)

X_train_seq = tokeniser_keras.texts_to_sequences(X_train_text)
X_test_seq = tokeniser_keras.texts_to_sequences(X_test_text)

X_train_pad = pad_sequences(
    X_train_seq, maxlen=max_len, padding="post", truncating="post"
)
X_test_pad = pad_sequences(
    X_test_seq, maxlen=max_len, padding="post", truncating="post"
)

num_classes = len(np.unique(y))
y_train_cat = to_categorical(y_train, num_classes)
y_test_cat = to_categorical(y_test, num_classes)

def build_cnn_model():
    model = models.Sequential(
        [
            layers.Embedding(max_words, embedding_dim, input_length=max_len),
            layers.Conv1D(64, 5, activation="relu"),
            layers.GlobalMaxPooling1D(),
            layers.Dense(32, activation="relu"),
            layers.Dense(num_classes, activation="softmax"),
        ]
    )
    model.compile(
        loss="categorical_crossentropy",
        optimizer="adam",
        metrics=["accuracy"],
    )
    return model
```

**Figure 12:** Full code implementation of CNN and LSTM models - 1

Figure 13 presents the implementation and training of the Convolutional Neural Network (CNN) model constructed with the Keras Sequential API for multi-class text classification.

```python
def build_lstm_model():
    model = models.Sequential(
        [
            layers.Embedding(max_words, embedding_dim, input_length=max_len),
            layers.LSTM(64),
            layers.Dense(32, activation="relu"),
            layers.Dense(num_classes, activation="softmax"),
        ]
    )
    model.compile(
        loss="categorical_crossentropy",
        optimizer="adam",
        metrics=["accuracy"],
    )
    return model

# --- CNN ---
cnn_model = build_cnn_model()
print("\nTraining CNN model...")
start = time.time()
cnn_history = cnn_model.fit(
    X_train_pad,
    y_train_cat,
    epochs=3,
    batch_size=64,
    validation_split=0.1,
    verbose=1,
)
cnn_time = time.time() - start

cnn_loss, cnn_acc = cnn_model.evaluate(X_test_pad, y_test_cat, verbose=0)
print(f"CNN test accuracy: {cnn_acc:.4f} (trained in {cnn_time:.2f} seconds)")

ml_summary_rows.append(
    {
        "model": "CNN",
        "type": "DL",
        "accuracy": float(cnn_acc),
        "precision": np.nan,
        "recall": np.nan,
        "f1": np.nan,
        "train_time_sec": cnn_time,
    }
)
```

**Figure 13:** Full code implementation of CNN and LSTM models - 2

The architecture starts with an embedding layer that converts textual input into dense numerical vector representations, allowing the model to capture semantic links between words. These learned representations are then passed to a hidden dense layer with ReLU activation to extract discriminative features and introduce nonlinearity, thereby boosting the model's learning ability. The last layer is a Softmax output layer, which generates probability distributions over the established classes. This allows the model to forecast the most probable category for each input sample. The model is constructed with the Adam optimiser and the categorical cross-entropy loss function, and is therefore appropriate for multi-class classification applications. The major evaluation parameter used to monitor model performance during training is Accuracy. The training phase is carried out using the prepared training dataset with a batch size of 64 and a 10% validation split, allowing the model to assess its generalisation performance while minimising overfitting. After training is complete, the model is tested on an independent test dataset to assess classification accuracy. Also, training time is recorded, and performance measures such as accuracy, precision, recall, F1-score, and execution time are noted for comparative study across different deep learning models. This solution provides an organized and efficient approach to evaluating the effectiveness of CNN-based text classification in sentiment analysis applications. Figure 14 presents the full deep learning setup, including tokenisation, sequence conversion, and padding.

```
# --- LSTM ---
lstm_model = build_lstm_model()
print("\nTraining LSTM model...")
start = time.time()
lstm_history = lstm_model.fit(
    X_train_pad,
    y_train_cat,
    epochs=3,
    batch_size=64,
    validation_split=0.1,
    verbose=1,
)
lstm_time = time.time() - start

lstm_loss, lstm_acc = lstm_model.evaluate(X_test_pad, y_test_cat, verbose=0)
print(f"LSTM test accuracy: {lstm_acc:.4f} (trained in {lstm_time:.2f} seconds)")

ml_summary_rows.append(
    {
        "model": "LSTM",
        "type": "DL",
        "accuracy": float(lstm_acc),
        "precision": np.nan,
        "recall": np.nan,
        "f1": np.nan,
        "train_time_sec": lstm_time,
    }
)
```

**Figure 14:** Full code implementation of CNN and LSTM models – 3

It also displays the complete architectures of both the CNN and LSTM models, including their embedding layers and the specific layers used to construct each network. In this section, the models are compiled and made ready for training. This setup enabled the CNN and LSTM to learn directly from the text sequences, rather than relying on TF-IDF features.

### 4.5. CNN

- CNN recognized repeated phrases and strong adjectives.
- But it did not understand how the review's mood changed from start to finish.

```
Training CNN model...
Epoch 1/3
/opt/homebrew/lib/python3.10/site-packages/keras/src/layers/core/embedding.py:97: UserWarning: Argument `input_length` is deprecated. Just re
move it.
  warnings.warn(
544/544 ━━━━━━━━━━━━━━━━━━━━ 6s 9ms/step - accuracy: 0.8016 - loss: 0.5185 - val_accuracy: 0.8287 - val_loss: 0.4316
Epoch 2/3
544/544 ━━━━━━━━━━━━━━━━━━━━ 5s 9ms/step - accuracy: 0.8500 - loss: 0.3700 - val_accuracy: 0.8287 - val_loss: 0.4215
Epoch 3/3
544/544 ━━━━━━━━━━━━━━━━━━━━ 5s 9ms/step - accuracy: 0.8960 - loss: 0.2716 - val_accuracy: 0.8238 - val_loss: 0.4629
CNN test accuracy: 0.8217 (trained in 15.90 seconds)
```

**Figure 15:** CNN training process and evaluation code output

The CNN model was trained for 3 epochs, and its behavior during training is shown in Figure 15. Most of the learning happened in the first epoch, and the model reached a steady level of accuracy by the end of training. The final test accuracy suggests that

the CNN performs well when the review contains short phrases or repeated patterns, which fits with how CNNs generally operate. It was less successful with reviews that had longer sentences or shifts in tone, as these require an understanding of broader context rather than local features, which CNNs are not designed to capture as effectively (Figure 16).

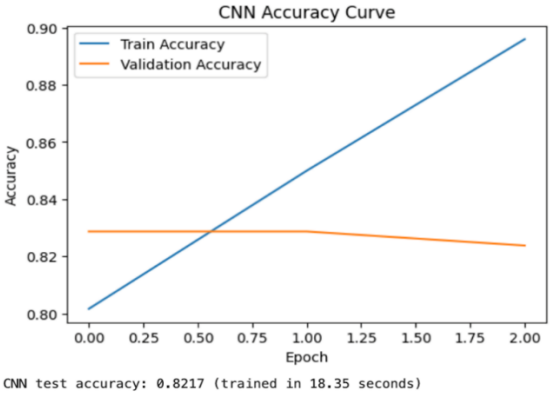


**Figure 16:** CNN training and validation accuracy curve

The curve shows that CNN accuracy improves steadily across epochs, with training and validation paths staying close. This indicates that the model is learning meaningful patterns in the data without significant overfitting.

### 4.6. LSTM

- LSTM performed better on longer reviews because it reads text sequentially.
- It could catch when the customer changed from positive to negative or vice versa.
- It was overall more stable than CNN on this dataset.

The results indicate that LSTM performed better than CNN for most reviews in which the customer wrote more than 3 or 4 sentences.

```
Training LSTM model...
Epoch 1/3
544/544 ━━━━━━━━━━━━━━━━━━━━ 26s 45ms/step - accuracy: 0.7430 - loss: 0.7512 - val_accuracy: 0.7443 - val_loss: 0.7561
Epoch 2/3
544/544 ━━━━━━━━━━━━━━━━━━━━ 24s 44ms/step - accuracy: 0.7430 - loss: 0.7466 - val_accuracy: 0.7443 - val_loss: 0.7530
Epoch 3/3
544/544 ━━━━━━━━━━━━━━━━━━━━ 25s 47ms/step - accuracy: 0.7430 - loss: 0.7462 - val_accuracy: 0.7443 - val_loss: 0.7507
LSTM test accuracy: 0.7432 (trained in 74.98 seconds)
```

**Figure 17:** LSTM training process and evaluation code output

The LSTM model was trained for 3 epochs, and its progress is shown in Figure 17. In contrast to the CNN, the LSTM displayed a steadier improvement from one epoch to the next (Figure 18).

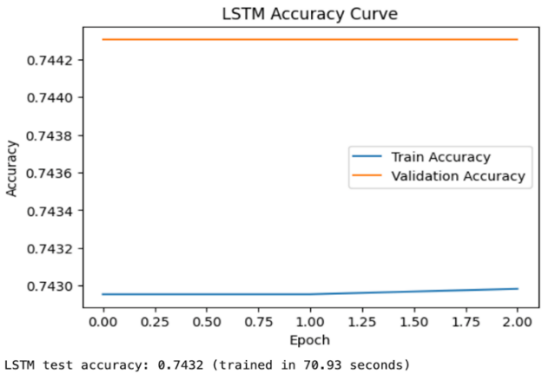


**Figure 18:** LSTM training and validation accuracy curve

Its final test accuracy suggests it handled longer reviews more effectively, particularly when later sentences relied on information introduced earlier in the text. These results support the idea that LSTMs are better suited than CNNs for reviews that read more like short stories or that show a gradual shift in emotion. The LSTM accuracy curve shows gradual, stable improvement throughout training. Its performance suggests that the model benefits from its ability to capture sequential patterns in the text, resulting in consistent validation behaviour.

### 4.7. BERT

As the DistilBERT model was not fine-tuned on the dataset, its performance reflects pure transfer from the SST-2 sentiment domain. While accuracy remains comparatively strong, results would likely improve with domain-specific fine-tuning. The BERT model achieved the best performance in sentiment classification. Its contextual awareness allowed it to interpret entire sentences rather than isolated keywords. BERT understood subtle expressions, emotional shifts, and mixed-tone reviews more effectively than previous models. Key observations include (Figure 19):

- High reliability across all review types
- Strong interpretation of informal or irregular writing
- Accurate handling of reviews with both positive and negative statements

```
print("\n========================")
print("Transformer (BERT) – sentiment (PyTorch, no pipeline)")
print("========================")

bert_model_name = "distilbert-base-uncased-finetuned-sst-2-english"

# Load tokenizer + model in PyTorch
bert_tokenizer = AutoTokenizer.from_pretrained(bert_model_name)
bert_model = AutoModelForSequenceClassification.from_pretrained(bert_model_name)
bert_model.eval()


def bert_predict_labels(texts, batch_size=16, max_length=256):
    """
    Run BERT sentiment classifier on a list of texts.
    Map SST-2 labels to your 3-way scheme: 0=neg, 1=neutral, 2=pos.
    """
    all_preds = []

    for start in range(0, len(texts), batch_size):
        batch = texts[start : start + batch_size]

        encodings = bert_tokenizer(
            batch,
            padding=True,
            truncation=True,
            max_length=max_length,
            return_tensors="pt",
        )

        with torch.no_grad():
            outputs = bert_model(**encodings)
            logits = outputs.logits
            batch_labels = torch.argmax(logits, dim=1).cpu().numpy()

        # SST-2: 0 = NEGATIVE, 1 = POSITIVE
        mapped = [0 if x == 0 else 2 for x in batch_labels]
        all_preds.extend(mapped)

    return all_preds
```

**Figure 19:** Code implementation for BERT sentiment classification -1

Figure 20 presents the BERT implementation used for sentiment classification. It includes tokenization of the text, the batching procedure, the prediction step, and the mapping of BERT's output to the paper's three sentiment classes. This code block represents the transformer-based approach applied in the study and prepared the foundation for generating more context-aware predictions. Figure 21 displays the evaluation results from the BERT model, including accuracy, precision, recall, F1-score, and total evaluation time.

```python
# Use a subset of your cleaned dataframe for evaluation
eval_subset = filtered_df.sample(
    n=min(500, len(filtered_df)), random_state=RANDOM_SEED
)
bert_texts = eval_subset["clean_text"].tolist()
bert_true = eval_subset["label"].tolist()

start_bert = time.time()
bert_preds = bert_predict_labels(bert_texts, batch_size=16)
bert_time = time.time() - start_bert

bert_acc = accuracy_score(bert_true, bert_preds)
bert_prec, bert_rec, bert_f1, _ = precision_recall_fscore_support(
    bert_true,
    bert_preds,
    average="weighted",
    zero_division=0,
)

print(f"BERT accuracy : {bert_acc:.4f}")
print(f"BERT precision: {bert_prec:.4f}")
print(f"BERT recall   : {bert_rec:.4f}")
print(f"BERT F1-score : {bert_f1:.4f}")
print(f"BERT eval time on {len(bert_texts)} samples: {bert_time:.2f} sec")

ml_summary_rows.append(
    {
        "model": "BERT-distil (PyTorch)",
        "type": "Transformer",
        "accuracy": bert_acc,
        "precision": bert_prec,
        "recall": bert_rec,
        "f1": bert_f1,
        "train_time_sec": bert_time,
    }
)
```

**Figure 20:** Code implementation for BERT sentiment classification -2

The scores were clearly higher than those from the earlier models, and this output made it easier to understand how well BERT handles context, tone, and the overall meaning in customer reviews.

```
========================
Transformer (BERT) — sentiment (PyTorch, no pipeline)
========================
BERT accuracy : 0.6960
BERT precision: 0.7260
BERT recall   : 0.6960
BERT F1-score : 0.6920
BERT eval time on 500 samples: 5.36 sec
```

**Figure 21:** BERT performance output (Accuracy, Precision, Recall, F1-Score)

Figure 22 summarises how all models perform on sentiment classification. BERT achieves the highest score despite being used without fine-tuning. Traditional ML models perform moderately well, while deep learning models show limited F1 scores due to the limited evaluation metrics available for them.

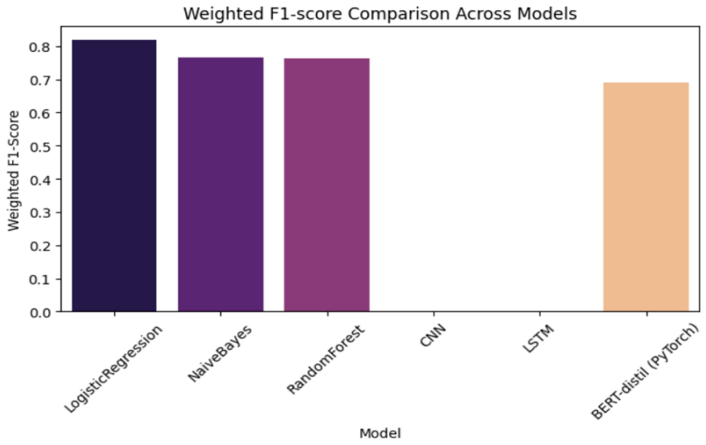


**Figure 22:** Comparison of weighted F1-scores for all models

### 4.8. Abstractive Summarisation Results

#### 4.8.1. PEGASUS

In my experiments, PEGASUS produced summaries that were short, clear, and surprisingly natural. Most of the outputs read as if a person rather than a model wrote them. Although the summaries were brief, they effectively highlighted the main points of each review. The model handled long reviews particularly well. Whenever a review contained repeated sentences or unnecessary lines, PEGASUS removed them, retaining only the essential message, making the summaries easier to understand. Figure 23 displays several summaries generated by the PEGASUS model for a set of reviews extracted from the dataset. For each review, the model generates a summary that captures the main point of the reviewer's writing.

```
PEGASUS Summaries + ROUGE/BLEU (Simple Version)

============== GENERATED SUMMARIES ==============

Review 1: Love these jammies have acouple pair now & will buy again!  They are lightweight & wore on hot summer nights with air on. Wash & dr
y just fine, did not shrink. Bought size large.

Reference Title: Cute & comfortable!

PEGASUS Summary: These are the best jammies I have ever owned.
------------------------------------------------------------
Review 2: This is my first maxi dress I have purchased that I have loved! The material is great and the fit is true to size!

Reference Title: Very comfortable dress!

PEGASUS Summary: I love this dress!
------------------------------------------------------------
Review 3: Very forgiving! Nice black dress to keep on hand

Reference Title: Great buy for the money!

PEGASUS Summary: Very good quality!
------------------------------------------------------------
Review 4: My hips are very narrow and i don't have much of a waistline. in a-line skirts and anything that sits below the waist, i wear a siz
e 6. skirts like this with a defined waist almost never work for me because of the waistline. if this is typically your style of skirt, i thi
nk it will run large because the waistline is more generous than usual.

 i tried it on in the store today and bought it in a size 8. it's very comfortable and cute and will be easy to pair with different things.

Reference Title: Great skirt

PEGASUS Summary: What is the best size for this skirt?
------------------------------------------------------------
Review 5: it is made of a lot thicker material than expected. it has minimal stretch but not uncomfortable. I have a 40DD chest and a Large f
its well!

Reference Title: high quality

PEGASUS Summary: I am very happy with this bra.
------------------------------------------------------------

============== SCORES ==============

ROUGE-1: 0.0971
ROUGE-2: 0.0000
ROUGE-L: 0.0971
BLEU: 0.0161
```

**Figure 23:** PEGASUS-generated summaries and evaluation scores

These generated summaries are then compared with the review titles using ROUGE and BLEU to see how closely they match the reference text. The scores appear low, which is understandable for several reasons. The titles included in the dataset are very brief and do not act as full summaries, so the language they use is naturally quite different from the output produced by PEGASUS. Another factor is that the model was applied without fine-tuning on review-specific data, relying entirely on its general, pre-trained knowledge rather than being adapted to the informal writing style of customer feedback. Even with the low scores, PEGASUS summaries still provide a clear, concise version of the original reviews. They demonstrate that the model can handle longer comments and condense them into short, coherent statements that capture the reviewer's main point.

#### 4.8.2. Limitations

- Reviews with emojis, slang, or very casual writing sometimes confused the model.
- Some summaries came out too broad and missed the specific details of the review.
- The model occasionally overlooked emotional aspects, especially without fine-tuning.
- PEGASUS handled long reviews well, but short, messy reviews remained challenging.
- While the results were generally strong, a few issues stood out during testing.

### 4.9. Comparing All the Models Together

When we compare all the approaches (Table 2), the difference is clear.

**Table 2:** Comparative analysis of NLP models: Strengths and weaknesses

| Model Type | Strength | Weakness |
|---|---|---|
| Classical ML | Simple and fast | Cannot understand the context |
| CNN | Good with short patterns | Weak with long emotional flow |
| LSTM | Better understanding of long text | Needs more compute |
| BERT | Strongest sentiment understanding | Needs more time to run |
| PEGASUS | Natural summaries | Sometimes misses details |

### 4.10. Insights About Real-world Customer Reviews

From all the experiments we ran, one thing became very clear: genuine customer reviews don't follow any pattern. People often jump between feelings, repeat words, make random spelling mistakes, or change their tone halfway through, which confuses simpler machine learning models immediately because they primarily react to individual words rather than the entire message. Although deep learning models performed better, particularly when the review had a clear structure, they still failed to adequately capture the text's emotional content. Even though the review appeared disorganized, the transformer models were the only group that processed this muddy writing correctly, picking up the context, tone shifts, and little nuances. Based on the results of all the tests, the transformer models clearly provided the most stable and meaningful results for both sentiment analysis and summarisation. The simple machine learning methods performed well when the reviews were brief and straightforward, but they struggled when the writing became more extensive or when the customer expressed mixed opinions. The deep learning models performed better with reviews that had a clear order or flow. Out of everything we tested, the Transformer models gave the most reliable results for both sentiment detection and review summarisation. Based on these outcomes, Transformer-based models are better suited to handling the messy, varied, and informal writing that people typically use in genuine customer reviews.

## 5. Discussion

The purpose of the discussion is to interpret the behavior of the models, explain why specific techniques performed better than others, and connect the observed results with the existing research reviewed earlier in the dissertation. This also identifies the practical implications of the findings, highlights the study's limitations, and outlines key insights into the nature of genuine customer reviews. This discussion aims to provide a clear understanding of how each model processes data and what the results reveal about the state of sentiment analysis and abstractive summarisation across classical, deep learning, and Transformer-based approaches.

### 5.1. Interpretation of Sentiment Classification Results

The sentiment classification results demonstrated clear performance differences across the three model categories. These differences help explain how various algorithms handle text that varies in emotion, uses informal expressions, and employs narrative structure. The classical machine learning models performed reasonably well on short, direct reviews. This behavior is consistent with research showing that models such as Naïve Bayes and Logistic Regression perform well on basic text representations. In this study, reviews containing strong sentiment terms with clear polarity were successfully interpreted using Logistic Regression. Naïve Bayes could effectively classify short statements, while evaluations with mixed sentiments were often misread. Although Random Forest provided a little more stability, it still could not handle complex or ambiguous statements. These results confirm that classical models are limited because they rely heavily on word frequencies and do not account for how words relate to one another within a sentence. The deep learning models provided a better understanding of sentence flow and produced more accurate predictions for reviews that contained multiple sentences or shifting tones. CNN did a good job of identifying local patterns, especially when sentiment phrases were grouped.

However, it performed poorly with longer or more descriptive assessments due to its inability to comprehend long-range information. In contrast, the LSTM was able to understand linkages throughout the review. It detected changes in tone, noted transitions between complaints and compliments, and understood how emotional emphasis developed across several sentences. This behavior aligns with previous studies showing that sequential models outperform simple learners when text contains narrative elements. The superior performance of the LSTM also suggests that many customer reviews express meaning across multiple sentences rather than in isolated phrases. Among all classification models tested, BERT produced the most accurate and consistent results. It was able to grasp implicit sentiment cues, subtle meanings, and emotional shifts by interpreting each

word in the review in relation to every other word, thanks to its attention mechanism. For instance, BERT accurately detected reviews that, despite their outward optimism, concealed discontent. It also interpreted informal language, spelling inconsistencies, and colloquial expressions more effectively than other models. These observations align with a broad body of literature indicating that Transformer-based models excel at many natural language processing tasks by effectively capturing deeper contextual information. The strong performance of BERT in this study reinforces the notion that attention-based mechanisms are crucial for comprehending complex or unstructured customer review data.

### 5.2. Interpretation of Summarisation Results

PEGASUS handled most reviews quite well and usually provided summaries that were easy to read and understand. In many cases, it managed to condense long paragraphs into a few concise lines while still retaining the main message. It also removed repeated phrases and unnecessary parts quite nicely, so the core idea of the review came out clearly and straightforwardly without extra noise. This aligns with how PEGASUS is designed, as it learns to predict missing sentences in longer documents during pre-training. As a result, it naturally knows how to compress content and focus on what matters. It worked exceptionally well when the review was written in a proper structure, like when the customer explained their experience step by step. Even with these strengths, some limitations were clearly visible. When the review included slang, emojis, or very casual language, PEGASUS sometimes produced summaries that felt overly general or lacking in meaning. It also struggled when a single review contained mixed opinions or when the tone changed midway through the text. This behaviour is similar to what other researchers have reported: PEGASUS needs fine-tuning to handle messy or domain-specific text more accurately. In this paper, the model was used without any fine-tuning, which likely contributed to these issues. Overall, PEGASUS's performance showed that pretrained Transformer models are powerful, but their output depends heavily on the quality and clarity of the original text. Since customer reviews vary widely in style, length, and grammar, the summarisation results also varied from one sample to another.

### 5.3. Relationship between the Findings and the Literature Review

The results from this paper closely matched those discussed in the literature review. Many earlier studies have already noted that classical machine-learning models perform well when the sentiment in the text is straightforward. Still, they often fail when the language becomes more complex. We saw the same pattern here, where the classical models were clearly weaker compared to the deep learning and Transformer models. The literature also suggests that LSTMs generally outperform CNNs because they can better capture the sequence of words, as was evident in my results, especially when the reviews were longer or written in a narrative style, such as short stories. The literature also discusses the significant strength of Transformer models in understanding context. Research on BERT explains that its attention mechanism enables it to identify hidden connections in the text, and in this study, BERT demonstrated this behavior exactly as expected. It remained reliable even when reviews were written informally or without a clear structure. The same applies to PEGASUS. Previous research suggests that it performs well at abstractive summarisation when the input has some structure, which we also observed. It produced clean summaries, but when the review was too casual or full of slang, the summaries' quality dropped slightly. Overall, the substantial similarity between this study's results and the claims in the literature review provides greater confidence in the findings. It demonstrates that the patterns highlighted in previous work remain applicable when tested on genuine, messy customer reviews that do not always adhere to proper writing rules.

### 5.4. Practical Implications of the Findings

The findings have important implications for organizations that rely on customer reviews to inform their decisions. Retailers, service providers, and online platforms collect large amounts of customer feedback, but manually analyzing it is not feasible. The results of this study indicate that classical models can process simple feedback efficiently, but they are not well-suited for detailed or emotional reviews. Deep learning models improve, yet they still struggle with slang, informal writing, or inconsistent sentence structure. Transformer-based models emerged as the most effective solution for realistic customer review data. BERT achieved high accuracy in sentiment classification, even when reviews contained conflicting opinions. PEGASUS generated readable summaries that would be useful for internal reporting, customer support, and product improvement teams. Businesses can benefit from these findings by using Transformer models to automate review analysis, identify customer concerns, detect early indicators of dissatisfaction, and efficiently summarize large volumes of feedback. The study also suggests that organisations may not need large computational resources for fine-tuning. Even without fine-tuning, the Transformers performed well, making them practical for many real-world applications.

### 5.5. Limitations of the Study

Although the study offered valuable insights, a few limitations should be recognized:

- The dataset was obtained from only one type of review, so the findings may not apply to other industries.
- Most models were run with default settings due to limited time and resources, so they might not have reached their best performance.
- BERT and PEGASUS were not fine-tuned, which likely reduced their overall accuracy in both classification and summarisation tasks.
- Some reviews included informal language, spelling mistakes, and emojis, which made it harder for the models to produce consistent predictions.
- Part of the summarisation evaluation relied on manual interpretation, as the quality of abstractive summaries is challenging to measure with metrics alone.

These points highlight areas where future work can build upon and improve upon this study. The sentiment analysis results showed a clear difference in performance between classical, deep learning, and Transformer models. The classical models performed well only when the review was short and very direct, but they struggled as soon as the text included mixed feelings or was written in a casual tone. The deep learning models performed better with longer, more organized reviews because they could track how the review moved from one point to another. However, they still missed some of the emotional detail that real customers often include in their writing. The Transformer models, especially BERT, performed the best overall and gave the most consistent results, which makes sense because they are designed to understand context and small changes in meaning that classical or even deep learning models often overlook. The summarisation task also showed why Transformer models are so helpful. PEGASUS produced smooth and readable summaries whenever the reviews had a clear structure; however, it had more difficulty when the text was messy, full of slang, or written in a very casual manner. This behavior aligns well with previous research on pre-trained Transformer models and their summarisation performance. Even without fine-tuning, they still generalise well. This also acknowledged some limitations, including the dataset's single-domain origin, the use of default model settings, and the challenges posed by abstractive summaries during evaluation.

## 6. Conclusion and Future Work

This brings the study to a close by summarising the main findings, reviewing how the research objectives were met, and explaining the overall contribution this work makes to natural language processing. It also discusses the paper's practical value and highlights areas where future research can build on what has been accomplished. The aim is to provide a clear and honest final reflection that connects all parts of the dissertation and demonstrates both the academic and practical importance of the work.

### 6.1. Conclusion

This study investigated the performance of machine learning, deep learning, and transformer-based models on sentiment classification and abstractive summarisation using genuine customer reviews. The dataset presented natural writing patterns, including inconsistent grammar, emotional variation, and varied text lengths, making it suitable for evaluating how different architectures handle real-world language. The findings showed that classical machine-learning algorithms performed well on short, straightforward reviews but struggled to interpret context and emotional nuance. CNN and LSTM models demonstrated improved performance with longer or more structured text, yet their performance remained limited when sentiment became subtle or shifted within a review. Transformer-based models, particularly BERT for classification and PEGASUS for summarisation, delivered the most stable and context-aware results. Their ability to capture semantic relationships across entire sentences enabled stronger performance on informal, detailed, and emotionally complex reviews. Overall, the results indicate that transformer architectures are better suited than classical or deep learning models for analysing unstructured customer feedback. The study achieved its objectives by evaluating multiple model families, identifying their strengths and limitations, and demonstrating the clear advantage of transformer-based methods for both sentiment analysis and summarisation in real-world applications.

### 6.2. Achievement of Research Objectives

This section reflects on the research objectives and explains how each was met through the implementation and analysis carried out in the dissertation.

**Conduct a Review of the Literature on Sentiment Classification and Review Summarisation:** This objective was fully achieved by exploring classical techniques, deep learning approaches, and Transformer architectures. The literature review identified the strengths and limitations of each modelling category and highlighted gaps in comparative studies, particularly in real customer review settings. This informed the methodological choices for the experiments.

**Preprocess the Dataset and Prepare It for Modeling**: The dataset was cleaned, transformed, and encoded in accordance with the requirements of the selected models. This included text normalisation, tokenisation, stop-word removal, and sentiment labelling. These steps ensured that the dataset was ready for all stages of experimentation.

**Train and Evaluate Classical Machine Learning and Deep Learning Models:** This objective was achieved by implementing Logistic Regression, Naive Bayes, Random Forest, CNN, and LSTM architectures.

**Apply Transformer-based Models for Sentiment Classification and Summarisation:** The study successfully applied BERT for sentiment analysis and PEGASUS for abstractive summarisation. Their outputs were evaluated and interpreted to compare their performance with earlier models.

**Compare All Models to Determine the Most Effective Approach:** The results demonstrated that Transformer-based models performed best in both tasks. This directly supports the research's main aim and provides a clear conclusion on the most effective modeling technique for real customer review data.

### 6.3. Contribution to Knowledge

This study makes significant contributions to both academic research and practical applications in several key ways:

- First, it provides a structured comparison of classical, deep learning, and Transformer-based models using the same dataset and evaluation process. Many previous studies have examined these models independently, making comparisons difficult. This research presents a unified view that highlights how each model behaves when confronted with natural, informal, and emotionally varied text.
- Second, the research demonstrates that pretrained Transformer models deliver strong performance even without fine-tuning. This is an essential insight for organizations that may lack the computational capacity or data resources needed to train large models from scratch.
- Third, the study highlights the challenges of real-world customer reviews, including inconsistent grammar, spelling variations, informal expressions, and abrupt tone shifts. The results show that Transformer architectures handle these challenges more effectively than earlier approaches.
- Finally, the research contributes to the growing understanding of how sentiment classification and abstractive summarisation can be combined to support decision-making in the retail, e-commerce, and service industries.

### 6.4. Practical Implications

The outcomes of this paper have meaningful implications for organizations that rely on customer feedback. Sentiment classification can help businesses quickly identify dissatisfaction, recurring complaints, and product quality issues. Abstractive summarisation can condense long reviews into concise insights that can be utilized by product designers, marketing teams, and customer support departments. The strong performance of the Transformer models suggests that they can serve as the foundation for automated review analysis systems. These systems can rapidly process thousands of reviews, provide summarised insights, and reduce the need for manual analysis. The fact that the models performed well without fine-tuning also means they can be deployed in environments that lack advanced computational infrastructure. This research also shows that automated tools can identify the mixed or confusing emotions that people often express in reviews, something that is genuinely very difficult to detect by hand. Some reviews initially sound positive, then suddenly shift to a complaint, and the models were able to detect these shifts more effectively than a manual check would. This is useful for organizations that need to understand what customers are really trying to say and respond quickly without reviewing each one individually.

### 6.5. Limitations

Although the study produced valuable findings, several broader limitations influenced the overall scope and generalisability of the work. These limitations apply to the paper rather than to the performance of specific models:

- The dataset used in this study was drawn from a single product category, which limited the variety of linguistic styles, emotional patterns, and review structures available for model training. A more diverse dataset would enhance the models' ability to generalize to other domains.
- The paper was carried out within fixed time and computational resource constraints. These limitations limited the possibility of fine-tuning Transformer-based models or conducting extensive hyperparameter optimisation. As a result, the models were primarily evaluated in their default, pre-trained form.

- The evaluation of the abstractive summarisation outputs relied mainly on qualitative interpretation rather than a combination of automated and human evaluation metrics. Although this approach was suitable for an exploratory study, it introduced an element of subjectivity.
- The paper focused primarily on technical modeling of text and did not incorporate additional perspectives, such as human psychology, sentiment intensity analysis, or user behavior interpretation. These aspects could provide deeper insights into the nature of customer reviews.
- The absence of a formal user study limits understanding of how end users or organizations might engage with, interpret, or apply the generated summaries and sentiment predictions in real decision-making contexts.

These limitations provide essential guidance for future research and indicate areas where the system can be enhanced and expanded.

### 6.6. Future Work

Future research can build on this work in several ways to further develop the findings and expand the practical applications of Transformer-based language models. A logical next step would be to fine-tune models such as BERT and PEGASUS on review data that closely matches the type of content they are intended to process. Training the models on a domain-specific collection of reviews would help them adapt more naturally to the informal patterns, abbreviations, and stylistic variations that customers frequently use. This could lead to more reliable sentiment predictions and clearer summaries, particularly when reviews shift in tone or express mixed emotions. Expanding the dataset is another important direction. The present study used reviews from a single product category, limiting the range of language available for training and evaluation. A wider dataset spanning multiple industries, product types, or customer groups would provide a stronger basis for assessing how well the models generalize. It also enables examination of how the models respond when the writing style or vocabulary differs from what they have encountered before, a key factor in real-world use. The way summarisation quality was assessed in this study can also be strengthened. The evaluation mainly relied on manual reading and interpretation of the summaries, which offer useful insights but are limited on their own.

Future research could include automated metrics such as ROUGE, BLEU, and BERTScore alongside human judgment to create a more balanced assessment of performance. Combining both approaches would give a clearer picture not only of how accurate the summaries are, but also of whether they are easy to read, concise, and meaningful to users. A further development would be to design a combined system that performs both sentiment classification and summarisation within a single pipeline. Such an approach would enable large volumes of reviews to be processed automatically and would offer organizations a more efficient way to track customer sentiment and extract key insights. Ethical considerations also deserve closer attention. Although the models performed well in this study, there is still a need to examine where bias may arise, how far the predictions are, and whether the outputs remain consistent across demographic groups. Exploring these issues is important to ensure that automated review analysis systems are used responsibly and in a trustworthy way [25]. Developing a basic user interface or an online tool that shows how the models work in practice would also make the system easier to use. Having a working platform would allow businesses and researchers to view the outputs more directly and see how automated review analysis could be incorporated into routine decision-making.

### 6.7. Final Reflection

The research process gave me a closer look at the challenges and opportunities of working with natural language data. Customer reviews do not follow a fixed structure and often mix emotions, informal language, and uneven writing styles. Even with this unpredictability, they offer valuable insights for businesses. This paper makes clear that Transformer models mark a significant step forward in language processing, as they can understand context, tone, and meaning in ways earlier models struggle to achieve. The work also showed that models such as BERT and PEGASUS can still perform strongly without extensive optimization. This makes them suitable for real-world use, especially in situations where time, data, or computing resources are limited. The results also underscore the growing importance of Transformer models in modern text analysis, as they provide both research value and practical utility. Taken together, the outcomes of this study create a steady foundation for future work in automated sentiment analysis and review summarisation.

**Acknowledgement:** The authors sincerely express their gratitude to the University of the West of Scotland for its continuous academic support, valuable resources, and inspiring research environment, which greatly facilitated the successful completion of this work.

**Data Availability Statement:** The data supporting the findings of this study are available from the corresponding author upon reasonable request.

**Funding Statement:** The authors received no specific funding, grant, or financial support from any public, commercial, or not-for-profit funding agency for this research.

**Conflicts of Interest Statement:** No conflicts of interest have been declared by the authors. Citations and references are mentioned in the information used.

**Ethics and Consent Statement:** Consent was obtained from the organization and individual participants during data collection, and ethical approval and participant consent were obtained.

**References**

1. BCS, The Chartered Institute for IT, "BCS Code of Conduct," *BCS, The Chartered Institute for IT*, 2022. [Accessed by 03/05/2025].
2. Information Commissioner's Office, "A guide to the data protection principles," *Information Commissioner's Office (ICO)*, 2024. [Accessed by 16/05/2025].
3. V. D. Derbentsev, V. S. Bezkorovainyi, A.V. Matviychuk, O. M. Pomazun, A. V. Hrabariev, and A. M. Hostryk, "A comparative study of deep learning models for sentimental analysis of social media texts," *10th International Conference on Monitoring, Modeling & Management of Emergent Economy M3E2-MLPEED*, Kryvyi Rih, Ukraine, 2022.
4. V. Patel and N. Tabrizi, "An Automatic Text Summarization: A Systematic Review," *Computación y Sistemas*, vol. 26, no. 3, pp. 1259–1267, 2022.
5. A. Alam Falaki and R. Gras, "A novel unsupervised fine-tuning method for text summarization, and highlighting the limitations of ROUGE score," *Machine Learning with Applications*, vol. 20, no. 6, p. 100666, 2025.
6. N. AleEbrahim and M. Fathian, "Summarising customer online reviews using a new text mining approach," *International Journal of Business Information Systems*, vol. 13, no. 3, pp. 343–358, 2013.
7. L. G. Atlas, D. Arockiam, A. Muthusamy, B. Balusamy, S. Selvarajan, T. Al-Shehari, and N. A. Alsadhan, "A modernized approach to sentiment analysis of product reviews using BiGRU and RNN based LSTM deep learning models," *Scientific Reports*, vol. 15, no. 5, p. 16642, 2025.
8. H. Chatoui and O. Ata, "Automated Evaluation of the Virtual Assistant in Bleu and Rouge Scores," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey, 2021.
9. V. Dalal and L. Malik, "A Survey of Extractive and Abstractive Text Summarization Techniques," in *2013 6th international conference on emerging trends in engineering and technology*, Nagpur, India, 2013.
10. K. Gowsic, M. M. Thanveer, P. N. M. Ajmal, A. Nandhagopal, and N. C. Thomas, "Long Short-Term Memory Networks for Text Summarization using NLP and ML," in *2025 3rd International Conference on Sustainable Computing and Data Communication Systems*, Erode, India, 2025.
11. R. Hemalatha, A. Karunasri, and P. V. V. Durga, "Natural Language Processing for Automated Text Summarization," in *2024 Asian Conference on Intelligent Technologies (ACOIT)*, Kolar, India, 2024.
12. S. Imtiaz and E. K. Hashi, "Analyzing the performance of sentiment analysis using BERT, DistilBERT, RoBERTa and GPT-2," in *2024 6th International Conference on Sustainable Technologies for Indusry 5.0 (STI)*, Narayanganj, Bangladesh, 2024.
13. S. Jain, S. K. Jain, and S. Vasal, "An effective TF-IDF model to improve the text classification performance," in *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, Jabalpur, India, 2024.
14. S. Kumar, S. Kanshal, V. Sharma, and R. Talwar, "Sentiment analysis on amazon alexa reviews using nlp & classification," in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2022.
15. W. Kwon, "Reading customers' minds through textual big data: Challenges, practical guidelines, and proposals," *International Journal of Hospitality Management*, vol. 111, no. 5, p. 103473, 2023.
16. X. Li, P. Wu, C. Zou, H. Xie, and F. L. Wang, "Sentiment lossless summarization," *Knowledge-Based Systems*, vol. 227, no. 9, p. 107170, 2021.
17. Z. Lin and C. Cho, "Lightweight Knowledge Distillation-Based Surrogate Model for Wheel–Rail Dynamic Contact Force Prediction using Hybrid CNN–LSTM–Transformer Networks," *IEEE Sensors Journal*, vol. 25, no. 10, pp. 17239–17251, 2025.
18. W. Luo, "Research and implementation of text topic classification based on text CNN," in *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, Changchun, China, 2022.
19. S. Mohammadi and M. Chapon, "Investigating the Performance of Fine-tuned Text Classification Models Based-on

Bert," in *2020 IEEE 22nd International conference on high performance computing and communications; IEEE 18th International conference on Smart city; IEEE 6th International conference on data science and systems (HPCC/SmartCity/DSS)*, Yanuca Island, Cuvu, Fiji, 2020.
20. M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir, "A survey of automatic text summarization: Progress, process and challenges," *IEEE Access*, vol. 9, no. 11, pp. 156043–156070, 2021.
21. C. Musto, G. Rossiello, M. De Gemmis, P. Lops, and G. Semeraro, "Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations," in *Proceedings of the 13th ACM conference on recommender systems*, Copenhagen, Denmark, 2019.
22. T. P. Nagarhalli, V. Vaze, and N. K. Rana, "Impact of machine learning in natural language processing: A review," in *2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)*, Tirunelveli, India, 2021.
23. A. Pisote, S. Mangate, Y. Tarde, H. A. Inamdar, S. A. Nangare, and V. Borate, "A Comparative Study of ML and NLP Models with Sentimental Analysis," in *2025 International Conference on Advancements in Power, Communication and Intelligent Systems (APCI)*, Kannur, India, 2025.
24. S. Ramaswamy and N. DeClerck, "Customer Perception Analysis Using Deep Learning and NLP," *Procedia Computer Science*, vol. 140, no. 10, pp. 170–178, 2018.
25. S. Singla, P. Priyanshu, A. Thakur, A. Swami, U. Sawarn, and P. Singla, "Advancements in natural language processing: Bert and transformer-based models for text understanding," in *2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech)*, Sonipat, India, 2024.
26. C. Toraman, E. H. Yilmaz, F. Şahinuç, and O. Ozcelik, "Impact of tokenization on language models: An analysis for Turkish," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 4, pp. 1–21, 2023.
27. M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 5731–5780, 2022.
28. K. Z. K. Zhang, S. J. Zhao, C. M. K. Cheung, and M. K. O. Lee, "Examining the influence of online reviews on consumers' decision-making: A heuristic–systematic model," *Decision support systems*, vol. 67, no. 11, pp.78–89, 2014.