Sentimental Analysis of Customer Product Reviews to Understand Customer Needs Using Machine Learning

by

Md Abdullah Al Symum
18201007
Subarna Yeasmin Sheemu
19101297
Abu Saleh Md. Asif
19301125
Konika Islam
17201007

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
May 2023

© 2023. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md Abdullah Al Symum
18201007

Subarna Yeasmin Sheemu
19101297

Abu Saleh Md. Asif
19301125

Konika Islam
17201007

Approval

The thesis titled "Sentimental Analysis of Customer Product Reviews to Understand Customer Needs Using Machine Learning" submitted by

1. Md Abdullah Al Symum(18201007)
2. Subarna Yeasmin Sheemu(19101297)
3. Abu Saleh Md. Asif(19301125)
4. Konika Islam(17201007)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 29, 2023.

Examining Committee:

Supervisor:
(Member)

Arif Shakil
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

People are influencing aspects of the digital world through machines. As a result, it is crucial to upgrade and use this aspect to do so. In the past, people used written letters to provide feedback. However, people are now posting these reviews to the seller's page directly on the internet.In the digital age, user feedback, and reviews have a significant impact on shaping businesses. However, it is challenging to analyze and understand the sentiments conveyed owing to the large volume of data and the presence of spam.If we can develop automated systems that can interpret sentiments of people and emotions from user reviews, which would help to leave a great impact on improving their marketing strategies and can understand the requirements of customer. However, machines are constrained by binary language, and, thus faces difficulties in comprehending human emotions and thoughts.By leveraging machine learning algorithms for sentiment analysis, we aim to evaluate sentiment in a vast collection of customer reviews. Sentiment analysis is an essential domain in machine learning and natural language processing, which focuses on identifying and classifying sentiments, opinions, and emotions expressed in textual data. This paper presents a comprehensive overview of sentiment analysis within the framework of machine learning approaches. For sentiment analysis, a wide variety of machine learning techniques and methods have been studied, including more established methods like deep learning models such as Convolutional Neural Networks (CNNs) and Transformers like BERT as well as traditional approaches like Naive Bayes and linear Support Vector Machines (SVM), KNN, and logistic regression. . The paper also addresses the challenges associated with sentimental analysis, such as data preprocessing, extracting features, and selection of models. Furthermore, it emphasizes the significance of labeled data and underscores the role of sentiment lexicons and word embeddings in improving sentiment analysis performance. The paper concludes by discussing the prospects of sentiment analysis in machine learning, highlighting its significance in social media analysis, customer feedback analysis, and market research. Therefore, the research outcomes of our paper provide valuable insights for companies that would enable them to enhance their marketing strategies and improve their products to meet customer requirements more effectively based on the evaluation of customer reviews and feedback.

Keywords: Machine learning, Natural language processing, Customer feedback, Textual data analysis, Sentimental analysis, Transformer, sentiment lexicons, Word embeddings.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis has been completed without any major interruption. Secondly, to our co-advisor Arif Shakil sir for his kind support and advice in our work. He helped us whenever we needed help. And finally to our parents without their throughout support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent developments in consumer sentiment analysis have focused on analyzing the ever-increasing amounts of consumer feedback data available in the form of online evaluations. As stated by Sanchez-Rada and Iglesias (2019), sentiment analysis is now increasingly accepted among scholars in addition to corporations, governments, and other organizations. [29] For practical applications, sentiment analysis based on online reviews is gaining widespread acceptance. Consumer behavior analysis, decision-making, and gathering valuable information for organizational growth are a few examples of these uses. Customers' reliance on opinion-based internet reviews has dramatically expanded due to the gradual transition from offline to digital markets.

Opinion mining and emotional intelligence are both terms that are interchangeably used with sentiment analysis [18]. Sentiment analysis gathers insightful information from unstructured language collected from various internet sources, including blogs, Twitter, WhatsApp, Facebook, and user comments. DM and text classification tasks involve computationally analyzing a consumer's sentiments, views, and attitudes toward services or products.

Opinion mining, or sentiment analysis, is a prevalent topic for researchers and industry, as it is a constantly growing sector with a trillion of data from all over the world. Sentimental analysis is a field within natural language processing or NLP [14]. Knowing better about an aspect or sentiment of a customer toward a product or other thing is crucial for research and industry. In this sector, we can see the variation in people's sentiments. In online reviews, people tend to show their comments in their native language or slang, including the aspect's polarity. For example, "it was a terrible experience," which we can detect as a negative sentiment, but "the quality was not mentionable," from this aspect, it is hard to identify as a negative or positive sentiment.

With advancing technology, people are more comfortable using emojis and slang, which are more effective in expressing sentiment than words. Emojis make the gathering of emotional responses an easy and enjoyable process. [28] It is a more fun way for users to give feedback about a particular product or express their opinions. The most common and universal emotions are happiness, sadness, disgust, surprise, and fear. If an efficient method can be introduced to detect these emoji

automatically, it will make a striking improvement to this sector.

Moreover, sarcasm, polysemy, and negation are other problems in this sector. Sarcasm is a fun way to give feedback on a product in a sarcastic way, which tends to attract the evaluator more interestingly. Talking about a product or a feedback or comment in an opposite way, such as "The product was so good that I have to throw it outside." In this, the classification algorithm will address this as a good review. However, this aspect is shown as a negative review as the word good is mentioned, and the result of the classification data will be brought down. Classifying tweets into sarcastic or non-sarcastic classes can be further enhanced by performing analysis based on frequency, Written-Spoken, intensity, structure, sentiments, synonyms, and ambiguity. [16]

Our study will mainly focus on sentimental analysis based on customer reviews and use different models to identify the best model for our dataset. Meanwhile, we will discuss different aspects of the models used in this research.

1.1 Research Problem

Sentiment analysis is a fast-evolving machine learning technique used in customer feedback analysis to classify and analyze human thoughts, feelings, attitudes, and opinions expressed in text, star ratings, and thumbs-up and thumbs-down comments about items. Many goods are now offered for sale online due to the rapid development of e-commerce, and a rising number of consumers are making their purchases through online marketplaces like Amazon, Aliexpress, etc. Consumers frequently give comments on products they have recently purchased through reviews. Customers' dependence on e-commerce companies, which conduct most of their business online, highlights how widespread Internet use has become. Since users can write reviews of a wide range of goods and services on these websites, an astronomically vast number of reviews are available online. This has raised the need to examine these reviews to comprehend customer feedback. The ability to synthesize customer sentiment and find company improvement opportunities is made possible through sentiment analysis. The consumer insights function is an effective tool that identifies what works well and what needs to be fixed immediately. In the end, it also makes it possible to concentrate on the business's most important and influential areas to improve the client experience, resulting in decisions that boost client loyalty and happiness.

We now utilize emotion to distinguish between a robot and a human since a robot or machine lacks emotion. However, it is exceedingly challenging to deduce from a robot the characteristics of emotional behavior in people. Specifically, our mood and emotional state are how we communicate our emotional content. As a result, requesting something can indicate a lot of different emotions in addition to what someone is asking. We are perplexed by how important it might be to a binary machine. Thus, we employ sentimental analysis, divided into three categories: document-level, sentence-level, and aspect-based. Aspect-based sentimental analysis, or AbSA, is now more effective because it explicitly states the sentence and tries to grasp the aspect before summarizing all of the aspects and using it on documents

and other things.

The issue with AbSA is that, despite being more beneficial than previous SA procedures, it is still challenging for a machine to comprehend the components we are attempting to describe. We have divided the entire AbSA method into three categories: abstract extraction, aspect sentimental analysis, and sentimental evaluation. However, since it might be difficult to classify or distinguish between an aspect and an opinion, aspect extraction still needs to be solved. Hu and Liu's [4] first goal was to extract all common phrases from reviews; however, they eventually expanded it to incorporate abstract and opinion extraction. Eirinaki suggested AskUs, a mining search engine that is based on opinions, as a way to improve the current scenario [11]. This index is intended to rank and prioritize the information according to user-provided factors and viewpoints. Additionally, the aspect extraction is divided into three groups: implicit aspects, explicit aspects, and opinion-focused expressions. There is still much research being done, and everyone working on it is attempting to improve how well and precisely machines can grasp emotions [10].

Aspect sentiment analysis (ASA), which assigns a sentimental score and improves the sentimental classification accuracy, is performed in AbSA's second phase. However, this poses a special challenge because assessing can often be more challenging.

Handling sentences with several components becomes more challenging when conjunction is included. This was demonstrated by Wang et al. [20], who suggested an organized aspect-specific attention network for enhanced sentimental classification. In order to handle both general and target-specific sentiment, he proposed six separate target-sensitive recurrent attention memory networks to capture the relation between the aspects (targets) and the attitudes associated with their surroundings. The last section of AbSA is Sentiment Evolution, which discusses how sentiments evolve. A general agreement is challenging to develop during a period of emotional shift. The fundamental problem at this phase is identifying the elements that affect people's decisions to acquire a strong sentiment and alter their cognitive attitude. According to Fu and Wang [3], SE generally follows the rule of majority adaptation and minority avoidance. As a result, it can be challenging to identify how our cultural, social, and personal preferences influence how we interpret emotions. Regardless of the issue, we must improve the scenario for machines to comprehend the sentiment. Some ML algorithms and search engines will make things simpler for the binary machine.

1.2 Research Objective

Computer research techniques such as sentiment analysis glean subjective data from text. One may use sentiment analytics to determine what consumers think of their products and brand. The objective is to create a system enabling consumers to share their experiences and give prospective buyers accurate information about the product's functioning. The objectives of this research are now described below:

1. To deeply understand the psychology of customers.
2. To establish a direct relationship between business and customers.
3. To get relevant information to upgrade.
4. To categorize and determine the polarity of web data.
5. To improve the current method or methodology to get significantly better outcomes.
6. For enhanced customer experience, sentiment analysis may identify client views regarding services, goods, and campaigns and gauge their mood and tone.

1.3 Research Motivation

Sentimental analysis is essential for customers and producers and provides a precious "customer opinion" about products and services. As the world is going so fast and the Internet is available to everyone, now is a time to give reviews and post anything positive or negative on the Internet. Customers can now communicate their opinions, feelings, and sentiments about products online. Thus, opinion-rich textual data are now available in abundance on the Internet. Sentiment analysis uses this data to gather vital information about client opinions. Sentimental analysis in customer product reviews is valuable since it helps businesses dig into the customer's minds and determine their satisfaction level. By analyzing sentimental analysis, businesses learn how customers accept their product and their valuable behavior toward it. Therefore, businesses can acquire valuable customer reactions by pinpointing positive and negative reviews. Then, businesses go forward to upgrade the standard of the product and earn customer satisfaction. Sentimental analysis helps to find out the weaknesses and strengths of the product. Based on customer reviews, businesses find out the major weakness of the product. By utilizing information, businesses take relevant action to develop the product. This information will result in customer fulfillment and loyalty. The sentimental analysis also helps businesses monitor the product for improvement. Overall, the sentimental analysis will permit businesses to make proper strategies and well-informed commitments based on customers' attitudes and sentiments towards products.

1.4 Research Outline

Our study's significant goal is to improve customer loyalty and retention through enhanced customer experiences. The technique used to obtain it is sentimental analysis. Additionally, we have annotated the data to give an overview.
Chapter 1: explains the goals of our study and the inspiration behind continuing our work. This section also explains the steps taken to conduct a problem-solving investigation.
Chapter 2: This section includes related work parts that skim the published topic that, strictly speaking, corresponds to our work.
Chapter 3: The dataset that we used for our research has been covered in this part.

We discuss how the dataset was created and how our models are connected.
Chapter 4: This chapter provides a detailed explanation of each architecture that we used in our paper. Then we gave those models illustrations.
Chapter 5: This chapter provides the architecture and evaluation procedures details before we look at and evaluate the data.
Chapter 6: The Future Work provides an idea of what we can accomplish using our research and the features we developed. Moreover, the conclusion brings the paper to a close by discussing a few distant research job preferences.

Chapter 2

Literature Review

From a research perspective, sentiment analysis of customer product reviews represents an exciting frontier in data-driven decision-making. By harnessing the power of advanced computational techniques, researchers can delve into the rich tapestry of customer sentiments, unraveling valuable insights that were once hidden beneath vast amounts of textual data. This research perspective unlocks a world of possibilities, enabling scholars to explore new avenues, uncover hidden patterns, and gain a profound understanding of customer needs and preferences. At the core of sentiment analysis lies the ability to accurately decipher the sentiments, emotions, and opinions expressed by customers. The strength of sentiment analysis lies in its ability to extract emotions and its capacity to analyze and categorize vast amounts of customer feedback at an unprecedented scale.

2.1 Related Works

In the paper [40], building upon recent advancements in neural networks, the author introduces a deep learning-based sentiment analysis model named lexicon integrated two-channel CNN-LSTM family models, which combine CNN and LSTM/BiLSTM branches in parallel. The long short-term memory (LSTM) model and convolutional neural network (CNN) in [13] have gained significant attention in the field of sentimental analysis due to their high performances compared to Machine learning approaches. To be particular, machine learning approaches such as support vector machines (SVM) aims to find a hyperplane that maximizes the margin between positive and negative samples in sentiment analysis [15]. The paper discusses using LSTM and CNN models for sentiment analysis and emphasizes that their performance depends on determining the amount and quality of labeled data [40]. They have introduced the lexicon-integrated two-channel CNN-BiLSTM model, which includes two key contributions: a parallel two-channel structure and sentiment padding. Sentiment padding addresses the issue of gradient vanishing between layers and improves the representation of sentiment information. The model achieves improved performance in sentiment analysis by combining CNN for local feature extraction and BiLSTM for long sequence processing. According to this paper [40], the CNN model excels at extracting local patterns and features from the input data. In contrast, the BiLSTM model is adept at processing dependencies and contextual information. The combination of these two models incorporates a well-designed loss function and achieves improved performance on SST and reversed SST datasets. The LSTM

model [9] incorporates textual order and sequence length into consideration, which enables it to capture the temporal dependencies and context within the text. In contrast, the CNN [7] focuses on extracting local features within the text to identify patterns and essential information at a smaller scale. All the models need effective word embedding, but existing methods can only partially solve the embedding problem for sentimental analysis. Besides, the design of neural network architecture needs to have theoretic guidance. Additionally, the paper [32] focuses on a tree-structured regional CNN-LSTM model, which consists of two parts: one is regional CNN, and the other one is LSTM, which helps to predict the valence-arousal (VA) ratings of texts. The proposed model separates text into regions, weighs their affective information, and uses LSTM to integrate these weighted inputs for accurate prediction—incorporating structural information through region division significantly improves performance. The proposed regional CNN differs from a conventional CNN by utilizing specific text regions as input, dividing it into multiple regions to extract and weigh relevant affective information from each region for accurate prediction in the context of VA [32]. The paper proposes a unified architecture that integrates bidirectional LSTM (BiLSTM), attention mechanism, and the convolutional layer to tackle the complexities of high dimensionality, complex semantics, and sparsity in text classification [27].

The utilization of word and phrase embeddings has consistently been proven to enhance improvements in its performance across various natural language processing (NLP) tasks when employed as the input representation within a learning system [6]. Initially, convolutional neural networks (CNNs) were employed to reduce the dimensionality of the feature space and extract meaningful features from text [41]. In a basic CNN architecture, the convolutional operation is applied to local n-gram tensors [21] using a particular set of kernels. Subsequently, max pooling is applied to reduce the output size from the preceding convolutional layer [32], effectively reducing computational complexity and streamlining the overall analysis process. Jianqiang et al. [19] employed contextual semantic features and co-occurrence statistical features of words in tweets, utilizing an n-gram feature input convolutional neural network to analyze sentiment polarity. The paper [42] combines sentiment lexicon with Convolutional Neural Network (CNN) and attention-based Bidirectional Gated Recurrent Unit (BiGRU). This proposed method enhances sentiment features and integrates text context features by analyzing the factor like analyzes factors like thesaurus size, sentence length, and model iterations to optimize the performance of the sentimental analysis. This model enhances sentiment features in the input text using a sentiment dictionary to weight sentiment words, employing CNN to extract essential features from the input matrix. At the same time, BiGRU considers order information and extracts context features. The attention mechanism assigns weights, highlighting sentiment features [42]. This integration demonstrates that the deep learning model (CNN and BiGRU) achieves significantly superior classification performance in the sentimental analysis compared to the machine learning model (NB and SVM) [42]. Hyun et al. [26] introduced a target-dependent convolutional neural network that considers the distance relationship between target words and surrounding words to capture the influence of context on target words. Attention mechanisms emerge as each word in a sentence plays a distinct role in determining the emotional polarity. BiLSTM and attention mechanisms exhibit more significant

impacts on the classification accuracy compared to the convolutional layer [27]. Li et al. [21] presented a joint structure that combines CNN and RNN. The structure utilizes RNN to guide CNN's positioning and incorporates a global average pooling layer to capture long-term dependencies in conjunction with CNN. When combined, CNNs and LSTMs complement each other's strengths and offer a more comprehensive representation of sequential data. CNNs can efficiently capture local patterns and generate meaningful features from the text, which can then be fed into LSTMs to model the long-term dependencies and contextual information [27]. Xu et al. [12] developed a robust method called the "divide-and-conquer" approach for sentiment analysis. They used a neural network-based model to classify sentences and then applied a convolutional neural network to classify sets of sentences. This combination of techniques allowed them to accurately determine the sentiment in text, capturing the intricate details of language with great accuracy.

Traditional classification methods (e.g., SVM, LDA, Naïve Bayes) based on linguistic features such as n-grams, POS tags, and lexical features suffer from sparse and high-dimensional feature space and labor-intensive feature engineering drawbacks: sparse and high-dimensional feature space and labor-intensive feature engineering [43]. To overcome these drawbacks, the author [43] proposed an Attention-based Bidirectional CNN-RNN Deep Model (ABCDM), primarily focusing on polarity detection in sentiment analysis at the document level. This model leverages publicly accessible pre-trained GloVe word embedding vectors as the initial weights for the embedding layer, ensuring a rich foundation for the model's representation of words. CNNs were employed to reduce the dimensionality of the feature space and extract relevant features from text [41]. Word embedding is crucial in sentiment analysis as it captures semantic relationships, represents words in a dense vector space, and enhances the understanding of sentiment by leveraging contextual information. In a recent study by Liu and Guo [27], they introduced word embedding AC-BiLSTM. This model combines bidirectional LSTM and CNN networks with an attention mechanism for sentiment analysis and question-answering. AC-BiLSTM employs CNN on the word embedding layer, followed by BiLSTM to capture long dependencies. The attention mechanism is then applied to emphasize crucial areas of the text.

In a significant contribution by the authors in [38], they introduced RTAnews, a benchmark dataset of multi-label Arabic news articles for text categorization. They extensively evaluated various state-of-the-art multi-label learning algorithms for Arabic text categorization on RTAnews. The evaluation involved popular algorithms such as binary relevance, classifier chains, calibrated ranking, SVM, k-nearest neighbors (KNN), random forest, and adaptation-based algorithms. The results highlight the superior speed and effectiveness of adaptation-based algorithms compared to transformation-based algorithms. SVM faces limitations in parameter selection, algorithmic complexity, multiclass data sets, and imbalanced data sets, with reduced popularity for large data sets due to long training times and poor accuracy for imbalanced data **cervantes20comprehensive**. However, the paper [39] introduces a novel approach that enhances the support vector machine's kernel function for text sentiment analysis. By leveraging probabilistic latent semantic analysis, SVM's proposed Fisher kernel function captures the probability and latent relationships among text, vocabulary, and subject. This improved kernel function

effectively addresses the issue of ignoring latent semantic features, resulting in an average accuracy of 87.20% on the Twitter sentiment corpus. The study [37] also investigates the predictive ability of SVM models for sentiment in future e-mail responses. The results show that while more accurate than extracting e-mail sentiment, it is still possible to predict the sentiment of the following e-mail in the thread. The Linear SVM algorithm achieves a mean F1-score of 0.688 and a mean AUC of 0.805, suggesting a predictable pattern in e-mail conversations for anticipating the polarity of upcoming e-mails.

The paper [42] proposes SLCABG, a new sentiment analysis model that combines sentiment lexicon, CNN, and attention-based BiGRU. SLCABG overcomes the limitations of existing models by leveraging sentiment lexicon and deep learning techniques. It enhances sentiment features using the lexicon, extracts primary sentiment and context features with CNN and BiGRU, and employs an attention mechanism for feature weighting. The model is trained and tested on actual book evaluations from a Chinese e-commerce website.

In a recent study by Mahudeswaran et al., [30], a groundbreaking approach was introduced to detect false or negative news using Naive Bayes and Random Forest. The approach described in the study utilizes the tf-idf Vectorizer along with the cosine similarity method for tokenizing a collection of text documents and constructing a vocabulary of existing words. This technique allows for efficient processing and analysis of text data, enhancing the overall performance and accuracy of the system. In the paper [30], the author tried to analyze people's sentiments in a political context from Twitter data through advanced text processing and employing the Naive Bayes method. The proposed result showcases the superiority of the Naive Bayes method with an impressive accuracy rate of 80.90%, surpassing KNN (75.58%) and SVM (63.99%).

BERT, a pre-trained language model, has surpassed previous benchmarks in eleven NLP tasks, including sentence-level sentiment classification. Liu et al. [24] investigate using customer reviews as a valuable source of knowledge for answering user questions. They introduce the Review Reading Comprehension (RRC) task and propose a post-training approach that fine-tunes the BERT network to enhance performance. They also transform Aspect Sentiment Classification (ASC) into a Machine Reading Comprehension (MRC) problem, focusing on the polarity of specific aspects.TD-BERT incorporates target information into BERT models, resulting in state-of-the-art performance in aspect-level sentiment classification onSemEval-2014 and a Twitter dataset [35]. Including target information consistently improves accuracy, highlighting its crucial role in enhancing BERT's performance.

Ref	Task	Classifier	Dataset	Accuracy
[25]	Analyzing Sentiment classification towards Specific Entities	Target-Dependent BERT (TD-BERT) BERT-pair-QA-M	2 datasets: SemEval-2014 and Twitter Dataset	83.09% and 80.27% for TD-BERT and BERT-pair-QA-M respectively
[35]	Extracting Valuable User Experiences for Answering User Questions	transformer-based language model-BERT fine-tuned with NLP	RRC dataset and SQuAD	BERT-MRC: 81.06% BERT-PT: 84.26%
[34]	Comparison of Naïve Bayes, SVM and KNN regarding political sentiment	Naïve bayes, svm and K-Nearest Neighbor (K-NN)	Crawler data from Twitter	naïve bayes: 75.58%, svm; 63.99% and K-NN: 73.34%.
[12]	Categorizing and calssifying target based opinion	BiLSTM with conditional random fields and 1D-CNN	MPQA opinion corpus v2.0	BiLSTM-CRF detects opinion targets better than CRF
[37]	Predicting sentimental response of customer regarding customer support through email	SMV , Dummy and Random Classifier	Dataset from Swedish telecom company	F1-score of 83% and mean AUC of 89%
[39]	Probabilistic Latent Semantic Analysis for text classification characteristics	SVM classifier	Dataset collected from twitter	FK-SVM accuracy is 87.20%, recall rate is 88.30%
[27]	Extracting semantic relationships and meanings of words based on sentiments	SoftMax Classifier, MULTI task LSTM, CNN multi-channel	IMDb and RT-2k	BLSTM: 87.9% for IDMB and 87.2% for RT-2k.
[43]	Sentiment polarity detection for ong review and short tweet s	Conditional Random Fields	Kindle, CDs and Vinyl dataset, T4SA, Sentiment140	96%, 87%, 84.3%, 52.7% and 8.1% respectively for datasets

Ref	Task	Classifier	Dataset	Accuracy
[21]	classification of sentiments on large dataseet	Bidirectional Gated Recurrent Units with convolutional neural network	Stanford Twitter Sentiment Corpus dataset	GloVe-Bi-GRU-CNN : 87.58%, LSTMCNN : 86.10%
[26]	Target dependent text classification	Temporal Convolutional Layer	Real-world twitter datasets	Macro-F1 score of $0.491 \pm 0.0073\%$
[40]	Extracting negative and positive opinion for review based sentiments	Attention-Based LSTM Classifier and Convolutional Neural Networks (CNN) Classifier	nSST dataset, reversed SST dataset, and Chinese tourism review dataset	94.63% for Word2vec

Table 2.1: Summary of all papers related to our research

Chapter 3

Dataset

In this section, we have discussed the dataset used for our research purposes. We want to elaborate on the whole process, from collecting data to annotating and pre-processing the data for implementation in our model.

The sentimental analysis includes the CSV type of data, which is primarily text and has the format to allow the dataset to be in table format. So for our research, we have structured the dataset in a CSV file and collected the data from an online platform. We will go through the step-by-step process below to give a decent idea of our dataset.

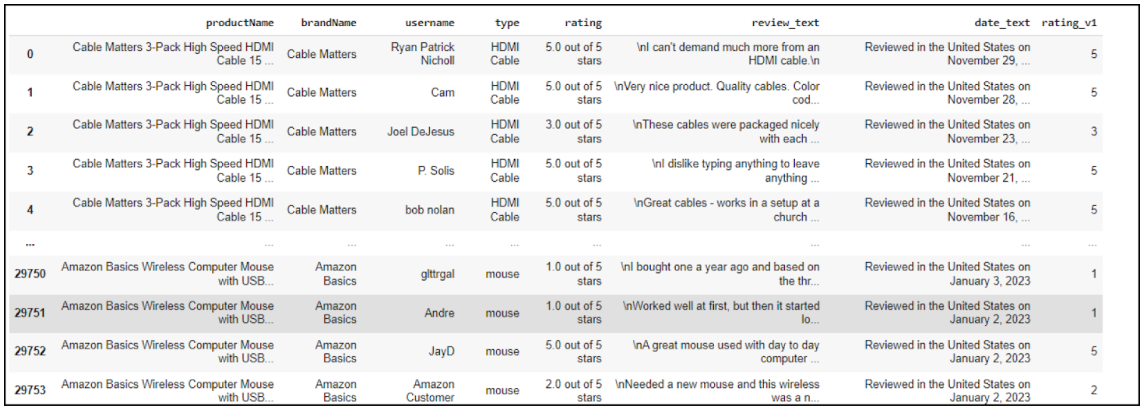
3.1 Data Collection

Our primary concern was to build a complete, unique dataset. So for sentiment analysis of customer product reviews, we go through many items we can search for on the internet. As in this sector of sentiment analysis, there can be thousands of products and categories. Such as movie ratings, food reviews, product reviews, etc. So after discussing it with ourselves, we decided to look at the product reviews in Amazon's computer accessories and peripherals category.

After deciding the category, we started web scraping the customer's reviews using Beautiful Soup, a Python library that allows us to parse and scrape the HTML and XML pages. We have decided to maintain the user name, user ID, user reviews, user rating, and the date of the reviews. We wanted to maintain the most recent data in our dataset, so that was our biggest concern with dates. Therefore, we gathered information after 2021 and not before. In order to acquire a total of 200 reviews from each page, we scraped the reviews from 20 pages of each product category. In this manner, we gathered 36,792 items from the 15 categories of products. However, there is substantial redundant data in the dataset. So we reduced the redundant data, and after losing all the redundant data, there were almost 29,755 rows of data in our dataset.

3.2 Data Sample

A sample of the collected data is given here in figure 3.1



	productName	brandName	username	type	rating	review_text	date_text	rating_v1
0	Cable Matters 3-Pack High Speed HDMI Cable 15 ...	Cable Matters	Ryan Patrick Nicholl	HDMI Cable	5.0 out of 5 stars	\nI can't demand much more from an HDMI cable.\n	Reviewed in the United States on November 29, ...	5
1	Cable Matters 3-Pack High Speed HDMI Cable 15 ...	Cable Matters	Cam	HDMI Cable	5.0 out of 5 stars	\nVery nice product. Quality cables. Color cod...	Reviewed in the United States on November 28, ...	5
2	Cable Matters 3-Pack High Speed HDMI Cable 15 ...	Cable Matters	Joel DeJesus	HDMI Cable	3.0 out of 5 stars	\nThese cables were packaged nicely with each ...	Reviewed in the United States on November 23, ...	3
3	Cable Matters 3-Pack High Speed HDMI Cable 15 ...	Cable Matters	P. Solis	HDMI Cable	5.0 out of 5 stars	\nI dislike typing anything to leave anything ...	Reviewed in the United States on November 21, ...	5
4	Cable Matters 3-Pack High Speed HDMI Cable 15 ...	Cable Matters	bob nolan	HDMI Cable	5.0 out of 5 stars	\nGreat cables - works in a setup at a church ...	Reviewed in the United States on November 16, ...	5
...	...	...	...	...	...	...	...	...
29750	Amazon Basics Wireless Computer Mouse with USB...	Amazon Basics	glttrgal	mouse	1.0 out of 5 stars	\nI bought one a year ago and based on the thr...	Reviewed in the United States on January 3, 2023	1
29751	Amazon Basics Wireless Computer Mouse with USB...	Amazon Basics	Andre	mouse	1.0 out of 5 stars	\nWorked well at first, but then it started lo...	Reviewed in the United States on January 2, 2023	1
29752	Amazon Basics Wireless Computer Mouse with USB...	Amazon Basics	JayD	mouse	5.0 out of 5 stars	\nA great mouse used with day to day computer ...	Reviewed in the United States on January 2, 2023	5
29753	Amazon Basics Wireless Computer Mouse with USB...	Amazon Basics	Amazon Customer	mouse	2.0 out of 5 stars	\nNeeded a new mouse and this wireless was a n...	Reviewed in the United States on January 2, 2023	2

Figure 3.1: Data Sample

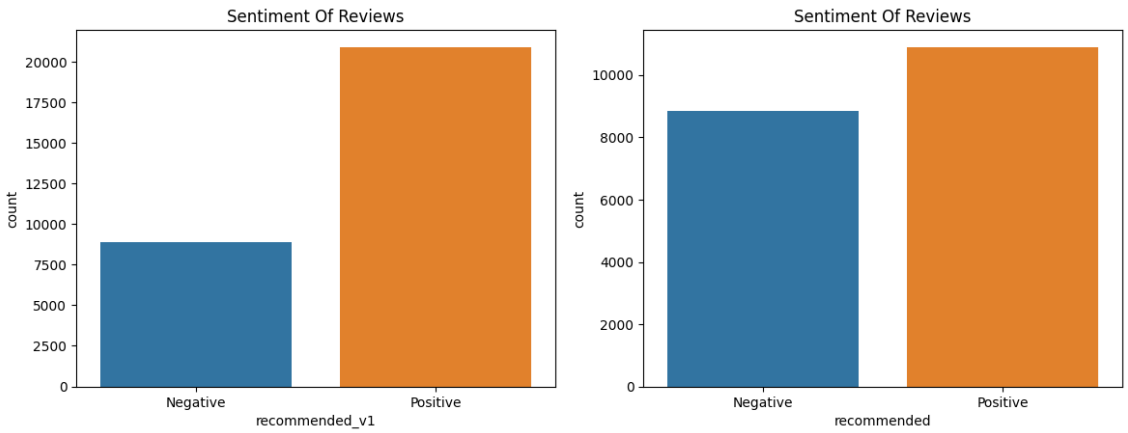
We can see the collected data, where the columns are the product name, brand name, user name, type, rating review, text-date, text, and rating.

3.3 Data Description

After data collection, our dataset had roughly 30,000 records from 15 categories. In order to keep the dataset unique and usable for future research, we wanted to keep the most recent data. The user name, user id, user location, product name, product category, user reviews, and user rating were all retained in our dataset. The reviews on Amazon were used to determine the user rating, which ranged from 1 to 5. The category included several product types, including HDMI cables, mouse, keyboards, headsets, flash drives, etc. The brand name and product code were also included in the product name. The user ID was fetched to handle the duplicate data more efficiently. Overall, our dataset has a large number of miscellaneous items.

3.4 Data Annotation

To keep the data unique, we have annotated the data ourselves. Though it was a hassle job, it was fruitful after finishing it. However, before doing that, we dropped the duplicate data from our dataset and started to annotate it. We annotate the data based on positive and negative reviews. After our annotation, we found that there needed to be more consistency between positive and negative reviews. Therefore, we removed about 10,000 of our positive ratings, which constituted a sum of around 20,000 data reviews in our dataset, to manage this issue better. So the new sum of the data was perfect for our next step in preprocessing: a total of 19755 data reviews consisting of 10,898 positive reviews and 8,857 negative reviews.
This is the inconsistent dataset in figure 3.2a. The following figure, 3.2b, is the cleaned dataset after the annotation. We kept only the review text, labeled columns for the next step, and dropped the other column. With this, we moved forward to the next step in our preprocessing.



(a) Inconsistent Data (b) Consistent Data

Figure 3.2: Data Annotation

3.5 Data Pre-processing

We begin preprocessing the dataset for incorporation into the model we will use after acquiring and labeling the data. We start preprocessing the dataset because our study will use BERT, LSTM, RNN, KNN, Logistic Regression, and Naive Bayesian. The data preprocessing is different for the BERT because it automatically does some stages. The rest of the model, however, requires preprocessing.
Data preprocessing is a phase in the data mining and evaluation procedures that converts raw data into a structure that computers and machine learning algorithms can comprehend and evaluate. We will go through the following preprocessing phases to prepare the data for the model we use in this study.

3.5.1 Punctuation Removal

For text preprocessing, punctuation removal is often essential, as it will help treat each text equally. [44] For example, the words “how” and “how?” will be treated equally after removing the punctuation. In our dataset, as we collected our data from Amazon, there will be variants of words. If we do not remove the punctuation from our dataset, it will reduce the accuracy of the models like LSTM, KNN, RNN, and others we will use for our research purposes. So for that reason, we have removed the punctuation from the reviews and cleaned the data for further use.

3.5.2 Lower Casing the Data

Lower casing the data is an essential preprocessing step as well. Switching all words to lowercase will help the model determine the word more precisely. For example, the words ”Good,” ”GOOD,” ”good,” ”GooD,” and ”gOoD” will all be changed to ”good.” The model will be able to comprehend the word’s stem effectively in this way. The lower casing is vital, especially in certain circumstances, such as the vectorization and tokenization processes. We have finished lowercasing the review, given that we are heading to the next phase.

3.5.3 Tokennization of Data

As a following step, we created a tokenizer to turn the reviews into tokens after reducing the data to lower case and clearing any punctuation. The tokenization method involves breaking the text down into significant pieces. [22] These items are referred to as tokens. For instance, “I have a beautiful dog” will be converted to [I], [have], [29], [beautiful], [dog]. After that, it acts as input for our research model.

3.5.4 Stopword Removal

Next, we want to eliminate any filler or stop words. It is unnecessary for our analysis. In the case of text analysis, the most frequently used phrases in the English dictionary—and, the, be, too, and of—are eliminated. Therefore, both positive and negative reviews will mention this. These frequently used words will accomplish the grammatical requirement but provide little content for the machine or models. We also generated a word cloud to get a glimpse of the words most frequently used in positive and negative reviews. Due to the model’s inconsistencies caused by these words appearing in both reviews, the model’s prediction rate decreases.

(a) Positive Word Cloud (b) Negative Word Cloud

Figure 3.3: Word Cloud

Figures 3.3a and 3.3b indicate that the words drive, work, good, one, cable, headset, and bought were redundant or similar in both reviews, after the word clouds were generated. These words are going to act as stopwords for the model. Therefore, in addition to stopwords, we also deleted these words from the tokenized dataset.

3.5.5 Lemmatization of data

Lemmatization is the method that strips down all words to their original ”lemma” or ”root” form. We must understand these two terms, semantics and syntax, to understand them better. The structure of a phrase is referred to as syntax and sentence meaning is referred to as semantics. [22] Lemmatization, in particular, addresses the issue by grouping the terms involved in the difficulty. Text lemmatization improves the efficiency of algorithms by correctly classifying tokens at the cost of complexity. After our reviews were lemmatized, they were ideally suited for the model’s input.

3.5.6 Text Vectorization

The process of changing text into numerical representations is called text vectorization. Since text data cannot be applied directly for modeling, it has to be numerically processed. To understand the dataset, we must transform the text into a numerical form that a machine or model will recognize as input. [2] In order to accomplish our objectives, we utilized word2vec to vectorize the text using word embedding. Words are mapped to vectors with dimensions as their core technique. Due to the training process considering the current word's context, a low-dimensional dense vector with semantic information can be developed.

3.5.7 Data Classification

We divided the dataset into the train, test, and validation sets in an 8: 1: 1 ratio with a random state of 42, keeping the training size at 0.80, the test size at 0.10, and the validation size at 0.10.

Training Set

Any deep-learning model needs a training set, which is data collection. Eighty percent of the data in our dataset was used to train our model. Our model can produce more precise predictions when combined with a validation set when we give it more training data to train on. We used 15,804 pieces of data for our training set.

Testing Set

A testing set is a set of data that will be used to evaluate the performance of a deep learning model and predict how well the model will generalize to new, untried data. For the testing set, we chose data from 1974, representing around 10% of our dataset's total data.

Validation Set

A validation set is a set of data removed from the training set for creating a model and used to assess the model's efficacy. The model's hyperparameters—parameters defined before training the model and not accessible to learning from data—are tuned using the validation set. The best values for the hyperparameter are selected by considering the model's performance on the validation set. We used data from approximately 1,976 for the validation set or 10% of the whole dataset.

Chapter 4

Research Methodology

4.1 Working Progress

The flowchart for our thesis project is depicted in Figure 4.1 and includes the following steps: data collection, preprocessing, data splitting, architectural implementation, testing and validation, evaluation performance, and result analysis.
Data will first be gathered from the computer accessories and peripherals category on Amazon and via online scraping with the lovely Soup Python web scraping package. We manually labeled the data once it was collected and then preprocessed it so the machine could comprehend it. The information will then be split into training, testing, and validation sets. 8:1:1 will be the divisions. It will then be incorporated into the models. The models BERT, LSTM, Naive Bayes, Logistic Regression, Linear SVC, KNN, and Fitting Decision Tree Classifier will all be run. The testing will then be completed, and we will assess the models using the validation set and performance data. The results will then be examined, and the models will be contrasted.

Figure 4.1: Working Progress

4.2 Used Architectures

4.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformer architecture approach for text synthesis, sentiment analysis, and natural language processing. [5] The BERT model architecture is based on Transformers. It uses multilayer bidirectional transformer encoders for language representation. Z. Gao et al. [5] suggested that the BERT Model can be executed through two stages: the initial phase includes pretraining, wherein the model acknowledges its input data to understand language and context related to the data, and the following phase uses fine-tuning, in which at first the model receives and recognizes the best approach for a specific task. The objective of pretraining is to make BERT understand what the language is and what the context is. BERT is taught language by practicing two unattended tasks simultaneously: Masked Language Model (MLM) and Next Sentence Prediction (NSP). From MSM, BERT takes in a phrase with random phrases that include masks. The objective is to result in these mask tokens in order to help BERT comprehend the bi-directional context within a sentence. In the case of NSP, Bert requires two sentences and determines if the subsequent one follows the initial sentence in a case of binary classification. This way, BERT comprehends the context across various phrases. Using this, the NSP and MSM, BERT understand languages well. The MSM and NSP both work simultaneously to give BERT a better understanding of the language. The input that is provided is an array of two phrases with a portion of the words masked. Every token is a word converted into an embedding through pre-trained embeddings. On the result side, C is the binary outcome of the next sentence prediction(NSP). Each of the Ts corresponds to the mask language model.

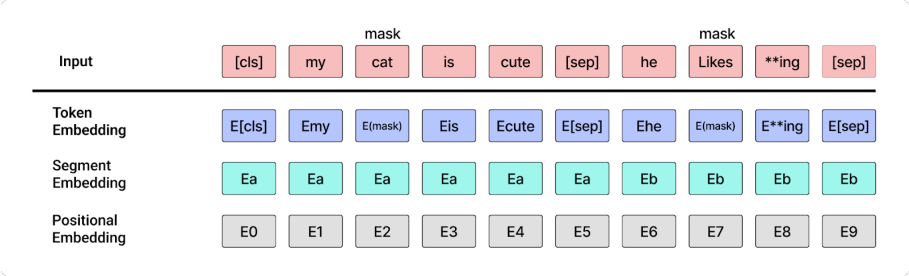


Figure 4.2: Figure of Embeddings

From Figure 4.2, we can see that the initial embedding of inputs begins with three vectors following tokens embeddings, segment embeddings, and position embeddings. The 30,000-token dictionary used in the token embeddings was pre-trained to be utilized with word piece embeddings. Positioning embedding entails encoding a word's position within a sentence compared to segment embedding, which involves encoding the sentence's number into a vector. These two embeddings preserved the ordering, as all these vectors will be fed into the BERT simultaneously, and the order of the token must be preserved for the language model. We get the embedding vector, which we implement as the BERT's input by incorporating all of them.
A simple pre-trained BERT is constructed with dropout regularization as well as a softmax classifier layer, as shown in 4.3. [33]. The word vectors are supplied

into a fully linked, layered output that has the exact same number of neurons as the vocabulary tokens. As a result, the word vector has to go through the corresponding 30,000 neurons in the output layer. This word vector will be converted into a distribution using a single hot-encoded vector for the word itself. After that, the resulting distribution will be correlated with the softmax layer, and then the neural network will be trained through the cross entropy loss. Only the mask word will be counted when calculating cross-entropy loss, and the remaining components will be dropped down so that it can only focus on the word's context.

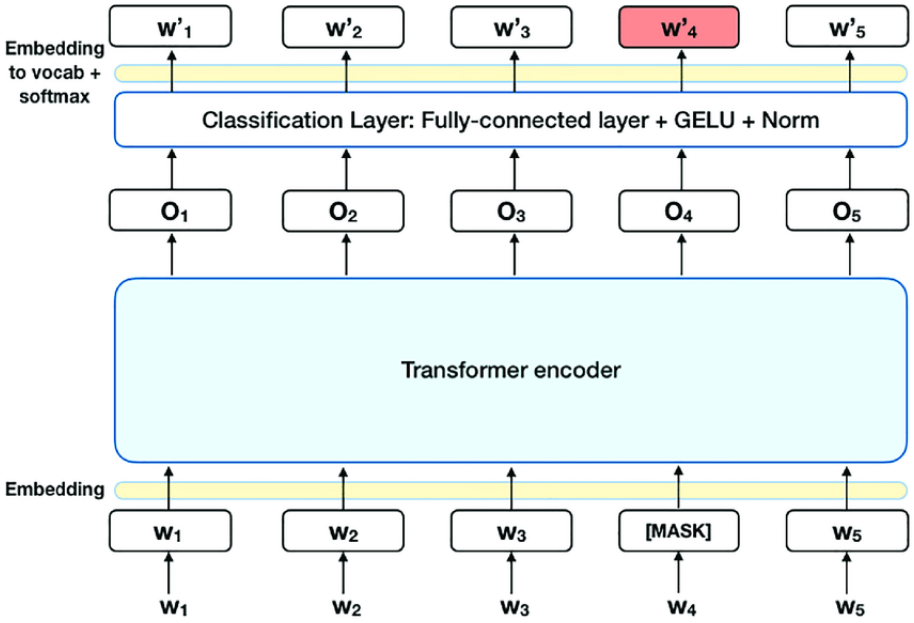


Figure 4.3: Figure of Bert Architecture

4.2.2 LSTM

LSTM, known as long short-term memory, is proposed by "Hochreiter and Schmidhuber" (1997) [1]. RNN architecture often includes cyclic interactions that allow the RNN to modify its present state based on prior states and the input data used at the time. [36], But because of the enormous distance between the pertinent input data, the RNN cannot link the pertinent data. As a result, LSTM was put forth, demonstrated the ability to provide intriguing results, and became well-liked for deep learning.

From Figure 4.4, let us consider vanilla recurrent neural networks, change all the hidden units with the LSTM cell, as well as add an additional connection from each cell, identified as the cell states. It is known as the LSTM. It was developed to solve the issue of disappearing and exploding gradient issues. [1] Aside from the hidden state vector, each LSTM cell additionally maintains a record of a cell state vector, and subsequent LSTM can decide whether to read straight from it or restart the cell using an explicit gating mechanism at each passing step. Every gate has a total of three gates of the same structure, known as the input, forget, and output gates. [8] The input gate decides whether to update the memory cell, the forget gate decides whether to clear the memory, and the output gate decides whether to display the

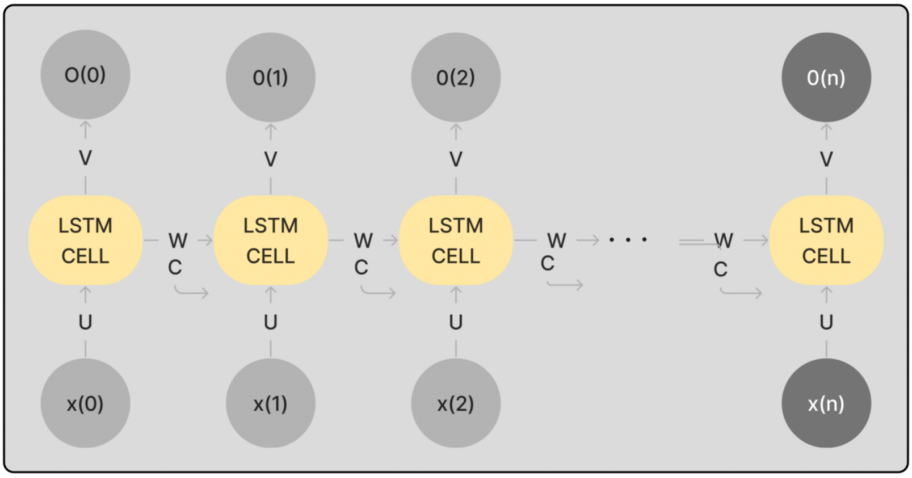


Figure 4.4: Vanilla recurrent neural networks

information about the current cell state.
In figure 4.5, the equation of the input, output, and forget gate goes as follows:

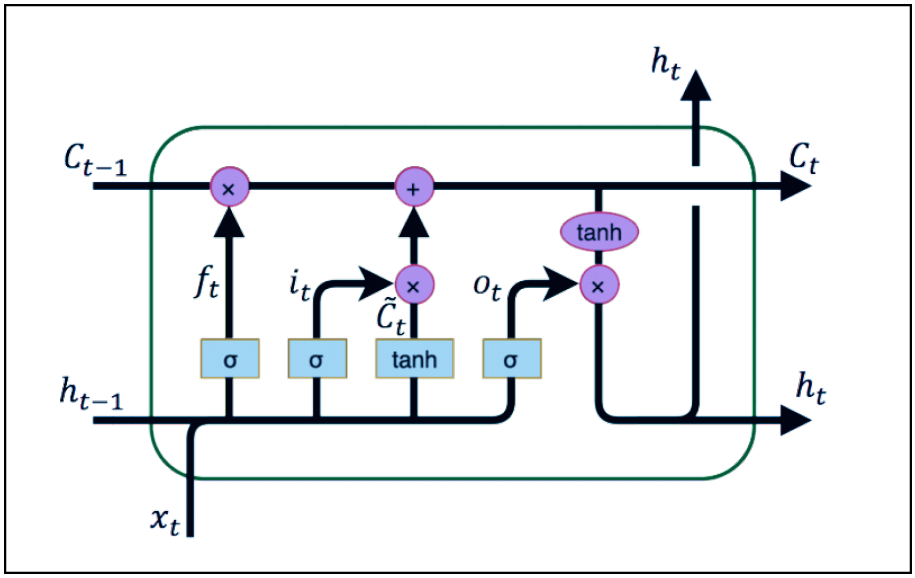


Figure 4.5: LSTM Architecure

$$it = (W_{ih}h_{t-1} + W_{ix}x_t + b_i)(\text{Input Gate}) \tag{4.1}$$

$$ft = (W_{fh}h_{t-1} + W_{fx}x_t + b_f)(\text{Forget Gate}) \tag{4.2}$$

$$ot = (W_{oh}h_{t-1} + W_{ox}x_t + b_o)(\text{Output Gate}) \tag{4.3}$$

LSTM has a few drawbacks. It takes longer to train neural networks because LSTM is not bidirectional; the model learns separately from right to left and from left to right.

4.2.3 K-Nearest Neighbor

The K-Nearest Neighbor (KNN) algorithm is one of the simplest machine learning algorithms to implement based on supervised learning technology. If the training data is extensive, the KNN algorithm would be more effective [31]. This algorithm presupposes correlation among new and available data and assigns new data to the category most closely simulating good categories. All available cases and coordinate new cases based on similarity. With the KNN algorithm, we can easily codify those data as new data emanates into felicitous categories. This algorithm can be used for retrogradation and classification, mainly for disputes. The KNN algorithm is also known as a nonparametric algorithm. Which means it makes no presumption about the prime data. It is also known as a delayed learning algorithm because it saves the dataset instead of learning from the training set instantly and executes a movement on the dataset during classification.

4.2.4 Naive Bayes

Machine learning techniques are used to categorize data using naive Bayes. It assumes that characteristics are highly independent of one another according to Bayes' theorem. As naive Bayes is useful for large datasets, first convert the data into a frequency table, then evaluate the probability and get a solution. In learning tasks, naive Bayes classifiers are scalable and call for linear constraints. Building classifiers using the Naive Bayes technique is easy to create models that classify problem occurrences. Component probabilities are multiplied by conditional probabilities. Here, it indicates that the method expects the presence or absence of a single class characteristic.
Now, P(m/n) = [P(n/m) P(m)] P(n), where p(m) is the predictor's previous probability, p(n) is the existing prospect, and p(m/n) appears for posterior probability
Sentiment analysis aims to identify the emotions and opinions expressed within a text. Naive Bayes is a widely used and effective technique for doing this. Naive Bayes can be used for sentiment analysis to classify text into positive, negative, or neutral sentiment categories. We can use Naive Bayes in sentimental analysis by preparing the data, feature extraction, training, model assessment, and sentiment prediction.

4.2.5 Support Vector Machine

SVMs (Support Vector Machines) are widely used for linear classification problems (Cherkassky, 1997). They aim to identify hyperplanes that can effectively separate the feature space into different classes. These hyperplanes are selected to maximize the distance from the closest data points of each class. The Linear SVC (Support Vector Classifier) is a straightforward and efficient SVM implementation specifically designed for scenarios where a linear separation between classes is assumed. The Linear SVC algorithm offers several benefits for sentiment analysis tasks. It is particularly advantageous when dealing with linearly separable data, often in sentiment analysis, where the goal is to distinguish between positive and negative sentiments [45]. Linear SVC effectively classifies the sentiment of text data by finding the optimal hyperplane that separates the two classes. Another advantage is

its robustness against noise and outliers. It aims to find a decision boundary that generalizes well to new data by maximizing the margin, thereby reducing the impact of noisy or irrelevant features. Furthermore, Linear SVC is computationally efficient and scales well with large datasets, making it ideal for sentiment analysis involving large amounts of text data. The algorithm provides interpretable results by assigning weights to different features, enabling users to understand the importance of specific words or features in the sentiment classification process. Linear SVC has a relatively low memory footprint, making it suitable for deployment in resource-constrained environments. However, it is important to note that sentiment analysis tasks may involve more complex relationships and non-linear patterns, for which advanced techniques like non-linear SVMs or deep learning approaches may be more appropriate.

Given a training dataset with input feature vectors x (representing text data) and corresponding labels y (representing class labels), where I range from 1 to n (number of training samples), the goal is to find a hyperplane that separates the feature space into different classes. The process of using Linear SVC for sentiment analysis can be outlined in several steps, along with the corresponding mathematical explanations and formulas:
The architecture of a linear support vector machine (SVM) consists of several key components. Firstly, the SVM takes as input a labeled dataset, where a feature vector and associated class label represent each sample. These feature vectors capture the relevant information from the input data. The Linear Support Vector Machine (SVM) model does not use a kernel function. Unlike non-linear SVMs, which employ kernel functions to map data into a higher-dimensional space for better separation, the Linear SVM assumes a linear separation between classes in the feature space. This makes the Linear SVM computationally efficient and suitable for large-scale datasets.
Next, feature extraction techniques transform the input data into a suitable numerical representation. These techniques convert the input data, such as text, into feature vectors with numerical values. This step ensures that the SVM can effectively process and analyze the data.
The linear SVM model's core is finding an optimal hyperplane that separates the different classes in the feature space. This hyperplane is a linear decision boundary defined by a weight vector and a bias term. The SVM aims to determine the optimal values for w and b during the training phase.
During the training phase, the SVM learns the optimal values of the weight vector and bias term. This involves solving an optimization problem that maximizes the margin between the support vectors and the decision boundary. The optimization problem aims to minimize the norm of the weight vector (||w||) while satisfying certain constraints.
Once the SVM model is trained, it can classify new, unseen samples. Classification is performed by evaluating the sign of the decision function, which is computed as w • x + b, where x is the input sample. If the decision function is positive (f(x) > 0), the sample is classified as one class. If the decision function is negative (f(x) < 0), the sample is classified as the other class. In some cases, a decision threshold might be applied to handle cases where the decision function is close to zero.
Now, let us discuss the mathematical formulas to implement a linear support vector

machine: Decision Function:
The decision function for a linear SVM can be written as:

$$f(x) = \text{sign}(w \cdot x + b) \tag{4.4}$$

Here:
f(x) represents the predicted class for the input sample x. w is the weight vector, representing the direction of the decision boundary. x is the input sample's feature vector. b is the bias term, which shifts the decision boundary.
Training Objective:
A linear SVM's training objective involves minimizing the weight vector's norm while correctly classifying the training samples. It can be formulated as:

$$\text{minimize} \quad \frac{||w||^2}{2} \tag{4.5}$$

$$\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 \quad \forall \text{ training samples } (x_i, y_i) \tag{4.6}$$

Here:
The objective is to minimize the squared norm of the weight vector ($||w||^2$) divided by 2. The constraints ensure that all training samples are correctly classified with a margin of at least 1. The linear SVM can classify new samples by evaluating the decision function by solving the optimization problem and determining the optimal values of the weight vector (w) and bias term (b).

$$\text{minimize} \quad \frac{\|w\|^2}{2} \tag{4.7}$$

$$\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 \quad \forall \text{ training samples } (x_i, y_i) \tag{4.8}$$

Here:
The objective is to minimize the squared norm of the weight vector ($||w||^2$) divided by 2. The constraints ensure that all training samples are correctly classified with a margin of at least 1.
The linear SVM can classify new samples by evaluating the decision function by solving the optimization problem and determining the optimal values of the weight vector and bias term.

4.2.6 Multinomial Logistic Regression

Logistic regression [17] is a popular algorithm for sentiment analysis, and its architecture involves several components. Firstly, it takes input as a labeled dataset, where a feature vector and associated class label represent each sample. Feature extraction techniques are then applied to transform the input data, such as text, into numerical feature vectors.
The logistic regression model aims to estimate the probability of a binary outcome

based on the input features. It achieves this by modeling the relationship between the features and the probability of belonging to a particular class using a logistic function known as the sigmoid function. The sigmoid function maps the linear combination of the input features and model parameters (weights) to a probability value between 0 and 1.
Softmax regression, or multinomial logistic regression, is a classification model operating on a labeled dataset. The input data consists of samples, each represented by a feature vector and an associated class label. To prepare the data with softmax regression, feature extraction techniques are applied to convert the input data into a suitable numerical representation. These techniques transform the data into feature vectors with numerical values, enabling the model to work effectively.
The architecture of softmax regression revolves around estimating the probability of each class given the input features. Using the softmax function, it models the relationship between the features and the probabilities of belonging to different classes. The softmax function calculates the probabilities by applying the exponential function to the linear combination of the feature vector and the model parameters. It then normalizes these probabilities to obtain a proper probability distribution over all classes.
During the training phase, the model learns the optimal values of its parameters by minimizing a cost function, typically the cross-entropy loss. Optimization algorithms like gradient descent are used to iteratively adjust the model parameters to minimize the difference between the predicted probabilities and the true class labels. The training process ensures that the model becomes more accurate in its predictions.
Once the softmax regression model is trained, it can classify new, unseen samples into multiple classes. Classification is performed by evaluating the probabilities predicted by the model for each class and assigning the class with the highest probability as the predicted class for the given sample. This approach allows the model to provide class predictions based on the input features.
Mathematical Formula for Softmax Regression:
Given an input sample with features represented by a feature vector x, the softmax regression model estimates the probability of each class, denoted as P(y=c|x), where c is the class label. The softmax function is used to compute these probabilities. The softmax function applies the exponential function to the linear combination of the input features and model parameters (weights) w for each class. Then it normalizes the results to obtain a proper probability distribution over all classes.
For a given class c, the probability is calculated as:

$$P(y = c|x) = \frac{e^{z_c}}{\sum_i e^{z_i}} \tag{4.9}$$

Where z_c is the linear combination of the feature vector x and the model parameters associated with class c, the summation is over all classes.
The linear combination z_c is given by:

$$z_c = w_{c_1}x_1 + w_{c_2}x_2 + \ldots + w_{c_n}x_n + b_c \tag{4.10}$$

Here, w_c, w_c, ..., w_{cn} are the weights associated with class c and input features x,

x, ..., x_n, and b_c is the bias term for class c.The linear combination w_{cx} represents the dot product between the weight vector and the feature vector.
During the training phase, the softmax regression model learns the optimal weights and biases by minimizing the cross-entropy loss, which measures the dissimilarity between the predicted probabilities and the true class labels.
Once the softmax regression model is trained, it can be used for classification. Given a new input sample with feature vector x, the model computes the probabilities of all classes using the softmax function. The class with the highest probability is assigned as the predicted class for that sample.
In summary, softmax regression extends logistic regression to handle multi-class classification problems. The model estimates the probabilities of each class using the softmax function. The model parameters are learned by minimizing the cross-entropy loss, and the class determines the predicted class with the highest probability.

4.2.7 Decision Tree Classifier

The decision tree classifier model [23] aims to learn a hierarchical structure of decisions based on the input features to predict the sentiment. It partitions the feature space into regions using decision rules. During training, the model recursively splits the data based on selected features and thresholds. The best feature and threshold selection at each internal node is determined using criteria such as information gain or the Gini index. Classification is performed by traversing the decision tree from the root node to a leaf node based on the feature values of the input sample. Each internal node decides based on the feature and threshold, guiding the traversal toward the leaf node representing the predicted sentiment label.
Mathematical Analysis:
The architecture of a decision tree classifier can be represented mathematically using a set of decision rules. Let X denote the feature vector of an input sample, and let D denote the decision tree model. A decision rule of the form represents each internal node of the tree: If $feature_i <= threshold_i$, go to the left child node. If $feature_i > threshold_i$, go to the right child node.

4.3 Implementation of Architectures

We aim to perform sentiment analysis using BERT on our dataset, as the BERT model is more accurate than the other model, and it is less hassle to do the preprocessing steps. For our research purposes, we have implemented the BERT-based uncased version.

4.3.1 Pre-Training the Model

We used the transformer library for hugging face to construct the BERT-based uncased. Therefore, we begin by loading the text of the customer review input data. The dataset was then cleared of all punctuation. We now input the data as a token using the word piece tokenizer. Using MSM (masked language modeling), which masks 15% of the input's words at random before processing the complete masked

phrase, the model will be used to turn this input into tokens, as we know from the design of BERT (Q). The model then has to predict whether the two sentences follow one another when using NSP (next sentence prediction), which concatenates two masked sentences as input during pre-training.

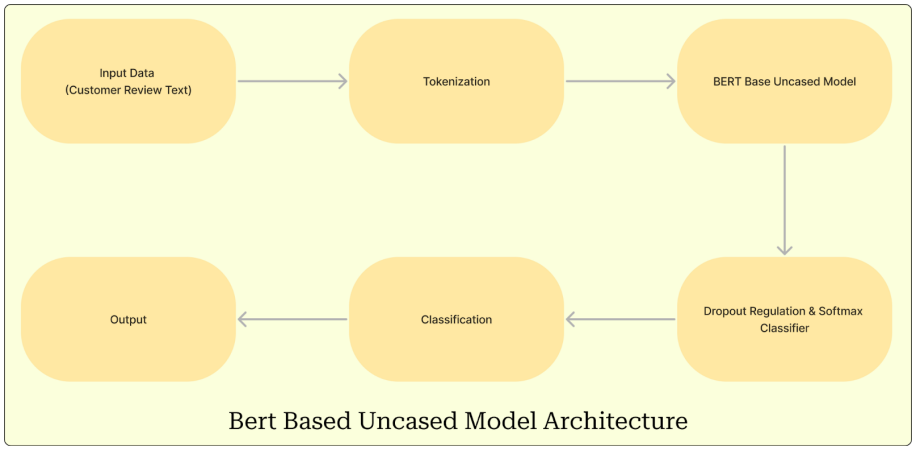


Figure 4.6: Bert Base Uncased

4.3.2 Fine Tuning BERT

Figure 4.6 shows that the dropout regularization as well as a softmax classifier layer comprise the final of the four key phases. Dropout regularization is employed to avoid overfitting, and the softmax function is utilized to translate the input scores into the output probability sum. The softmax function calculation equation is denoted by,

$$s(x_i) = \sum_{j=1}^{n} \frac{e^{x_j}}{e^{x_i}} \tag{4.11}$$

The values of the hyperparameters we utilized for the model are shown in Table 4.1. There are two phases in BERT, and the second phase comprises fine-tuning the BERT, according to Z. Gao et al. [5]. These settings have been utilized to fine-tune the BERT-Based-Uncased for our research.

Although we modified the maximum length, we left the output dimension, the feed-forward layer's dimensions in the encoder, and the number of multiheaded attention and transformer blocks precisely the same. From figure 4.7, we can see that the average max_length for our dataset was 200; however, we increased this to 300 and used 20 epochs every iteration as a precaution.

Name of the Hyperparameter	Hyperparameter Value
Output Dimentions	768
Dimensionally of the feed-forward layer in the encoder	3072
Transformer blocks layer nuumber	12
Multi headed Attention number	12
Total Parameters	110M
Learning rate	0.00001
Max Sequence Length	512
Max_Length	300
Epoch	20

Table 4.1: Hyperparameter Values for BERT-Based Uncased

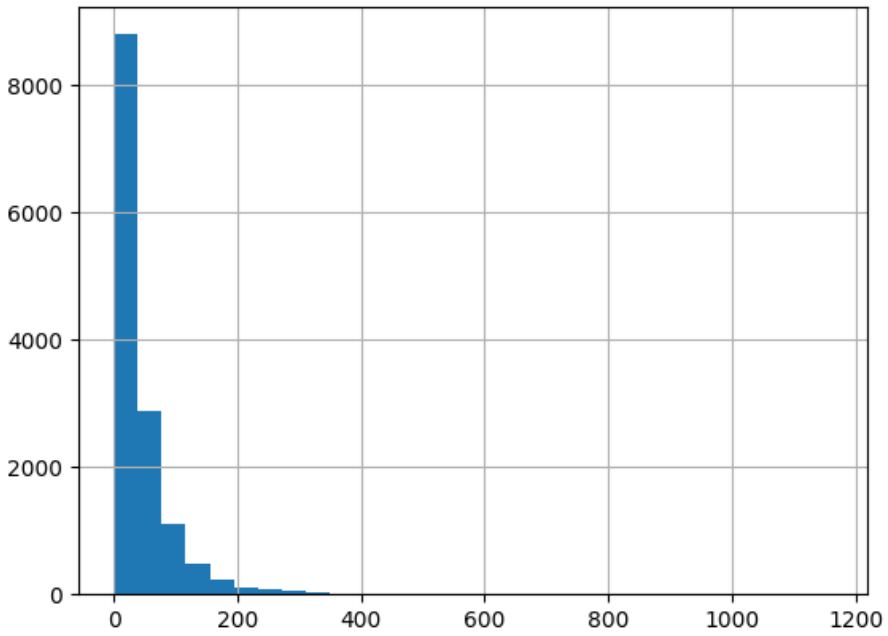


Figure 4.7: Word Max Length

4.4 Evaluation Method

To evaluate our model's working progress, we used the performance metric, for which we used accuracy, recall, precision, and the F1 score for all the models we implemented. Below, we will show the details of this performance metric.

4.4.1 Performance Metrics

The F1 score, accuracy, recall, and precision have all been used to assess the performance metric. We will be able to evaluate performance and better understand how well the models operate on the sentimental analysis tasks by using these performance metrics.
The four quantities of performance metrics are true positive, true negative, false positive, and false negative, or TP, FP, TN, and FN. (p) In our sentimental analysis, TP represents the proportion of data points correctly identified as positive reviews, and

FP represents the proportion of data points falsely classified as negative reviews despite being positive. Similarly, TN represents the number of negative reviews accurately detected, and FN represents the number of negative reviews wrongly identified.

Accuracy

In sentimental analysis, the accuracy metric can be defined as the number of positive sentiments correctly classified by the model divided by the total number of sentiments calculated by the model. More formally, the accuracy can be calculated using the following formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.12}$$

This formula can calculate the accuracy of a model we have used for our sentimental analysis tasks. As we have used the model BERT, Bi-directional LSTM, Logistic Regression, Naive Bayes, Linear SVC, KNN Classifier, and Fitting Decision Tree Classifier for our research, we have evaluated the accuracy for all models, and the data details are given below in 4.2

Name of the model	Accuracy (%)
Bert	84.01
Logistic Regression	83.45
Bi- Directional LSTM	85.08
Naïve Bayes	84.55
Linear SUV	84.45
KNN Classifier	56.45
Decision Tree Classifier	82.62

Table 4.2: Result Accuracy

From Figure a, we can see that BERT is giving the best score compared to other models.

Precision

We used precision to determine if an optimistic prediction was genuinely accurate or not to get beyond the limits of accuracy. We have calculated the model precision based on accurately presenting the outcomes and have established the precision for each model. By counting the data samples anticipated to be positive (TP) and dividing that sum by the total number of correct and incorrect positive predictions (TP, FP), we can see from Equation 1 that we may estimate a model's precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.13}$$

In the table 4.3below, we can see the precision of all the models used for our research.

Architecture	Precision(%)	
	0	1
BERT TUNED	83	85
Logistic Regression	86	85
LSTM	84	84
Naive Bayes	84	85
Linear SVC	84	85
KNN Classifier	87	56
Decision Tree Classifier	77	84

Table 4.3: Result Precision

Recall

The recall represents the percentage of actual positives that were accurately identified, much like accuracy does. Here, we have to calculate the predicted positive sentiment TP divided by the total number of positives, either successfully predicted as positive or mistakenly predicted as negative, which is TP and FN, to determine the recall of the models. The formula for recall calculation is given in Equation 1 below:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.14}$$

Table 4.4, describes the recall of all the models used in our research.

Architecture	Recall(%)	
	0	1
BERT TUNED	81	87
Logistic Regression	80	89
LSTM	78	88
Naive Bayes	80	88
Linear SVC	81	88
KNN Classifier	5	99
Fitting Decision Tree Classifier	81	80

Table 4.4: Result Recall

F1 Score

Another performance statistic that shows the mean of recall and precision is the F1 score. An F1 score can have a maximum value of 1, representing perfect precision and recall, and a minimum score of 0, which happens when either the precision or recall value is zero. The formula to calculate the F1 score is in the following:

$$\text{F1 Score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Precision} + \text{Recall}} \tag{4.15}$$

The F1 score of the models are given in the following table 4.5;

Architecture	F1 Score(%)	
	0	1
BERT TUNED	82	86
Logistic Regression	83	87
LSTM	81	85
Naive Bayes	82	86
Linear SVC	82	86
KNN Classifier	9	72
Decision Tree Classifier	79	82

Table 4.5: Result F1-Score

4.4.2 Tabular Evaluation

As it is simple and straightforward to comprehend, we developed a confusion matrix that describes the efficiency of all the models. It generates a table for the test set of the model that is used to determine how many of the model's predictions were true or erroneous. A test dataset or validation dataset with expected outcome values acts as the starting point for the confusion matrix approach. The confusion matrix separates the expected results and prediction counts from the number of correct assumptions and the number of incorrect assumptions for each class, sorted by the class that was predicted.

Figure 1 describes the fundamental concepts that will enable us to choose the metrics for our models. True positives (TP) occur when both the actual and anticipated values are positive. When both the actual value and the prediction are negative, this is referred to as a true negative (TN). When the fact is negative but the prediction is positive, this is known as a false positive (FP). When the fact is positive but the prediction is negative, this is known as a false negative (FN). The general format of a confusion matrix table for our sentimental analysis models goes as follows:
In figure 4.8, the columns correspond to the predicted class labels, whereas the rows correspond to the actual class labels (foreground and background) for a confusion matrix table.

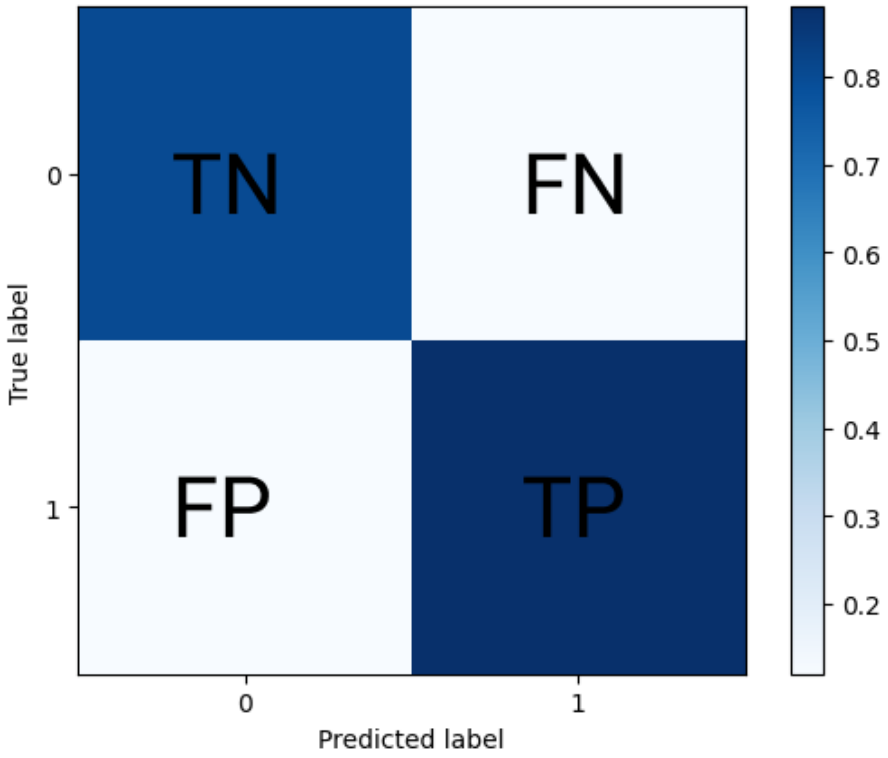


Figure 4.8: Basic 2 × 2 Confusion Matrix

Naive Bayes The confusion matrix table for this model is quite good, as seen in figure 4.9. 88% of the time, it can correctly identify the positive reviews, but 12% of the time, it makes a mistake and counts the good reviews as negative ones. A total of 80% of the negative reviews can be counted appropriately as negative reviews, and 20% of the time, the negative reviews are counted as positive reviews. Overall, it does a decent job at output prediction.

Logistic Regression Figure 4.10 indicates that logistic regression has a high true positive rate of 89, which is a good value for better understanding positive reviews. This model also shows a good percentage of negative reviews. In contrast to the FP rate of 11%, the FN rate is 20%. Overall, this approach can more reliably predict favorable evaluations.

Linear SVC Figure 4.11 illustrates that the linear SVC model has an excellent true positive rate of 88 percent, indicating that the model can successfully predict sentiments. For forecasting negative values, the actual negative rate is also enough. 80 percent of the numbers are TN, and 20 percent are FN. This model is adequate for value prediction.

Fitting Decision Tree Classifier Figure 4.12 shows how the fitting decision tree classifier can more accurately predict the TN, which is 81%. Even though the

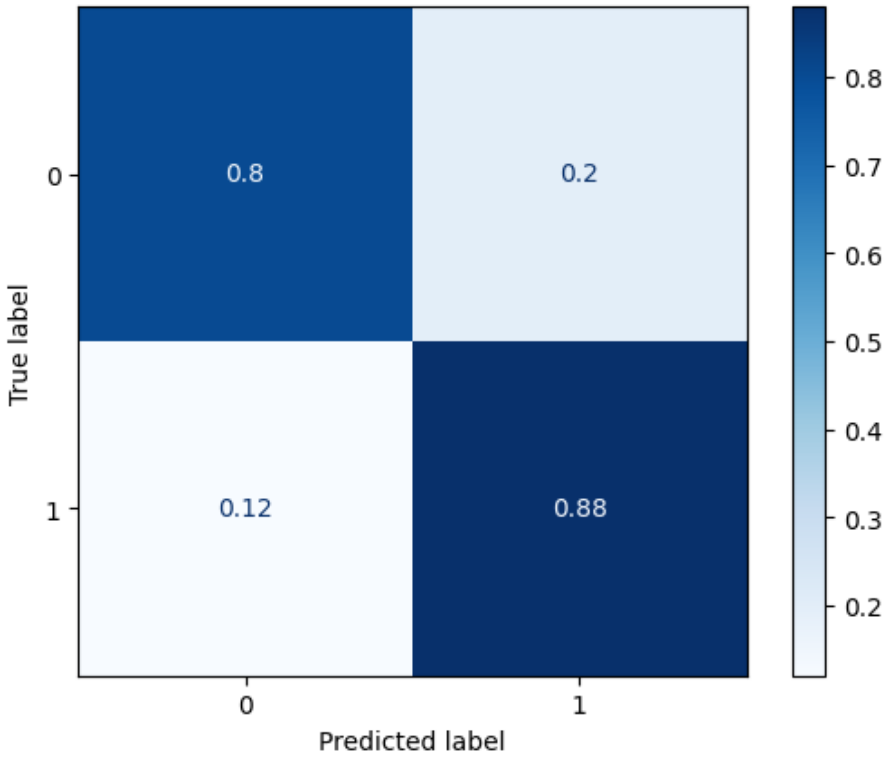


Figure 4.9: Confusion Matrix of Naive Bayes

TP rate is 80%, the algorithm does a decent job at predicting positive reviews. Compared to the other model, the FN rate is lower at 19%. In conclusion, although the TN of the model is similar to other models, the model is incomparable to other models when predicting the TP.

BERT Figure 4.13 shows that the BERT model predicts the TP and TN with an accuracy of 87 percent and 81 percent, respectively, just as the other models do. However, compared to the other variants, this model has a greater FN rate. Therefore, as it has both a greater prediction level of TP and TN, this model may provide a respectable prediction to the reviewers.

LSTM Figure 4.14 shows that the TP and TN can be accurately predicted by the LSTM by 87 and 78 percent, respectively. The TN rate is quite low when compared to the other models, even if the TP prediction rate is quite similar to the other model. It generates a FN of 22%, which is a substantial number in comparison to the other model.

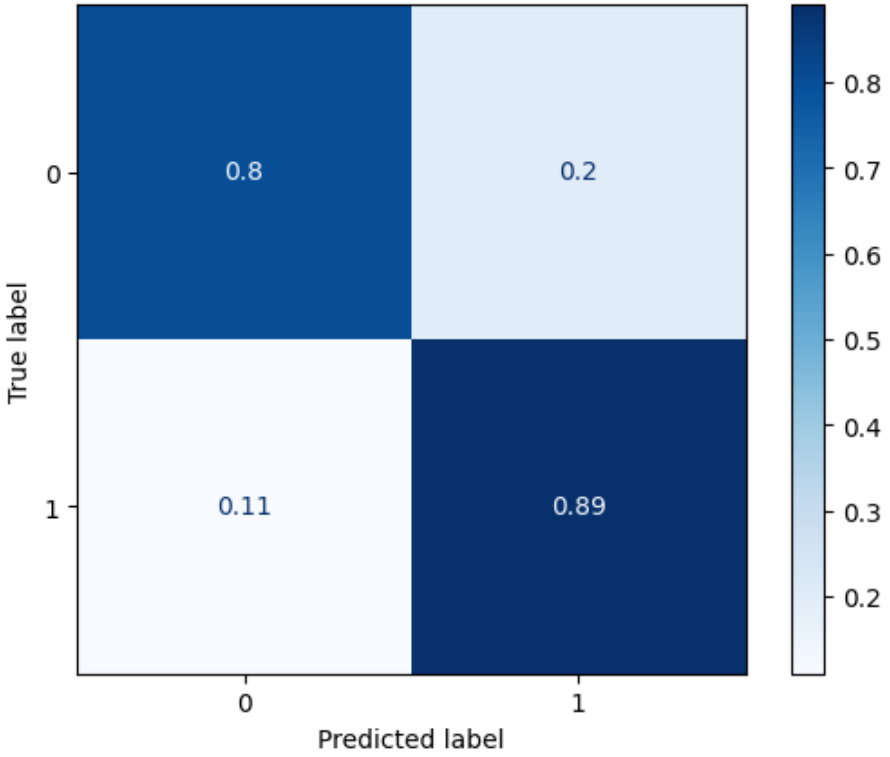


Figure 4.10: Confusion Matrix for Logistic Regression

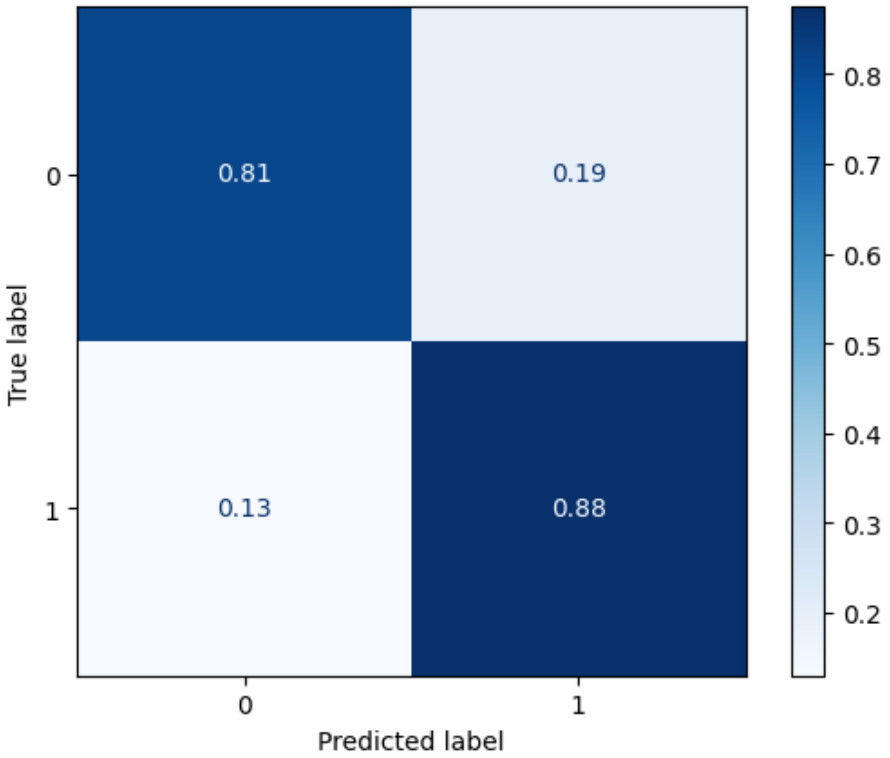


Figure 4.11: Confusion Matrix for Linear SVC

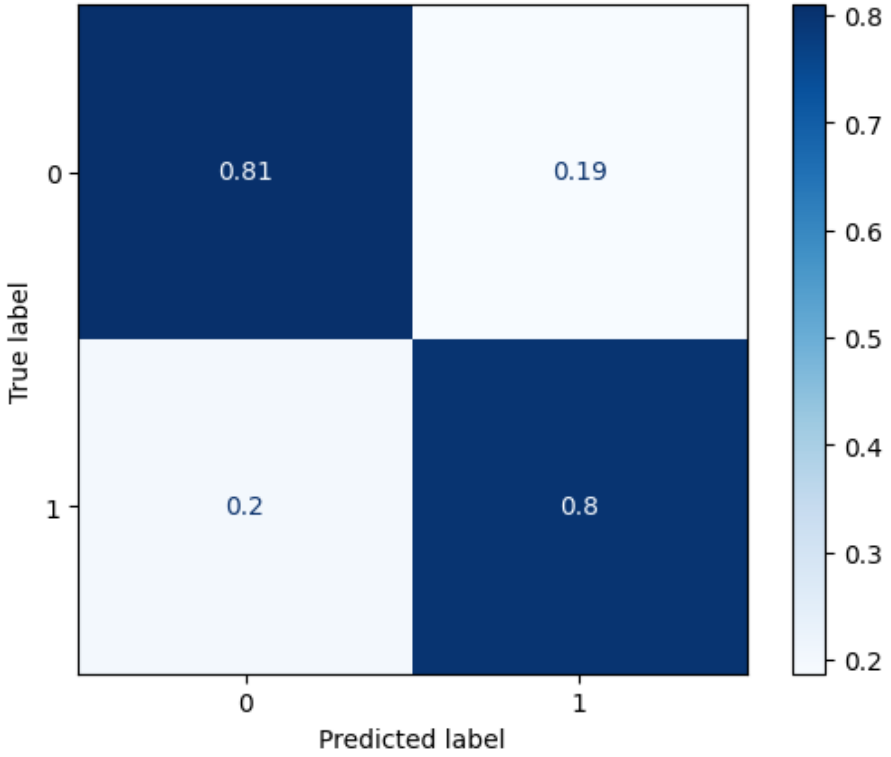


Figure 4.12: Confusion Matrix for Fitting Decision Tree Classifier

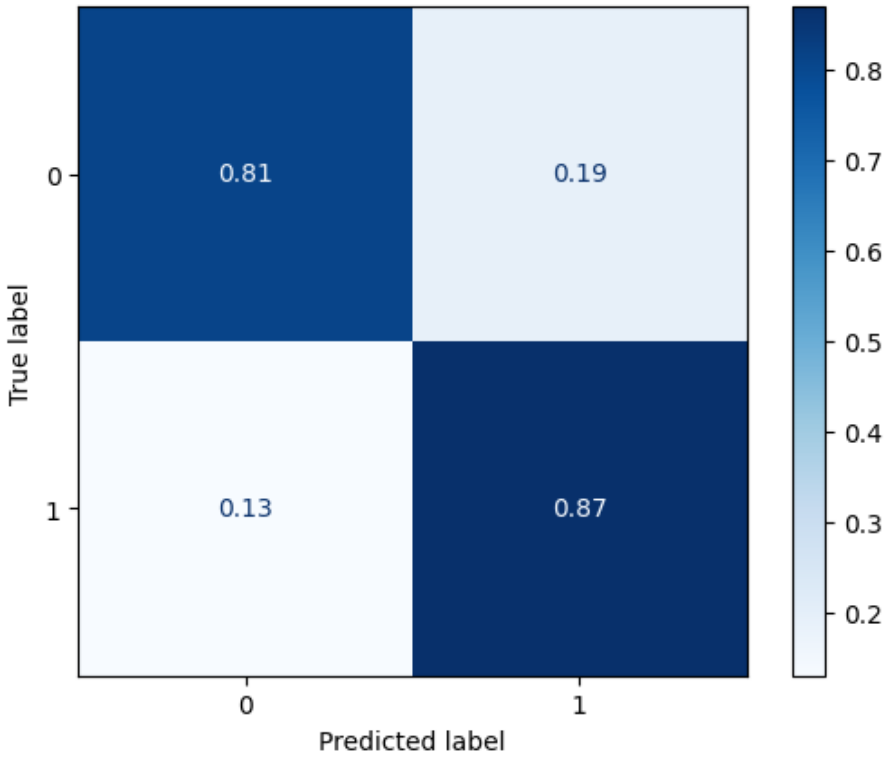


Figure 4.13: Confusion Matrix for BERT

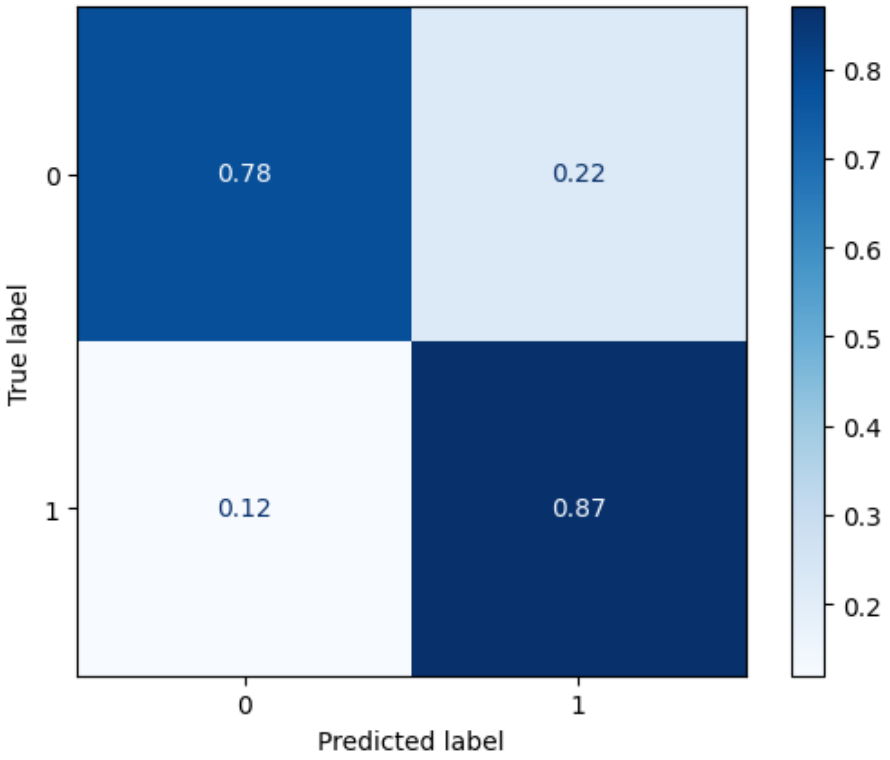


Figure 4.14: Confusion Matrix for LSTM

Chapter 5

Result Analysis

For our result analysis of the architecture model, we have used the explainable AI lime version and will go through the process of how lime works and the effect of the lime version in our research.

5.1 Explainable AI Lime

Machine learning models are frequently called ”black boxes” because we merely provide the model with input and receive the output without fully understanding how this occurred. This is where explainable AI becomes relevant since it clarifies how these models operate and how they will forecast the data. The reason why we need to comprehend what is occurring within the machine learning models is now up for debate. Please assume that the accuracy matrix, recall, and precision in our BERT model produce a high outcome, such as 99 percent or something like that. How can we be sure that machine learning models like BERT and others will behave the same way in the real world? That is the question at hand. We cannot trust the model's prediction based on the other performance criteria. We must, therefore, better comprehend the model in light of what it predicts and based on which it provides outcomes. Explainable AI was therefore put to use.
A description of an arbitrary instance from the test set that was produced using the lime package is provided below in Figure 5.1

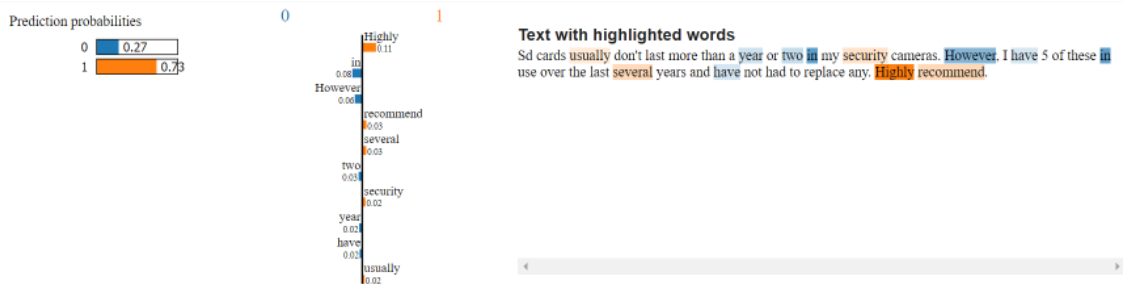


Figure 5.1: Positive Sentiment Lime Visualization

Figure 5.1 shows the words colored blue and yellow as indicators of positive and negative reviews, respectively. The only way to comprehend the models much better is to have an explanation of this, as this is not achievable in the real world by simply

observing the correctness of the raw data. We can see that the model predicts that the sentence will receive 73 percent of negative reviews. However, it is not clear from the accuracy matrix why the model has assigned the sentence a positive review score of 27 percent; however, with the help of the lime, it becomes clearer why the sentence was classified as negative data. In the same way, in Figure 5.2, we can see that the sentence was predicted to be negative for 92 percent.

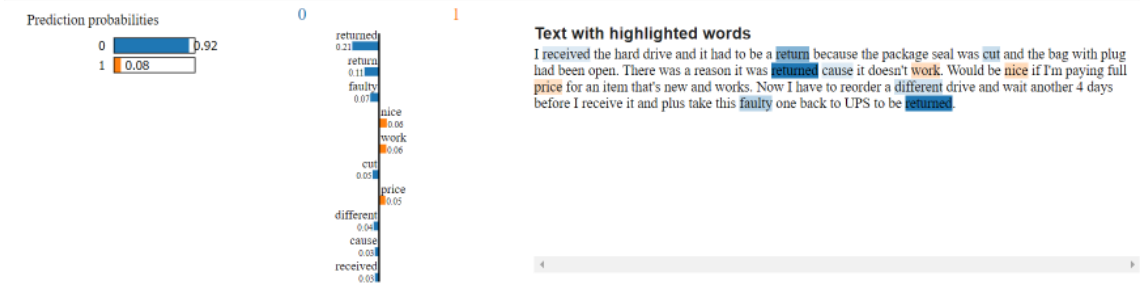


Figure 5.2: Negative Sentiment Lime Visualization

There are ten feature maps in Lime. They are in charge of selecting random words from a sentiment review, and color will subsequently show these attributes. The more positive or negative a word is the darker the hue. Lime will then display the accuracy of each word in the middle line after selecting the words and coloring them. After determining whether a word's accuracy is positive or negative, the overall accuracy is displayed in the left-top corner, as seen in the figure 5.2.

Architecture	Accuracy(%)	Recall(%)		Precision(%)		F1 Score(%)	
		0	1	0	1	0	1
BERT TUNED	84.01	81	87	83	85	82	86
Logistic Regression	83.45	80	89	86	85	83	87
LSTM	85.08	78	88	84	84	81	85
Naive Bayes	84.55	80	88	84	85	82	86
Linear SVC	84.45	81	88	84	85	82	86
KNN Classifier	56.45	5	99	87	56	9	72
Fitting Decision Tree Classifier	82.62	81	80	77	84	79	82

Table 5.1: Accuracy of all architectures

So, as a conclusion to our research, we can observe from the table **??** that the models LSTM, Naive Bayes, Linear SVC, and modified BERT provide the best values among the other models based on the accuracy matrix. However, it is clear from the confusion matrix and all other performance matrices that BERT provides the best outcome. Ince the model providing the best accuracy could more welly create the true positive or true negative better than the BERT, it is clear from other confusion matrices. In addition, BERT does not require the preprocessing steps that other models do, making it more challenging to train the model when using training data already processed. Overall, it is clear that the BERT model provides the most outstanding results compared to the other models and is simple to implement. There are ten feature maps in Lime. They are in charge of selecting random words from a sentiment review, and color will subsequently show these attributes. The more positive or negative a word is the darker the hue. Lime will then display the accuracy of each word in the middle line after selecting the words and coloring them. After

determining whether a word's accuracy is positive or negative, the overall accuracy is displayed in the left-top corner, as seen in the figure 5.2.

Chapter 6

Future Work and Conclusion

6.1 Conclusion

Sentiment Analysis of Customer Product Reviews is a powerful tool for understanding customer needs and feedback. Rapid and precise automatic processing and analysis of massive quantities of customer evaluations are now possible quickly and accurately because of advancements in machine learning. This can provide businesses with valuable insights into customer opinions, preferences, and needs, which can be used to improve products and services and increase customer satisfaction. The results of our study highlight the significance of using modern deep learning algorithms like BERT and show that these models can deliver cutting-edge sentiment classification results on the SemEval-2014 and Twitter datasets. By incorporating target information, the model allows for capturing the nuanced aspects of sentiment, resulting in improved accuracy of 84.01%. The other models, including logistic regression, LSTM, Naive Bayes, linear SVC, and decision tree classifier, KNN, demonstrated competitive performance in sentiment analysis with an accuracy of 85.08%, 84.55%, 84.45%, 82.62%,56.45% respectively. Through an in-depth analysis of their strengths and limitations, it has been unequivocally established that each model possesses unique advantages and disadvantages. Logistic regression, a parsimonious and interpretable algorithm, can handle binary and multi-class classification problems with alacrity. Multinomial naive Bayes, a probabilistic model, is particularly well-suited for text classification tasks; however, it assumes that features are independent. Linear support vector classifiers, a powerful model for text classification, can easily handle large datasets and high-dimensional feature spaces. The results of this study unequivocally demonstrate that the optimal model for a sentiment analysis task will be contingent upon the dataset's specific characteristics and the analysis's objectives. This research has far-reaching implications for natural language processing and machine learning, as it provides a profound understanding of the strengths and weaknesses of these models for sentiment analysis.

6.2 Future Work

In our work, we implemented the Bert Base Uncased model and refined it for research. Additionally, we used precision, recall, precision, and f1 score as accuracy matrices. In the future, once the model is optimized, we can implement a product

recommendation system. Moreover, based on sentiment analysis, if the sentiment of the reviews is positive, we can generate a specific product recommendation system.

Bibliography

[1] J. Schmidhuber, S. Hochreiter, *et al.*, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] S. A. Macskassy, H. Hirsh, A. Banerjee, and A. A. Dayanik, "Converting numerical classification into text classification," *Artificial Intelligence*, vol. 143, no. 1, pp. 51–77, 2003.

[3] F. Fu and L. Wang, "Coevolutionary dynamics of opinions and networks: From diversity to uniformity," *Physical Review E*, vol. 78, no. 1, p. 016 104, 2008.

[4] S. Huang, X. Liu, X. Peng, and Z. Niu, "Fine-grained product features extraction and categorization in reviews opinion mining," in *2012 IEEE 12th International Conference on Data Mining Workshops*, IEEE, 2012, pp. 680–686.

[5] R. Dong, M. Schaal, M. P. O'Mahony, and B. Smyth, "Topic extraction from online reviews for classification and recommendation," in *Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 13), Beijing, China, 3-9 August 2013*, AAAI, 2013, pp. 1310–1316.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[7] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using convolutional neural network," in *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*, IEEE, 2015, pp. 2359–2364.

[8] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[9] D. Li and J. Qian, "Text sentiment analysis based on long short-term memory," in *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, IEEE, 2016, pp. 471–475.

[10] T. A. Rana and Y.-N. Cheah, "Aspect extraction in sentiment analysis: Comparative analysis and survey," *Artificial Intelligence Review*, vol. 46, pp. 459–483, 2016.

[11] A. Calheiros, S. Moro, and P. Rita, "Sentiment classification of consumer generated online reviews using topic modeling," *Journal of Hospitality Marketing Management*, vol. 26, Mar. 2017. DOI: 10.1080/19368623.2017.1310075.

[12] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using bilstm-crf and cnn," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.

[13] M. Cliche, "Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms," *arXiv preprint arXiv:1704.06125*, 2017.

[14] N. Gaur and N. Sharma, "Sentiment analysis in natural language processing," *Int. J. Eng. Technol*, vol. 3, pp. 144–148, 2017.

[15] A. S. Manek, P. D. Shenoy, and M. C. Mohan, "Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and svm classifier," *World wide web*, vol. 20, pp. 135–154, 2017.

[16] A. G. Prasad, S. Sanjana, S. M. Bhat, and B. S. Harish, "Sentiment analysis for sarcasm detection on streaming short text data," in *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, 2017, pp. 1–5. DOI: 10.1109/ICKEA.2017.8169892.

[17] W. Ramadhan, S. A. Novianty, and S. C. Setianingsih, "Sentiment analysis using multinomial logistic regression," in *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, IEEE, 2017, pp. 46–49.

[18] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Information fusion*, vol. 36, pp. 10–25, 2017.

[19] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *IEEE access*, vol. 6, pp. 23 253–23 260, 2018.

[20] B. Wang and W. Lu, "Learning latent opinions for aspect-level sentiment classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[21] F. Abid, M. Alam, M. Yasir, and C. Li, "Sentiment analysis through recurrent variants latterly on convolutional neural network of twitter," *Future Generation Computer Systems*, vol. 95, pp. 292–308, 2019.

[22] M. Anandarajan, C. Hill, T. Nolan, M. Anandarajan, C. Hill, and T. Nolan, "Text preprocessing," *Practical text analytics: Maximizing the value of text data*, pp. 45–59, 2019.

[23] V. A. Fitri, R. Andreswari, and M. A. Hasibuan, "Sentiment analysis of social media twitter with case of anti-lgbt campaign in indonesia using naïve bayes, decision tree, and random forest algorithm," *Procedia Computer Science*, vol. 161, pp. 765–772, 2019.

[24] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with bert," *IEEE Access*, vol. 7, pp. 154 290–154 299, 2019. DOI: 10.1109/ACCESS.2019.2946594.

[25] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with bert," *Ieee Access*, vol. 7, pp. 154 290–154 299, 2019.

[26] D. Hyun, C. Park, M.-C. Yang, I. Song, J.-T. Lee, and H. Yu, "Target-aware convolutional neural network for target-level sentiment analysis," *Information Sciences*, vol. 491, pp. 166–178, 2019.

[27] G. Liu and J. Guo, "Bidirectional lstm with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[28] B. G. Priya, "Emoji based sentiment analysis using knn," *International Journal of Scientific Research and Review*, vol. 7, no. 4, pp. 859–865, 2019.

[29] J. F. Sánchez-Rada and C. A. Iglesias, "Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison," *Information Fusion*, vol. 52, pp. 344–356, 2019, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.05.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253518308704.

[30] K. Shu, D. Mahudeswaran, and H. Liu, "Fakenewstracker: A tool for fake news collection, detection, and visualization," *Computational and Mathematical Organization Theory*, vol. 25, pp. 60–71, 2019.

[31] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 1255–1260.

[32] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional cnn-lstm model for dimensional sentiment analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 581–591, 2019.

[33] T. Wolf, L. Debut, V. Sanh, *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[34] M. Wongkar and A. Angdresey, "Sentiment analysis using naive bayes algorithm of the data crawler: Twitter," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, IEEE, 2019, pp. 1–5.

[35] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," *arXiv preprint arXiv:1904.02232*, 2019.

[36] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[37] A. Borg and M. Boldt, "Using vader sentiment and svm for predicting customer response sentiment," *Expert Systems with Applications*, vol. 162, p. 113 746, 2020.

[38] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[39] K.-X. Han, W. Chien, C.-C. Chiu, and Y.-T. Cheng, "Application of support vector machine (svm) in the sentiment analysis of twitter dataset," *Applied Sciences*, vol. 10, no. 3, p. 1125, 2020.

[40] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, "User reviews: Sentiment analysis using lexicon integrated two-channel cnn–lstm family models," *Applied Soft Computing*, vol. 94, p. 106 435, 2020.

[41] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, "Recent trends in deep learning based personality detection," *Artificial Intelligence Review*, vol. 53, pp. 2313–2339, 2020.

[42] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in chinese based on sentiment lexicon and deep learning," *IEEE access*, vol. 8, pp. 23 522–23 530, 2020.

[43] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[44] H.-T. Duong and T.-A. Nguyen-Thi, "A review: Preprocessing techniques and data augmentation for sentiment analysis," *Computational Social Networks*, vol. 8, no. 1, pp. 1–16, 2021.

[45] N. Yadav, O. Kudale, A. Rao, S. Gupta, and A. Shitole, "Twitter sentiment analysis using supervised machine learning," in *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020*, Springer, 2021, pp. 631–642.