UNIVERSITI TEKNOLOGI MARA

ASPECT-BASED SENTIMENT ANALYSIS OF CLOTHING REVIEWS

MUHAMMAD EHSAN BIN MAT SUKURY

BACHELOR OF INFORMATION SYSTEMS (HONS.) INTELLIGENT SYSTEMS ENGINEERING

FEBRUARY 2024

UNIVERSITI TEKNOLOGI MARA

ASPECT-BASED SENTIMENT ANALYSIS OF CLOTHING REVIEWS

MUHAMMAD EHSAN BIN MAT SUKURY

Report submitted in fulfilment of the requirements for Bachelor of Information Systems (Hons.) Intelligent Systems Engineering

College of Computing, Informatics and Mathematics

February 2024

SUPERVISOR APPROVAL

ASPECT-BASED SENTIMENT ANALYISI OF CLOTHING REVIEWS

By

MUHAMMAD EHSAN BIN MAT SUKURY

2022930715

This report was prepared under the supervision of the project supervisor, Madam Nurzeatul Hamimah Abdul Hamid. It was submitted to the College of Computing, Informatics and Mathematics and was accepted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Hons.) Intelligent Systems Engineering.

Approved by

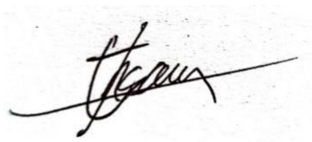
Madam Nurzeatul Hamimah Abdul Hamid
Project Supervisor

January 23rd, 2024

STUDENT DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Teknologi MARA. It is original and is the results of my own work, unless otherwise indicated or acknowledged as referenced work. This thesis has not been submitted to any other academic institution or non-academic institution for any degree or qualification.

I, hereby, acknowledge that I have been supplied with the Academic Rules and Regulations for Postgraduate, Universiti Teknologi MARA, regulating the conduct of my study and research.

Name of Student	:	Muhammad Ehsan bin Mat Sukury
Student I.D. No.	:	2022930715
Programme	:	Bachelor of Information Systems (Hons.) Intelligent Systems Engineering – CS259
Faculty	:	College of Computing, Informatics and Mathematics
Thesis Title	:	Aspect-based Sentiment Analysis of Clothing Reviews
Signature of Student	:	 ………………………………………………..
Date	:	January 2024

ABSTRACT

Understanding customer sentiments towards clothing products was crucial for businesses to enhance customer satisfaction and product development. This study presented an aspects-based sentiment analysis approach applied to a dataset of clothing reviews. The primary objectives of the study were to identify the key aspects mentioned in clothing reviews, including fabric, design, size, color, pattern, price, comfort, quality, durability, and service. The study utilized natural language processing (NLP) techniques to extract these aspects from the reviews and analyze the sentiments associated with them. Specifically, the study employed spaCy for aspect extraction using predefined patterns and sentiment analysis based on the extracted aspects. The sentiment analysis focused on identifying positive, neutral, and negative sentiments expressed towards each aspect, providing valuable insights into customer preferences and areas for improvement in clothing products. Finally, the study designed and developed an interactive dashboard that visualized the results of the ABSA, providing stakeholders with an intuitive tool to explore and understand customer sentiments towards clothing products. The dashboard utilized data visualization techniques such as sentiment distribution charts and aspect-specific sentiment trends, enabling users to gain actionable insights for product improvement and customer satisfaction enhancement. The study's findings contributed to the understanding of customer perceptions in the clothing industry and provided a valuable tool for businesses to make data-driven decisions based on customer feedback.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude and appreciation to my supervisor, Madam Nurzeatul Hamimah Abdul Hamid, for her guidance, support, and valuable insights throughout this project. Her expertise and encouragement had helped me in the quality of my work.

I would also like to extend my heartfelt thanks to my lecturer, Dr. Azliza Mohd Ali, for her continuous support and guidance in the subject CSP600 and CSP650. Her knowledge, and dedication to teaching have greatly contributed to my learning experience.

Lastly, I would like to acknowledge the support of my family and friends who have been a motivation and encouragement throughout this journey. Thank you all for your invaluable support and contributions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations

ABSA	Aspect-Based Sentiment Analysis
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
NaN	Not A Number
WOM	Word-Of-Mouth
E-WOM	Electronic Word-Of-Mouth
AI	Artificial Intelligence
SVM	Support Vector Machine
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers

CHAPTER ONE
INTRODUCTION

The research will explore a related topic with a major focus on aspect-based sentiment analysis of clothing reviews, which will be further analysed in detail to gain insights into customers' opinions and sentiments towards different aspects of clothing products. The background study, problem statement, research questions, research objectives, project scope, and significance related to aspect-based sentiment analysis of clothing reviews will be covered in this chapter.

1.1 Research Background

Understanding customer sentiment and preferences is becoming more important than ever for businesses in the rapidly changing world of the fashion industry. Clothing reviews are important in forming opinions among customers and influencing their buying decisions. Sentiment analysis has become a useful method for extracting useful information from this huge volume of client feedback. Businesses can extract and analyse the emotions mentioned in clothing reviews by using sentiment analysis techniques, which helps them better understand consumer satisfaction levels, pinpoint problem areas, and make data-driven decisions (Yu & Bai, 2021).

Electronic word of mouth, or E-WOM, refers to "any positive or negative verbal statement by potential, actual, and former customers of a product or company". Online customer reviews of products and services are a significant source of E-WOM which is an unstructured data that provides a deep understanding of customer sentiment. Social media networks generate enormous amounts of data every day, and human data analysis is time-consuming and expensive for businesses (Nawaz et al., 2021).

According to an article (Yu & Bai, 2021), this paper categorizes 1150 clothes reviews into 6 aspects which are fabric, style, size, colour and pattern, pricing, and others. The huge number of product reviews can often act as a useful source for knowledge extraction. Online reviews are believed to assist businesses in better understand customer wants, identifying product flaws, enhancing product quality, and generating more revenue.

In recent years, there has been a lot of interest in using sentiment analysis in the context of clothing evaluations. Sentiment analysis has the potential to reveal customer sentiments, identify new trends, and offer insightful feedback for product development and marketing plans, according to researchers and practitioners. Customers' satisfaction and loyalty are crucial success factors in the fiercely competitive clothing market, according to (Kumar et al., 2019). Finding out what customers think about clothing products has thus become a top focus for companies in this industry.

However, there are still a number of issues that need to be resolved despite the increased interest in sentiment analysis of apparel reviews. Sentiment analysis models face difficulties because of the complexity of language used in contexts relating to fashion, such as slang and fashion-specific terms. Furthermore, extensive analysis techniques are needed due to the subjective nature of feelings and the requirement to take into account a number of clothing-related factors, including fabric, fit, style, and cost.

1.2 Problem Statement

Nowadays, a lot of businesses are moving their operations online due to the rising popularity of online shopping among consumers, who like doing their shopping online (Marwat et al., 2022). Clothing makes up a significant proportion of e-commerce sales as a typical type of experience product. Leading e-commerce companies like Amazon, Taobao, and JD have created a separate category section on their websites just for apparel. Customers can use online reviews as a valuable reference when assessing products. On e-commerce platforms, however, reviews for moderately popular apparel items can reach tens of thousands, which puts the company at risk of information overload (Lu et al., 2023).

Additionally, ratings may not accurately reflect the sentiment expressed in the associated review texts (Almansour et al., 2022). In other words, while numerical ratings are often used as a proxy for sentiment in sentiment analysis models, there are instances where the ratings may not capture the full sentiment expressed in the textual content of the reviews. Companies and designers are unable to effectively use feedback from customers to develop their products and make informed choices due to an incomplete understanding of client opinions. Additionally, businesses find manual data analysis to be expensive and time-consuming due to the overwhelming volume of Internet reviews (Nawaz et al., 2021).

1.3 Research Questions

There are a few uncertainties that must be addressed to define the study's goals further. Questions such as these must be addressed:

I. What aspects are crucial for companies/designers based on customer reviews?
II. How can online reviews be analysed effectively?
III. What are the visualization techniques for analysing online reviews?

1.4 Research Objectives

Numerous objectives must be met to accomplish the study's objective, including the following:

I. To identify the key aspects of clothing reviews.
II. To apply Aspect-Based Sentiment Analysis for clothing reviews.
III. To develop a dashboard for Aspect-based Sentiment Analysis of Clothing Reviews.

1.5 Research Scope

The scope for this research includes:

I. The sentiment analysis will analyse a vast amount of customer feedback on clothing products.
II. The analysis will focus on reviews of top and bottom outfits.
III. The sentiment analysis will cover nine key aspects of clothing which is fabric, design, size, color, price, comfort, quality, durability, and service

1.6 Significance of Study

Sentiment analysis is important for helping companies understand completely how customers feel about their clothing products. Businesses may identify areas for improvement and change their product development plans by examining the opinions shared in consumer reviews. By better understanding their target market's preferences, companies can adapt their marketing strategies. Additionally, sentiment research enables companies to keep up with changing fashion trends and customer preferences. Businesses can successfully satisfy their customers by adjusting their products and marketing initiatives accordingly. By using the insights from sentiment analysis, companies may prioritize improvements based on feedback from customers and raise overall satisfaction with customers.

CHAPTER TWO
LITERATURE REVIEW

This chapter will discuss related studies conducted by other researchers as well as the subjects and approaches mentioned later in this chapter. To better grasp the knowledge domain, the researcher will also extract and explain the domain from their study. The researcher will also draw conclusions about the area and explain them. The facts and expertise on the study topic that will be useful for further research and project development are outlined in this chapter.

2.1 Clothing

One of an individual's basic needs is clothing. People dress differently for various occasions, from casual everyday wear to formal attire, depending on their needs and preferences. Clothes serve a variety of purposes, including danger or environmental protection, providing protection against cold or heat as well as potential harmful elements, and facilitating the wearer's daily activities. In addition to serving a practical purpose, clothing can convey a person's preferences, personality, and sense of style to the outside world (Purwaningtyas & Rahadi, 2021).

2.2 Online Business of Clothing

The growing popularity of e-commerce has completely changed the way businesses operate in the clothing sector by enabling them to connect with a larger client base and offer easier shopping experiences. Numerous studies have looked at different facets of e-commerce apparel businesses, revealing their strategies, challenges, and potential. The authors of a study by (Purwaningtyas & Rahadi, 2021) looked at the variables affecting customers' purchase intentions in online clothes businesses. They discovered online clothing purchase decision can be affected by ten factors that customers consider before making the purchase. The ten factors are price, promotion, product design/style, product quality, brand image, information availability, seller, trustworthiness, product variety, ease of use and service quality which directly or indirectly affects the purchase decision making process. The figure 2.1 shows the illustration of factors that customers consider before making the purchase.

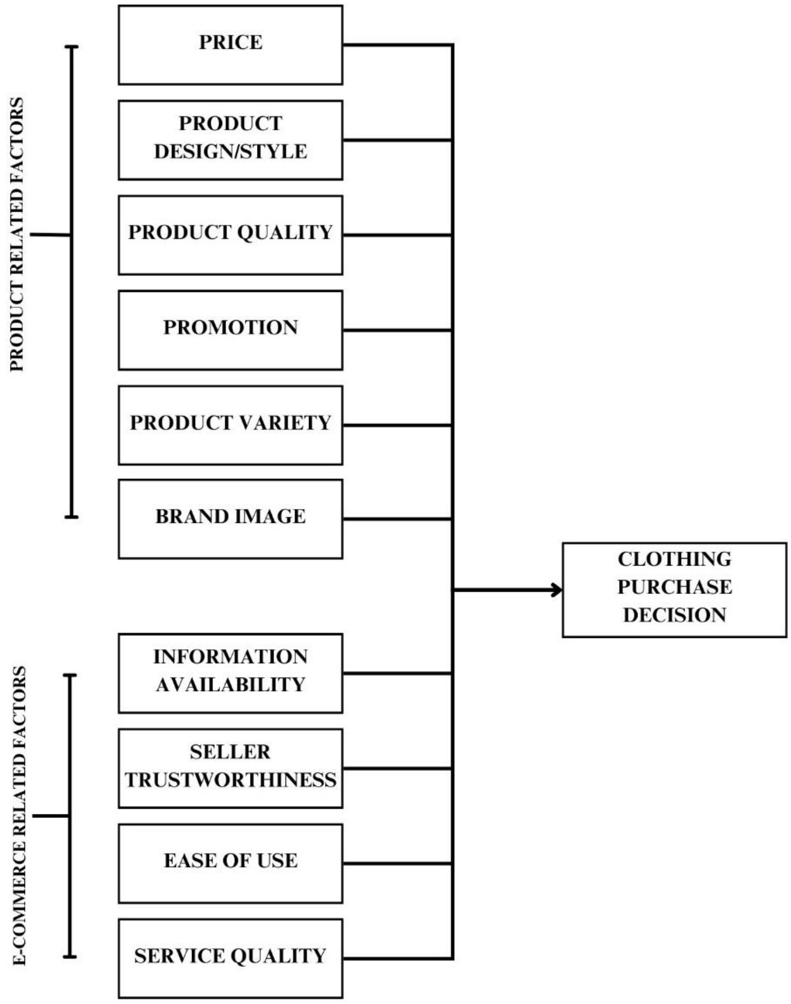


Figure 2.1 Factors of Customer Purchase Decision

(Source: Purwaningtyas & Rahadi, 2021*)*

This article also explains that product quality, which is better referred to as perceived product quality in the online environment. Customers can acquire accessible descriptions or information about the products and information from their prior knowledge as they cannot immediately touch or feel the products to determine the quality. Word-of-mouth and customer reviews are additional sources of product information that can aid consumers in determining the quality of a product.

2.3 Online Review

Today online communication tools and easy access to the internet have changed face-to-face word-of-mouth (WOM) communication to electronic word-of-mouth (E-WOM) communication. E-WOM is the term used to describe any statement, whether positive or negative, made by potential, actual, or past customers regarding a good or service and made available to a large number of individuals and organizations online. Customers can use a variety of internet-based platforms, including retailer websites, brand communities, independent web pages, consumer blogs, etc., to share their thoughts and experiences on a good or service(Tanvir Abir et al., 2020).

Online reviews are a type of electronic word-of-mouth (E-WOM) that offer information based on the publicly available personal experiences of past customers of a particular product, acting as free "sales assistants" to assist other consumers in choosing the product that best matches their requirements and tastes (Fresneda & Gefen, 2019). Online reviews are seen to be beneficial for both product manufacturers and potential purchasers. To give an example, customers may use online reviews to help them decide whether or not to buy a particular product. On the other hand, manufacturers of goods use online customer reviews to identify current consumer tastes and then apply this information to product development, marketing, and customer relationship management.

In fact, a Nielsen study of 28,000 Internet users from 56 countries found that 70% of them recognised online reviews as a reliable source of information, placing them as the second most dependable form of advertising out of 19 alternatives. As a result, experts in marketing want to understand how to effectively encourage and manage online reviews (Yousefi Dahka et al., 2019). Numerous studies looked at online reviews from a variety of angles, including how they affected consumer expectations, customer satisfaction factors, psychological factors, and firm responses.

2.4 Aspects of Clothing

One study explores the terms frequently used in web searches for inherent features in clothing fabrication. 31 experienced designers and 686 other participants completed a survey as part of the study, in which they were asked to list the terms they would use to look up specific aspects of clothing on Google. Themes found in the comments were examined, including aspects such as design, manufacturing, color, material, and type of clothing. The results imply that comprehending internet search trends can offer perceptions into the behavior of customers and facilitate efficient communication regarding clothing(Eckman et al., 2020).

In addition, a study that focuses on female Tuban consumers between the ages of 18 and 30 investigates consumer preferences for various clothing product aspects. Using a conjoint analysis and descriptive quantitative methodology, the study surveys 97 participants who were chosen by purposive sampling. According to the results, respondents ranked comfort and color as the most significant aspects of clothing products. They also ranked price ranges, loose comfort, trendy styles, soft colors, and cotton fabric types as their top combinations of features. The purpose of this study is to shed light on consumer preferences so that businesses such as Wear Saoirse can better comprehend and cater to the needs of their target market (Permana & Dewi, 2020).

Next study focuses on how small and medium-sized clothing e-commerce businesses can benefit from big data analysis. It offers practical ways for these businesses to use big data technology effectively. The study collected data through online questionnaires and web searches, obtaining 252 valid responses and 11,864 relevant comments. Some key findings include the fact that 68% of men buy clothes due to seasonal changes, compared to only 31% of women. Additionally, women with higher incomes pay more attention to clothing style and e-commerce service. The analysis also reveals that clothing style is the most important factor for customers when choosing clothes. Customer reviews mainly mention clothing style, fashion, service attitude, and price, with clothing style and fashion reviews ranking the highest. Finally, the study suggests marketing strategies based on these findings, such as gender-specific recommendations and personalized marketing. (Chang, 2020).

Lastly, the study shows that the concept of perceived quality in consumer behavior refers to how consumers assess a product's overall performance or superiority based on various cues, rather than solely on its inherent quality. These cues can be extrinsic, such as the company's reputation, brand image, and pricing, or intrinsic, including the physical attributes of the product like its features, design, size, materials, and perceived durability. In the context of clothing, consumers often develop an interest in a particular garment based on sensory observations like its appearance, texture, and overall feel. These sensory perceptions influence their judgments about the material, design, and craftsmanship of the clothing. Studies in the clothing industry have specifically noted a significant relationship between customers perceived quality and their interest in clothing (Rahman, 2017).

2.5 Sentiment Analysis

As opinions on social media and other websites increase more and more quickly, sentiment analysis is becoming increasingly important. Because of the recent significant explosion in communication, air traffic, and alternative markets, this enormous amount of data cannot be controlled or analysed using the conventional methods, so scientists and researchers have developed high-efficiency techniques to deal with this data. To make the best judgement in this situation, the sentiment analysis must analyse the input and understand its polarity. The five steps of sentiment analysis data processing method include data collection, text preparation, sentiment detection, sentiment classification, and output display, as shown in Figure 2.2 below (Aqlan et al., 2019).

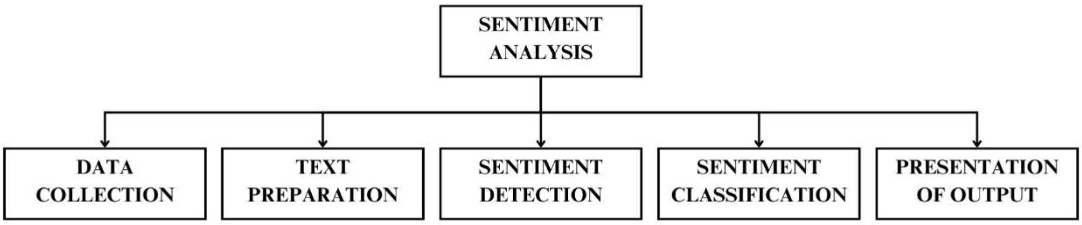


Figure 2.2 Steps of Sentiment Analysis

(Source: Aqlan et al., 2019*)*

2.5.1 Level of Sentiment Analysis

Sentiment analysis is a very active area of research in natural language processing that enables the extraction of opinions from a collection of documents. Sentiment analysis can be looked into on various levels which is document-level, sentence-level, phase level or aspect level (Wankhade et al., 2022).

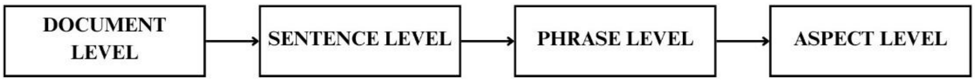


Figure 2.3 Sentiment Analysis Levels

(Source: Wankhade et al., 2022*)*

I. Document Level Analysis

A single polarity is assigned to the entire document after performing document-level sentiment analysis on it. Not many companies apply sentiment analysis of this kind. It can be used to categorize a book's chapters or pages as good, negative, or neutral (Wankhade et al., 2022). At this level, the document can be classified using both supervised and unsupervised learning approaches. The two most important issues related to document-level sentiment analysis are cross-domain and cross-language sentiment analysis (Saunders & Byrne, 2020). It has been demonstrated that domain-specific sentiment analysis can maintain a high level of domain sensitivity while achieving remarkable accuracy. The feature vector in these tasks is a set of words that must be constrained and domain specific.

II. Sentence Level Analysis

Each sentence is examined at this level of analysis in order to determine the polarity that corresponds with it. This is very helpful when a document is associated with a diverse range and mix of sentiments (Wankhade et al., 2022). According to (Rao et al., 2018), this classification level is connected to subjective classification. With more training data and processing resources, each sentence's polarity will be decided independently using the same approaches as the document level. Each sentence's polarity can be combined to determine the overall sentiment of the document or utilized individually. For certain uses, document-level sentiment analysis may not be sufficient (Behdenna et al., 2018). Finding subjective sentences has been a focus of earlier work on sentence-level analysis. However, more challenging tasks, include dealing with ambiguous or conditional sentences (Ferrari & Esuli, 2019). Sentiment analysis at the sentence level is essential in these situations.

III. Phrase Level Analysis

Opinion words at the phrase level will be mined as part of sentiment analysis, and the words will then be categorised. The number of elements in each phrase might range from one to many. Here, it is noted that a single aspect is stated in a phrase, which may be useful for product reviews with many lines (Wankhade et al., 2022). Recently, it has been a popular subject for research. While document-level analysis focused on classifying the entire document as subjective, either positively or negatively, sentence-level analysis is more useful because a document contains both positive and negative comments. The word is the most fundamental unit of language, and the subjectivity of the sentence or document in which it appears has a direct bearing on the word's polarity. A sentence is more likely to be an adjective sentence if it contains an adjective(Wankhade et al., 2022). Furthermore, the phrase selected for expression reflects the demographic characteristics of individuals, such as their age and gender, as well as their desires, social status, and personalities, as well as other psychological and social traits (Flek, 2020). Consequently, the term provides the structure for text sentiment analysis.

IV. Aspect Level Analysis

At the aspect level, sentiment analysis is conducted. There may be several aspects in each statement, hence aspect level sentiment analysis is necessary. Priority is given to each aspect that is employed in the sentence, and each aspect is given a polarity before an aggregate sentiment is derived for the entire sentence (Wankhade et al., 2022).

2.5.2 Sentiment Analysis Approach

Lexicon-Based Approach, Machine Learning Approach, and Hybrid Approach are the three most often used approaches for sentiment analysis. Additionally, scientists are constantly looking for more efficient methods to complete the task with higher precision and less computing expense. Fig. 4 depicts an overview of the many techniques utilised in sentiment analysis.

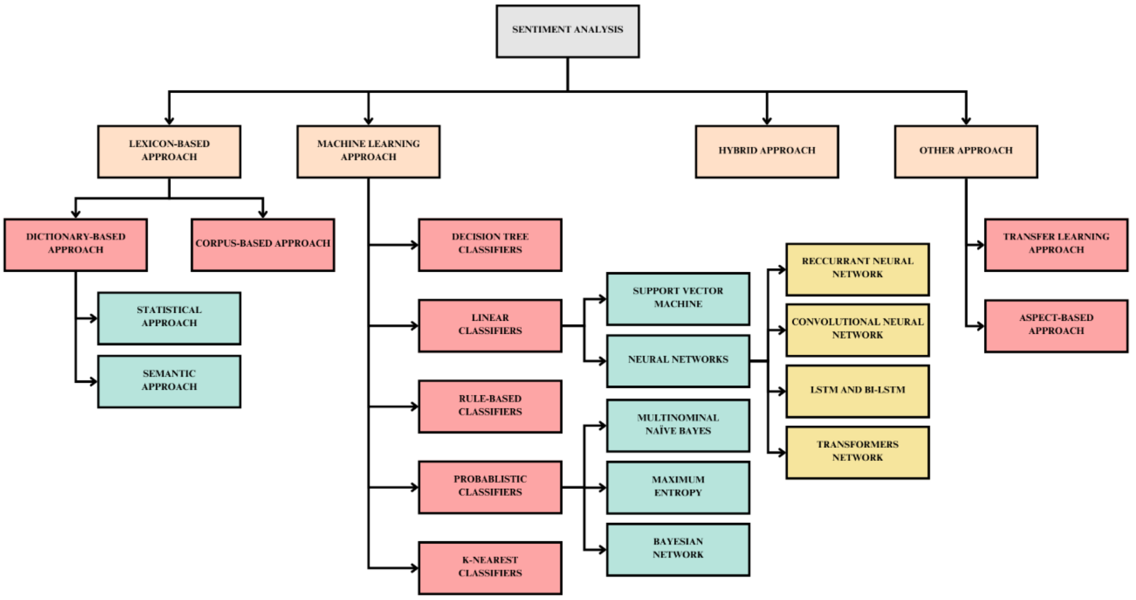


Figure 2.4 Technique in Sentiment Analysis

(Source: Wankhade et al., 2022*)*

I. Lexicon-Based Approach

Tokens are given predefined scores in the lexicon-based approach to sentiment analysis, indicating whether they are good, negative, or neutral. Scores might be anywhere from +1 and -1, corresponding to highly positive to highly negative sentiments. The method adds up the scores of each token, and the overall polarity of the text is decided based on the token with the highest score. Sentiment analysis at the sentence and feature levels is possible with this method because it is unsupervised and doesn't need training data. Its domain dependence, however, is its biggest drawback because words can have different meanings and sentiments depending on the context. Domain-specific sentiment lexicons can be created to address this issue, or already existing lexicons can be modified. Although the lexicon-based approach does not need training data, its domain specialisation prevents it from being applied to other domains (Mohammed Almosawi & Abdulbaki Mahmood, 2022).

II. Machine Learning Approach

Sentiment analysis, which involves determining and quantifying the sentiment of text or voice, makes extensive use of machine learning algorithms. Both supervised and unsupervised learning techniques can be used to achieve this. To train the algorithms and extract features from text input, supervised learning uses training data, producing accurate results. On the other hand, unsupervised learning techniques rely on databases, lexicons, and knowledge bases for sentiment analysis. Syntactic and linguistic factors are used in machine learning techniques to categorize sentiment by connecting textual data to certain class labels (Dhankhar et al., 2018). Common approaches include Naive Bayes, Support Vector Machine, Logistic Regression, Decision Tree, Maximum Entropy, K-Nearest Neighbours, and Semi-Supervised Learning (Wankhade et al., 2022). These algorithms increase the precision of sentiment analysis by allowing systems to comprehend contextual information, sarcasm, and incorrect word usage.

III. Hybrid Approach

Hybrid approach combines lexicon-based and machine learning techniques for sentiment analysis. This widely used method, which heavily relies on sentiment lexicons, combines the advantages of both approaches. To improve the outcomes of sentiment analysis, researchers have developed a variety of hybrid designs that combine both lexicon-based and automated learning techniques. For instance, (Hassonah et al., 2020) achieved good accuracy using a hybrid machine learning strategy that used SVM and feature selection algorithms. Other research has demonstrated how successful hybrid models are at enhancing sentiment analysis outcomes. The performance of hybrid models can still be improved by hyperparameter adjustment and architecture improvement, though, according to continuing research (Dang et al., 2021). Future research and development should focus on how hybrid models can perform better than standalone models and attain higher accuracy.

IV. Transfer Learning Approach

A pre-trained model can transfer its knowledge to a new model using the effective AI technique known as transfer learning. This method makes use of the similarities in data, distribution, and tasks. The new model can directly benefit without extensive training by using the previously acquired attributes. Its performance can be further enhanced by fine-tuning using training data. Transfer learning has been used in sentiment analysis to categorize sentiments across various fields. For instance, (Meng et al., 2019) created a transfer learning strategy combining fully connected layers that were modified and convolutional layers that had already been trained. The industry went through a transformation as a result of the launch of BERT, a revolutionary NLP model. Transformers and attention mechanisms are used by BERT to identify contextual relationships in text. It has demonstrated outstanding performance in a variety of tasks, including aspect detection(Li et al., 2019), sentiment analysis(Singh et al., 2021), and spam detection (AbdulNabi & Yaseen, 2021). The pre-training and fine-tuning phases of BERT are adaptable to different jobs. Additionally, BERT uses task-specific designs to handle single sentences as well as pairs of sentences. Sentiment analysis and other NLP tasks have generally advanced thanks to transfer learning and BERT.

V. Aspect-Based Approach

Aspect detection, sentiment categorization, and aggregation are the three main stages of aspect-based sentiment analysis (ASBA), a field within sentiment analysis that is rapidly expanding. Since it comes before sentiment analysis, aspect identification is essential in ASBA. The use of machine learning and NLP approaches allows for the implicit or explicit mining of aspects (Wankhade et al., 2022). ASBA is frequently used in hotel or product reviews to pinpoint specific points made by reviewers to address any negative sentiments related to them (Tran et al., 2019). This approach is beneficial to both consumers and producers. Aspect identification and classification problems in ASBA can be solved using advanced algorithms like LSTM, Bi-LSTM, or pre-trained models like BERT and GPT-2. Aspect detection is followed by polarity assigned to the mined aspects, which yields a total score that may be used to calculate the sentiment of the sentence overall similar to lexicon-based approach. For polarity assignments, a variety of techniques can be used, including dictionary-based methods and machine learning algorithms. Hard or soft voting is frequently used to evaluate the sentence's overall emotion. To evaluate customer sentiment and predict recommendation decisions, ASBA considers qualitative content, quantitative ratings, and cultural aspects (Jain et al., 2021).

2.5.3 Model Training

I. Naïve Bayes

The likelihood of a class given some observed attributes is determined by the probabilistic method Naïve Bayes, which is based on Bayes' Theorem. When it comes to text classification, the algorithm predicts the class or category to which a document will belong based on the words that are noticed in the document, or characteristics. Given the class of the document, the algorithm adopts the "naïve" assumption that each word's presence in a document is independent of the presence of other words. This assumption simplifies the probability computation and increases the computational efficiency of the procedure, particularly for big datasets with lots of words (Kim et al., 2018).

The Naïve Bayes algorithm for text categorization has various advantages, which are highlighted in the introduction. First of all, despite its simplifying presumption of word independence, it is renowned for being accurate and simple. It is therefore a well-liked option for numerous text classification problems. The algorithm's capacity to handle high-dimensional data, which is typical in text data with potentially large vocabulary, is an additional advantage. The curse of dimensionality, in which there may be a large number of words in the dataset relative to the number of samples, is less noticeable to Naïve Bayes. The technique is renowned for being simple to update incrementally as well. The algorithm may update its classification model without reprocessing previous training data when new documents are added to the training set. This functionality is very helpful in situations where the corpus of documents is always changing, like in online environments where new papers are added all the time (Kim et al., 2018).

II. Long Short-Term Memory

The weaknesses of traditional RNNs, which have trouble identifying long-term dependencies in sequential data, were intended to be addressed by the development of LSTM, a form of RNN. As LSTMs can retain and apply information over extended periods, they are well suited for processing and generating predictions based on data sequences, including text (Fu et al., 2018).

One-hot vector representations are frequently used in the conventional machine learning techniques for sentiment analysis, which might result in high-dimensional data and make it difficult to identify word correlations. On the other hand, word embeddings, such as Word2Vec and GloVe, can capture the syntactic and semantic links between words based on their contexts in big text corpora, and can represent words in a lower-dimensional space (Fu et al., 2018).

The potential of LSTM networks to organically model sequences is shown by their usage in sentiment analysis. Because of the arrangement of their memory cells, extensive sequences of information are retained by LSTMs, making them especially useful for processing and comprehending word sequences in text data. The introduction points out that, particularly when paired with word embeddings, LSTM-based models have emerged as the new standard for sentiment analysis (Fu et al., 2018).

There is additional discussion of the difficulties of applying word embeddings to sentiment analysis. Word embeddings have a few limitations, including the potential inability to effectively capture sentiment information due to their primary focus on modeling word contexts and potential inability to distinguish between words with opposite sentiments (e.g., "good" and "bad"). In order to increase word representation quality and capture more sentiment information, the introduction suggests a lexicon-enhanced LSTM model (LE-LSTM) that incorporates sentiment lexicon information into the LSTM (Fu et al., 2018).

LSTMs might display a bias in favor of recent inputs, which poses a hurdle when attempting to extract significant information from the start of a lengthy text sequence. In order to solve this, a new technique for calculating the attention vector in general sentiment analysis without a goal is proposed in the study. This technique seeks to extract global information from the entire text sequence and direct the LSTM to concentrate on significant portions of the text (Fu et al., 2018).

III. Recurrent Neural Network

An artificial neural network type called a recurrent neural network (RNN) is made to process sequential input by storing information in memory cells and using it to forecast or decide at each stage of the process. RNNs are ideally suited for tasks like text sentiment analysis in natural language processing, where the input data has a sequential or temporal structure (Cheng et al., 2020).

RNNs can be used to process word sequences in a sentence or document when text sentiment analysis is being performed, where the word order is important for understanding the meaning and sentiment of the text. RNNs preserve an internal state that records details about the sequence that has been viewed thus far, in contrast to conventional feedforward neural networks, which process input data independently of one another. RNNs are appropriate for tasks involving sequential data because of their internal state, which enables them to display temporal dynamic behavior (Cheng et al., 2020).

The capacity of RNNs to accommodate input sequences of varying length is one of its main advantages. Because of their adaptability, RNNs can handle inputs of various lengths, which makes them ideal for applications where the length of the input text can change, such as text sentiment analysis. However, vanishing gradients, an occurrence when the gradients used to update the network's parameters get incredibly small, causes slow or inefficient learning, especially over lengthy sequences, to be an issue for typical RNNs. Due to this constraint, conventional RNNs find it difficult to identify long-term relationships in sequential data, such as determining the sentiment of an extended text passage (Cheng et al., 2020).

As a specific RNN, Long Short-Term Memory (LSTM) networks were created to solve the problem of vanishing gradients. Compared to conventional RNNs, LSTMs are better able to capture long-term dependencies in sequential data because of their internal memory structure. For tasks like text sentiment analysis, where determining the sentiment of a text typically necessitates catching relationships between words that are far apart in the sequence, LSTMs are therefore especially well-suited (Cheng et al., 2020).

2.6 Related Works

2.6.1 Same Domain Different Technique

The process of categorising sentiments can be accomplished in sentiment analysis utilising a variety of methodologies within the same domain. Insights from text data are extracted, and techniques and methodologies are used to identify the sentiment contained within. Unfortunately, there aren't any research papers on lexicon-based, hybrid, or transfer learning approaches for reviewing clothing. Instead, this study selected a domain with a bigger scope which is product reviews on e-commerce websites.

Table 2.1 Same Domain Different Technique

Author & Year	Research Objectives	Methodology & Techniques	Sample Data	Finding/Results	Future Work /Recommendation
Michael Hakkinen, Ferry	To evaluate and use which methods for	Lexicon-based approach	The data from the top 10	Customer reviews are analysed using a lexicon of 1500	A semi-supervised method can be more effective for enhancing

Agustius Wong, Maria Susan Anggreainy, and Wahyu Raihan Hidayat (2021). (Hakkinen et al., 2021)	Tokopedia online retail sentiment analysis system evaluations.		Indonesian online retailers, including Shopee and Tokopedia.	words, and sentiment scores range from 0 to 100. Ranges of results are categorized, and a variable X is computed using terms that are positive and negative. A graph comparing the sentiment on Tokopedia with a competing platform shows that the sentiment on Tokopedia is consistent while the sentiment on the competitor platform is unpredictable.	sentiment analysis accuracy than merely depending on supervised or unsupervised learning algorithms. WordNet, for instance, can be used to find relevant words or viewpoints, while NLTK's probabilistic model, like Naive Bayes, can be used for sentiment analysis. The outcomes can be graphically displayed with the use of statistical analysis.
Masfiq Mahmud, Rafit Ahmed Mullick, and Muhammad Anas (2023). (Mahmud et al., 2023)	To analyse and predict the sentiment of customer reviews using machine learning algorithms and Natural Language Processing (NLP) techniques.	Machine learning approach	The dataset used in this study comprises 30,000 data points collected from an e-commerce platform in Bangladesh	The Random Forest Classifier identified as the best effective model for sentiment prediction in evaluations of women's clothing on e-commerce sites based on performance metrics with accuracy of 96%.	Explore other product categories and e-commerce sites to broaden the study. For improved performance, use pre-trained language models like BERT or GPT-3. Look at various deep learning methods, such as transformers and attention processes. To identify sentiment trends over time and comprehend the influence of product features and services, perform continuous analysis of customer

					reviews.
Vin Myca C. Sagarino, Jennen Isabelle M. Montejo, and Angie M. Ceniza-Canillo (2022). (Sagarino et al., 2022)	To do sentiment analysis on customer suggestions made in product reviews on Shopee Philippines.	Hybrid approach	This study collected every review from the top 10 products in each category, totalling 799 reviews that were saved in CSV file format.	Researchers discovered 83.6% positive reviews after using VADER to annotate cleansed data. SVM with VADER outperformed MNB with VADER, which had an accuracy of 82%, by achieving 83%. 79.5% of respondents said they would suggest a product based on positive ratings, while 82.7% said they wouldn't buy it if it had negative reviews.	Future work will include increasing the sample size and survey scope, automating translation, analysing emojis in reviews, extending the review collection to include products from other countries, clustering or classifying results based on demographic profiles, and interpreting the model by looking at learned vocabulary.
Ashra Sahar, Muhammad Ayoub, Shabir Hussain, Yang Yu, Akmal Khan (2022). Sahar et al., 2022)	To examine and predict customer reviews using Transfer learning (TL) approaches	Transfer learning approach	Majority of the data was acquired from comments on Amazon and Facebook.	By using the BERT for sentiment analysis, the study achieved the highest accuracy of 93.21%.	There is no future recommendation or future suggested in the paper.

2.7 Same Technique Different Domain

The same technique can be used in sentiment analysis to analyse and classify sentiments across many domains. The aspect-based approach was chosen for this study's sentiment analysis of clothing review. This comparison enables the use of a consistent methodology while changing to the specific characteristics and complexities of each domain, enabling a deeper comprehension of sentiment across various application domains.

Table 2.2 Same Technique Different Domain

Author & Year	Research Objectives	Sample Data	Finding/Results	Future Work /Recommendation
Brentton Wong Swee Kit and Minnu Helen Joseph (2023). (Kit & Joseph, 2023)	The project's goal is to be able to recognize the most important aspects of the movie review that are discussed in the review as well as its polarity.	- IMDB movie review dataset -Total of 50,000 rows	Every model that was used had a promising accuracy of 90% or higher. In order to choose the best models for aspect prediction and sentiment analysis, the team also compared the various models.	Future improvements to the project can be made by allowing multiple aspects to be reviewed. This is so because reviews typically cover a wide range of topics.
Jason Nguyen and Ritu Chaturvedi (2020). (Nguyen & Chaturvedi, 2020)	The purpose of this study is to analyse public opinion on Twitter on the recent coronavirus outbreak.	Three corpora of Tweets were gathered, each including roughly 800,000 microblog posts (Tweets).	Most opinions on several prominent topics are generally unfavorable. There is inherent risk in indexing the facet "corona" due to its ambiguity, as astronomers and beer lovers may have differing perspectives. Interestingly, "corona" shares sentimental	- Develop/use a larger and/or more recent dataset of Twitter posts. - Use techniques other than Naive Bayes for data mining. - Examine the fastText library, which is positioned as a replacement for Facebook's AI research lab's Word2Vec.

			characteristics with "covid" and "coronavirus."	
Dr.K.A.Waghmare and Sheetal K. Bhala(2020). (Waghmare & Bhala, 2020)	- to research the three aspect-based categories for sentiment classification. - to clarify the tourist reviews into positive and negative feelings.	These reviews are collected from social media platforms.	The study using deep neural networks, which perform more accurately and efficiently when categorizing and extracting sentiments.	There is no future recommendation or future suggested in the paper.
Hrithika Yadav, Kartik Dwivedi, G. Abirami(2022). (Yadav et al., 2022)	The objective of this study is to extract significant aspects from user reviews, and the overall sentiment is categorized.	- Data was extracted from Glassdoor, a review site for business employees. - Utilize feedback from current and past Google employees. - The data collection yielded 7340 reviews,	Decision Trees are shown to have the lowest accuracy (80.58%). The accuracy of the Multinomial Naive Bayes and Support Vector Machines classifiers is good (84.87 and 83.85 percent, respectively). The results show that Random Forest produces the best accuracy (88.07%).	- By developing rules that address negations, intensifiers, downtowners, and other aspects that alter attitudes, it is suggested that this technique be improved. - Intend to expand the study to cover ordinal classes.
Ian Michael Urriza and Maria Art Antonette Clarino (2021). (Urriza & Antonette Clarino, 2021)	The main goal of this research is to create an algorithm that can summarize Steam user reviews of titles.	There were 1000 reviews collected for each game	From evaluations that were obtained from Steam, this study was able to give a bi-level classification in terms of aspect and polarity. Using an SVM classifier, it	- Making use of a different kind of data set, such as games in Steam's Early Access. - To filter out reviews using the new Steam Review Awards.

			was able to differentiate between each sentiment for several game-related factors.	

2.8 Summary

In conclusion, this chapter has provided an overview of the subjects covered by the research domain, relevant methodologies, and related literature to this research study. To gather all pertinent knowledge and information, articles, journals, websites, and research papers were examined and evaluated; the results were then documented in writing and included in this chapter. All this knowledge and information will be helpful as we move forward with the following parts of this research study.

CHAPTER THREE
RESEARCH METHODOLOGY

The section describes the steps taken to address the identified problems and recommendations. It provides detailed explanations of the phases, activities, sources, and deliverables involved. Each phase's requirements and its deliverables are thoroughly explained in the following section.

3.1 Research Design & System Architecture

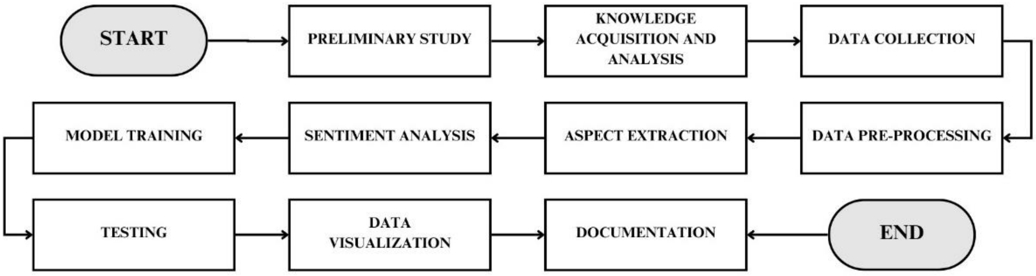


Figure 3.1 Research Design Architecture

The research design served as the framework for all the research approaches and techniques chosen to conduct the study. This design allowed the study to focus on creating research methods relevant to the subject and to set up the studies for success, particularly in achieving the objectives, phases, activities, sources, and deliverables specific to each part.

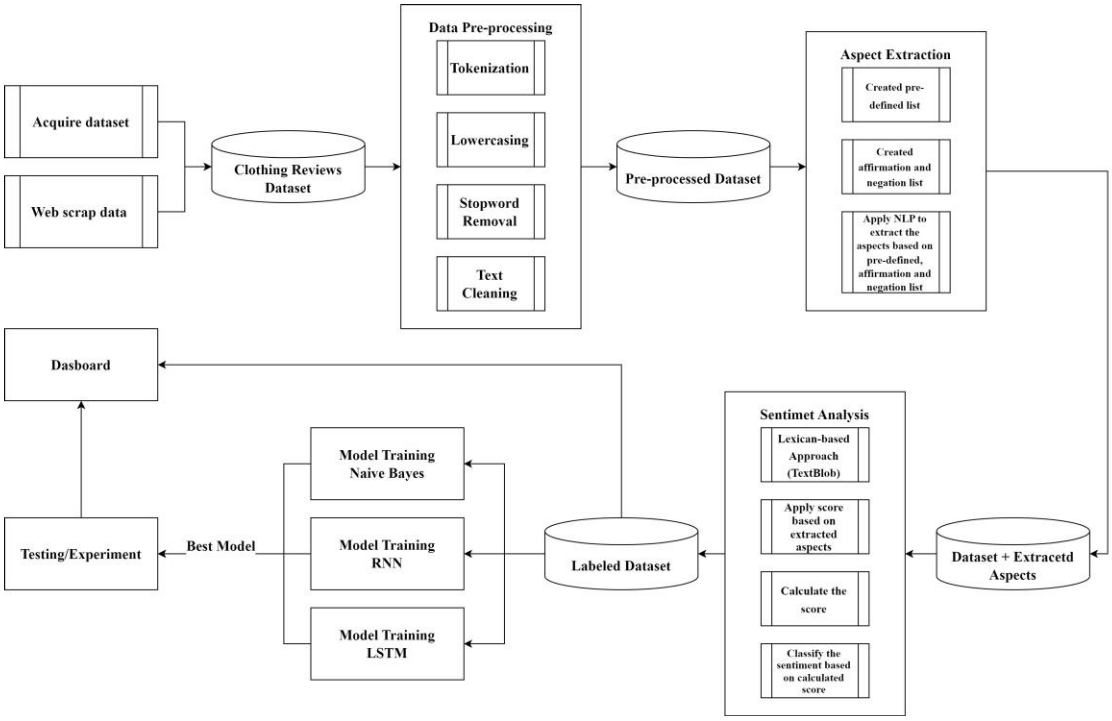


Figure 3.2 System Architecture

The system architecture provided an overview of the structure and components of a system, illustrating how they interacted and functioned together to achieve the study's objectives. The architecture served as a blueprint for the system's development and implementation, guiding the design and integration of its various components to ensure optimal performance. It also helped in understanding how the system would collect data, process data, and deliver outputs, providing a comprehensive view of its inner workings.

3.2 Preliminary Study

Table 3.1 Preliminary Study Table

Objectives	Phase	Activities	Sources	Deliverable
To identify the key aspects of clothing reviews.	Preliminary Study	Read and understood articles related to sentiment analysis and key aspects of clothing reviews.	Journal articles, books, thesis, and websites	• Background study • Problem Statement • Research Question • Project Scope • Significance

A preliminary study was a first investigation carried out to gather data and determine whether an overall research attempt was possible. Before beginning the primary study, it helped in gaining an understanding, seeing potential problems, and adjusting the research goals and procedures. This included reading reviews of related literature, investigating data availability, creating research tools, and compiling preliminary data.

3.3 Knowledge Acquisition and Analysis

Table 3.2 Research Methodology Table of Knowledge Acquisition and Analysis

Objectives	Phase	Activities	Sources	Deliverable
To identify the key aspects of clothing reviews.	Knowledge Acquisition and Analysis	• Conducted an in-depth literature review to identify the key aspects focused on clothing reviews. • Gained knowledge of the current research on sentiment analysis methods used in apparel reviews. • Compared different sentiment analysis methods that had been applied in the literature and evaluated their use and efficiency for the study of apparel reviews. • Selected the most suitable and best-performing sentiment analysis technique based on the evaluation and alignment with the research objectives.	Journal articles, books, thesis, and websites	• Mind Map • Literature Review

Knowledge acquisition was the process of extracting, classifying, and analyzing data from various sources, including Google Scholar, IEEE Xplore, Scopus, Web of Science, and articles or papers from open sources. The problem was determined together with its goal, range, and importance related to the connected subject by applying the literature analysis technique. It referred to the process of gathering all the facts and information. Earlier studies carried out by other analysts were examined in the evaluation of the literature, which helped a better understand the research topic. Since each problem could be presented and solved in a variety of ways, this phase enabled the acquisition of new knowledge. This phase supported the ways, strategies, or method algorithms that were chosen and used on this project based on their efficiency and accuracy in carrying out the relevant article. The analysis of the project, method, and approaches could also be decided based on how they were described in the prior research and on how to carry out each of the operations.

3.4 Data Collection

Table 3.3 Research Methodology Table of Data Collection

Objectives	**Phase**	**Activities**	**Sources**	**Deliverable**
To apply the Aspect-Based Sentiment Analysis for clothing reviews.	Data Collection	• Acquired the dataset from data.mendeley.com. • Web scraped POKOKS websites for Malaysian cultural outfits. • Additional data were added from dataset of Amazon's clothing products. • Analyzed the collected raw data.	Websites and Chrome extension for web scrap	Data of clothing reviews

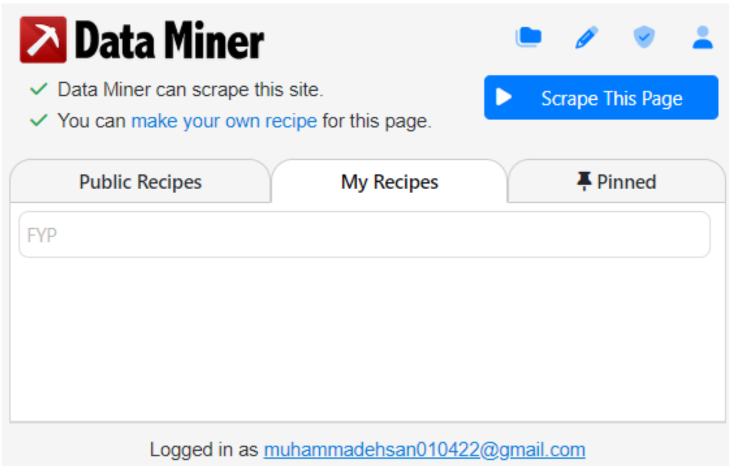
For data collection, the data was initially gathered from the website of data.mendeley.com with total data of 23,452, providing a foundational dataset which consisted of ten columns described as follows:

Table 3.4 Dataset Description

Column Name	**Description**
Clothing ID	An identifier for each clothing item in the dataset.
Age	The age of the reviewer who provided the review for the clothing item.
Title	The title of the review provided by the reviewer.
Review Text	The main body of the review text that contains detailed information about the reviewer's experience with the clothing item.
Rating	The numerical rating given by the reviewer to the clothing item, typically on a scale from 1 to 5.
Recommended IND	A binary indicator (0 or 1) indicating whether the reviewer recommended the clothing item to others.
Positive Feedback Count	The count of positive feedback received from other users

	on the review.
Division Name	The name of the clothes division whether it is general, general petite, or intimates
Department Name	The name of the clothes division which categorizes the clothes type whether it is tops, bottoms, dresses, jackets, and intimates.
Class Name	The details of clothes type based on department name such as pant, knit, blouses, pant, etc

Additional data was obtained by web scraping the pokoks.com website in order to improve this information with a focus on Malaysian national attire, such as Baju Melayu and Baju Kurung. In order to provide particular insights about Malaysian traditional apparel to the dataset, this approach involves extracting clothing reviews from the website using a Chrome extension called Data Scraper.



A unique recipe was needed for web scraping in order to use Data Scraper because different websites had distinct HTML and CSS codes. This included designing a customized set of guidelines that indicated where the required data items should be placed on the website. The Data Scraper tool could easily retrieve the intended information for additional analysis and inclusion into the research dataset by picking out the pertinent HTML or CSS code segments, like particular navigation components. Overall, 212 data instances, including Review Text, Rating, and Class Name, were scraped. Additionally, to add variety to the clothes categories, 10,082 reviews were collected from Amazon's clothing reviews dataset that were acquired through Kaggle.com websites and added to the dataset.

Table 3.5 Source and Total Data

Source	Data	Total Data
https://data.mendeley.com/datasets/7wmpsc8msm/1/files/23fc7013-dbb0-4417-a903-2a91d4257cd4	Dataset	23,452
https://pokoks.com/products/the-tanah-men-baju-melayu-navy-blue	Web scrap	212
https://www.kaggle.com/datasets/jocelyndumlao/consumer-review-of-clothing-product/	Dataset	10,082
	TOTAL	**34,573**

3.5 Data Pre-processing

Table 3.6 Research Methodology Table of Data Pre-processing

Objectives	Phase	Activities	Sources	Deliverable
To apply the Aspect-Based Sentiment Analysis for clothing reviews.	Data Preprocessing	Data cleaning by: • Tokenization • Lowercasing • Stopword Removal • Text Cleaning	Anaconda, Jupyter Notebook and Python Libraries	Acquired a clean data set that related to clothing reviews

The dataset was carefully examined in the beginning to identify any inconsistencies or missing data. Instances related to missing values were removed to maintain data integrity. The text processing approach involved multiple phases for the apparel reviews dataset. The text was first tokenization or separated into discrete words or tokens. To guarantee consistency and get rid of case-related issues, all text was then changed to lowercase. The next step was to eliminate common, non-informative words using stopword removal. Eliminating punctuation, numerals, special characters, and other unnecessary elements was known as text cleaning.

Furthermore, lemmatization and stemming reduced words to their simplest form. The quality of the dataset was raised and phrases with comparable meanings were combined thanks to this reduction in complexity. Further partitioning of the dataset into training and testing sets allowed for an objective assessment of the sentiment analysis model's performance.

The missing values handling and text processing methods (tokenization, lowercasing, stopword removal, and text cleaning) that were part of the data pre-processing phase changed the format of the clothing reviews dataset so that it could be used for aspects extraction and sentiment analysis phases. This phase was to made sure that the dataset was ready for additional modelling and analysis, which made it possible to extract useful data from customer evaluations.

3.6 Aspect Extraction

Table 3.7 Research Methodology Table of Aspect Extraction

Objectives	**Phase**	**Activities**	**Sources**	**Deliverable**
To apply the Aspect-Based Sentiment Analysis for clothing reviews.	Aspect Extraction	Applied techniques to identify and extract specific aspects mentioned in the clothing reviews.	Anaconda, Jupyter Notebook and Python Libraries	Extracted aspects.

In aspect-based sentiment analysis, specific results or findings are produced based on the analysis carried out in previous stages. These deliverables offer insightful understandings of the elements highlighted in the reviews and their corresponding sentiment. The extracted aspects, which are the characteristics or properties determined from the text data, are among the key outputs. To give these aspects more context, they are frequently categorized and given labels. Specific words, phrases, or noun phrases that describe the aspects being discussed, such as fabric quality or price, might be used as these aspects. The next step is to classify or categorize each extracted aspect to add further context. This helps classifying and organizing the aspects for analysis.

3.7 Sentiment Analysis

Table 3.8 Research Methodology Table of Sentiment Analysis

Objectives	Phase	Activities	Sources	Deliverable
To apply the Aspect-Based Sentiment Analysis for clothing reviews.	Sentiment Analysis	• Developed an aspect-based sentiment analysis using the Sentiment Lexicon approach to analyze clothing reviews from different aspects. • By adding up the sentiment scores of the words or phrases linked to each aspect, determined the overall sentiment of each aspect.	Anaconda, Jupyter Notebook and Python Libraries	Labeled dataset with regards to the polarity of its sentiment: - Positive - Negative - Neutral

Using the TextBlob library, a sentiment lexicon must be created in the first step. TextBlob is a flexible Python text processing toolkit with integrated sentiment analysis features. The pre-built sentiment lexicon in the library can be used, or it can be customized to include words and phrases specific to the specific domain. After the aspects extraction phase are determined, TextBlob's sentiment analysis features are implimented. Text can be given polarity scores by TextBlob, which indicates if each aspect of the text has a good, negative, or neutral sentiment attached to it. In order to offer a more detailed examination, aspect-level feelings are computed. In order to do this, the polarity scores of the words or phrases associated with each aspect are combined, usually using weighted average or average approaches. The result is an in-depth understanding of the emotions connected to every single aspect.

3.8 Model Training

Table 3.9 Research Methodology Table of Model Training

Objectives	Phase	Activities	Sources	Deliverable
To apply the Aspect-Based Sentiment Analysis for clothing reviews.	Model Training	• RNN • LSTM • Naïve Bayes	Anaconda, Jupyter Notebook and Python Libraries	The accuracy of each model

Recurrent neural networks (RNN), long short-term memory (LSTM) networks, and Naïve Bayes were to be implemented for comparative analysis in aspect-based sentiment analysis of clothing evaluations during the development of the model phase. The reason for integrating RNN and LSTM was the ability to handle sequential data effectively and identify dependencies in text reviews, which are critical for understanding complex sentiment expressions. Since Naive Bayes is a well-known probabilistic classifier with a reputation for being straightforward and effective at managing huge feature spaces, such as those present in text data, it was included in the analysis. Standard sentiment analysis criteria, including recall, accuracy, precision, and F1 score, were used to assess performance. To guarantee robustness, cross-validation procedures were also implemented. Statistical tests were also employed to compare the models and identify any noteworthy variations. It was anticipated that this thorough approach would provide insights into the relative benefits of different models in this specific case.

3.9 Testing

Table 3.10 Research Methodology Table of Testing

Objectives	Phase	Activities	Sources	Deliverable
To apply the Aspect-Based Sentiment Analysis for clothing reviews.	Testing	Evaluated and compared the accuracy of the dataset for clothing sentiment analysis.	Anaconda, Jupyter Notebook and Python Libraries	The accuracy of each experimented models and simple prototype using trained model for sentiment analysis

The model that was evaluated most effectively in terms of accuracy was chosen for additional experiments. The primary aim was to optimize performance by adjusting each of the three primary training parameters: which is batch sizes, epochs, and optimizers. The goal was to achieve the best possible balance between overfitting and model convergence by varying the number of epochs, which determined how many times the model was trained on the whole dataset. Similarly, by adjusting the batch sizes, the number of samples processed during each training iteration impacted the model's performance, affecting both generalization and training speed. Furthermore, experimenting with other optimizers such as Adamax and Adam gave insights into how they affected the convergence rate and general accuracy of the model. Additionally, a prototype was created to enable sentiment analysis model testing. With the help of this prototype, users may input text for analysis and get predictions about its sentiment which is positive, negative, or neutral. The goal was to improve the model's usability and robustness through this iterative testing and refining process, ensuring its efficacy in practical applications.

3.10 Data Visualization

Table 3.11 Research Methodology Table of Data Visualization

Objectives	Phase	Activities	Sources	Deliverable
To develop a dashboard for Aspect-based Sentiment Analysis of Clothing Reviews.	Data Visualization	• Developed a dashboard. • Displayed the overall results and insights in a dashboard by using proper tools for data visualization.	Power BI	Dashboard showing the results of sentiment analysis visualization.

The dashboard, which was developed using Power BI, utilized a dataset that had been labelled using the aspect-based sentiment analysis method in sentiment analysis phase. This allowed for a clear visualization of aspect-based attitudes in clothing reviews. To illustrate the distribution of sentiments across aspects, various visualization methods such as bar charts, pie charts, stacked bar charts, and etc were employed. The user interface presented the sentiment analysis results in a simple and accessible manner, enabling users to easily interpret the data. Overall, the dashboard provided a clear and straightforward visualization of the sentiment analysis results, facilitating a comprehensive understanding of the attitudes expressed in the clothing reviews.

3.11 Documentation

Table 3.12 Research Methodology Table of Documentation

Objectives	Phase	Activities	Sources	Deliverable
None	Documentation	Wrote a report for all the phases.	Microsoft Word, Mendeley	Full report

The documentation process, the final stage of the project, involved assembling all the results and data into a single report. The entire project's methods and steps were covered in detail and portrayed from the beginning of the chapter until its end. It also contained all the tools, data, techniques, approaches, analyses, and other essential components needed to accomplish the project's objectives.

3.12 Summary

This chapter constructed each step in accordance with the specialized methodologies, which were objective and systematic to finish the entire project effectively, meeting the three keys of the project's objectives. The following phases were preliminary study, knowledge acquisition & analysis, data collection, data pre-processing, aspects extraction, sentiment analysis, model training, testing, data visualization, and finally, documentation.

CHAPTER FOUR
RESULT AND FINDING

The aspect-based sentiment analysis results on apparel evaluations were reported in this chapter. The study aimed to investigate customers' opinions on multiple aspects of clothes, such as fabric quality, style, fit, color, pattern, cost, comfort, and longevity. By breaking the evaluations into each component and analyzing the sentiments around each, important insights into consumer preferences and perceptions of clothing products were gained from this chapter.

4.1 Aspects of Clothing

The first objective of this study, which aimed to identify the key aspects of clothing reviews, has been successfully accomplished through the Knowledge Acquisition and Analysis phase. As a result of this phase, the study has identified nine crucial aspects that are commonly mentioned in clothing reviews. These aspects are fabric, design, size, color, price, comfort, quality, durability, and service. The identification of these aspects is a significant achievement as they will serve as the foundation for the later phase of the study, which involves extracting these aspects from the reviews. By creating a predefined list based on these aspects, the study will be able to effectively extract and categorize the aspects mentioned in the reviews. This categorization will then be used for sentiment analysis based on the extracted aspects, providing valuable insights into the sentiments associated with each aspect of clothing reviews. The table provided in the study outlines these aspects and their references, forming a comprehensive framework for the subsequent phases of the research.

Table 4.1 The Aspects of Clothing

Aspects	Reference
Fabric	(Eckman et al., 2020)
Design	(Eckman et al., 2020), (Rahman, 2017)
Size	(Rahman, 2017)
Color	(Eckman et al., 2020)

Price	,(Permana & Dewi, 2020) (Chang, 2020)
Comfort	(Permana & Dewi, 2020), (Eckman et al., 2020)
Quality	(Eckman et al., 2020)
Durability	(Rahman, 2017)
Service	(Chang, 2020)

4.2 Data Pre-processing

The suitable Python libraries for preparing data were imported by the code. To transorm data, pandas (imported as pd) was utilized, and nltk provided natural language processing techniques including tokenization and stop word removal. Regular expressions were handled by the re module, and language recognition was done by langdetect. Together, these libraries made it possible to do operations that are necessary for prepping text data for additional analysis or modeling, such as language filtering, text cleaning, and word tokenization.

```
# Load the dataset in chunks
file_path = 'clothing_reviews.csv'

try:
    chunk_size = 1000
    data_chunks = pd.read_csv(file_path, chunksize=chunk_size)
except Exception as e:
    print(f"An error occurred while reading the file: {e}")
    data_chunks = []
```

Figure 4.1 Load the Dataset

In this part of the code, a file path to the combined dataset was specified. The dataset was then read in chunks of 1000 rows using Pandas' read_csv function with the chunksize parameter. An exception was handled in case there were any issues with reading the file.

```
# Initialize an empty DataFrame to store the preprocessed data
preprocessed_data = pd.DataFrame()
```

Figure 4.2 Initialized DataFrame

An empty Pandas DataFrame called preprocessed_data is initialized to store the preprocessed data.

```python
# Function to detect the language of a text
def detect_language(text):
    try:
        lang = detect(text)
        return lang == 'en'
    except:
        return False
```

Figure 4.3 Fuction of detect_language

The detect_language function is defined to detect the language of a given text. It uses the langdetect library to identify if the text is in English.

```python
# Process and filter rows with English text in chunks
for chunk in tqdm(data_chunks, desc="Processing Chunks"):
    if not chunk.empty:
        chunk['is_english'] = chunk['Review'].apply(detect_language)
        chunk = chunk[chunk['is_english']]

        # Remove duplicate reviews
        chunk.drop_duplicates(subset='Review', keep='first', inplace=True)

        # Tokenize the cleaned review and store the tokens as lists
        stop_words = set(stopwords.words('english'))

        def preprocess_text(text):
            # Lowercasing and stopword removal
            tokens = word_tokenize(text)
            filtered_tokens = [token.lower() for token in tokens if token.isalnum() and token.lower() not in stop_words]
            return filtered_tokens

        chunk['Tokenized_Review'] = chunk['Review'].apply(preprocess_text)

        # Continue with text cleaning and emoji removal for the cleaned review
        def clean_text(text):
            text = re.sub(r'[^a-zA-Z\s]', '', text)
            text = re.sub(r'\s+', ' ', text).strip()
            emoji_pattern = re.compile("["
                                       u"\U0001F600-\U0001F64F"  # emoticons
                                       # ... (emoji ranges, if needed)
                                       "]+", flags=re.UNICODE)
            text = emoji_pattern.sub(r'', text)
            return text

        chunk['Review'] = chunk['Review'].apply(clean_text)

        # Convert the list of tokens to a space-separated string
        chunk['Tokenized_Review'] = chunk['Tokenized_Review'].apply(' '.join)

        # Append the preprocessed chunk to the new DataFrame
        preprocessed_data = pd.concat([preprocessed_data, chunk])
```

Figure 4.4 Process and Filter Row

To handle huge datasets efficiently, the data was filtered and processed in chunks. Using a for loop, the code iterated through each chunk of the dataset, which represented a subset of the full dataset. The code first determined whether the chunk was not empty inside the loop, indicating that there was data to handle.

The code then used the language recognition function (detect_language) to find English text in each review in the chunk. In doing so, the dataset's non-English reviews were filtered out. Non-English reviews were eliminated, and reviews that were classified as English were kept in the chunk.

To make sure that every review was distinct, the code first filtered for reviews written in English before removing any duplicates from the chunk. By preventing redundancy, this assisted in preserving the dataset's integrity.

Next, each review's text was tokenized using the nltk library's word_tokenize function. The process of tokenization entailed dividing the text into discrete words, or tokens. After that, the tokens were transformed to lowercase and filtered using a set of predefined stopwords for the English language to eliminate any non-alphanumeric letters and stopwords (common words like "the," "is," etc.). A list of each review's filtered tokens was the end result, and these were kept in a new column named Tokenized_Review.

The code then used regular expressions to remove all non-alphabetic characters (except from spaces) and emojis from the reviews in order to carry out further text cleaning. By taking this step, the text was more consistent and any non-textual elements that would not be pertinent to the analysis were removed.

The tokenized review list was then saved in the Tokenized_Review column after connecting back together into a single string with a space as a separator. A new DataFrame (preprocessed_data) containing the preprocessed reviews from all chunks processed thus far was created by concatenating the preprocessed chunk with the preprocessed data that already existed. This process ensured that the preprocessing steps were applied consistently to each chunk of the dataset, and the results were aggregated into a single DataFrame for further analysis.

1	Id	Review	Rating	Class Nam	Label(Based on Rating)	Tokenized_ReviewTokenized_ReviewTokenized_ReviewTokenized_Review
2	0	Absolutely wonderful silky and sexy and com	4	Intimates	positive	absolutely wonderful silky sexy comfortable
3	1	Love this dress its sooo pretty i happened to	5	Dresses	positive	love dress sooo pretty happened find store glad bc never would ordered onlin
4	2	I had such high hopes for this dress and reall	3	Dresses	neutral	high hopes dress really wanted work initially ordered petite small usual size fo
5	3	I love love love this jumpsuit its fun flirty and	5	Pants	positive	love love love jumpsuit fun flirty fabulous every time wear get nothing great c
6	4	This shirt is very flattering to all due to the a	5	Blouses	positive	shirt flattering due adjustable front tie perfect length wear leggings sleeveless
7	5	I love tracy reese dresses but this one is not	2	Dresses	negative	love tracy reese dresses one petite 5 feet tall usually wear 0p brand dress pret
8	6	I aded this in my basket at hte last mintue to	5	Knits	positive	aded basket hte last mintue see would look like person store pick went teh da
9	7	I ordered this in carbon for store pick up and	4	Knits	positive	ordered carbon store pick ton stuff always try used top pair skirts pants every
10	8	I love this dress i usually get an xs but it runs	5	Dresses	positive	love dress usually get xs runs little snug bust ordered size flattering feminine u
11	9	Im and lbs i ordered the s petite to make sur	5	Dresses	positive	5 5 125 lbs ordered petite make sure length long typically wear xs regular retai
12	10	Dress runs small esp where the zipper area r	3	Dresses	neutral	dress runs small esp zipper area runs ordered sp typically fits tight material top
13	11	This dress is perfection so pretty and flatteri	5	Dresses	positive	dress perfection pretty flatteringdress perfection pretty flattering
14	12	More and more i find myself reliant on the r	5	Dresses	positive	find reliant reviews written savvy shoppers past right estimation product case
15	13	Bought the black xs to go under the larkspur	5	Intimates	positive	bought black xs go larkspur midi dress bother lining skirt portion grrrrrrrrrrr sta
16	14	This is a nice choice for holiday gatherings i l	3	Dresses	neutral	nice choice holiday gatherings like length grazes knee conservative enough off
17	15	I took these out of the package and wanted	4	Pants	positive	took package wanted fit badly could tell put would figure straight waist way s
18	16	Material and color is nice the leg opening is	3	Pants	neutral	material color nice leg opening large 5 1 100 length hits right ankle leg opening
19	17	Took a chance on this blouse and so glad i di	5	Blouses	positive	took chance blouse glad crazy blouse photographed model paired whit white
20	18	A flattering super cozy coat will work well fo	5	Outerwear	positive	flattering super cozy coat work well cold dry days look good jeans dressier out
21	19	I love the look and feel of this tulle dress i w	5	Dresses	positive	love look feel tulle dress looking something different top new year eve small c
22	20	If this product was in petite i would get the p	4	Blouses	positive	product petite would get petite regular little long tailor simple fix fits nicely 5
23	21	Im upset because for the price of the dress i	4	Dresses	positive	upset price dress thought embroidered print fabric think cried little opened bo
24	22	First of all this is not pullover styling there is	2	Dresses	negative	first pullover styling side zipper would purchased knew side zipper large bust si
25	23	Cute little dress fits tts it is a little high waist	3	Dresses	neutral	cute little dress fits tts little high waisted good length 5 9 height like dress love
26	24	I love this shirt because when i first saw it i w	5	Blouses	positive	love shirt first saw sure shirt dress since wear like dress need slip wear legging
27	25	Loved the material but i didnt really look at	3	Lounge	neutral	loved material didnt really look long dress purchased large medium im 5 5 atle

Figure 4.5 Pre-processed Data

Finally, the pre-processed data is saved to a CSV file named 'preprocessed_data.csv' without including the index column. This code essentially processed the combined dataset by removing non-English reviews, removing duplicates, tokenizing and cleaning the text, and then concatenated the preprocessed chunks into a single DataFrame for further analysis In conclusion, the data processing resulted in a reduction in the number of data points from 34573 to 32385, representing a comprehensive refinement of the dataset.

4.3 Aspect Extraction

The aspect extraction and visualization in the code made use of multiple Python libraries. For data manipulation and analysis, Pandas (imported as pd) was used, especially when reading preprocessed data from a CSV file into a DataFrame. Tokenization and the development of unique matching patterns for aspect extraction were two text-processing activities that were accomplished with the help of the natural language processing (NLP) library spaCy. Using user-defined patterns, the PhraseMatcher class from spaCy's matcher module was used to effectively match textual token sequences. The aspect extraction process was visualized using tqdm, a package for making progress bars. A word cloud visualization showing the frequency of extracted attributes was also produced using the WordCloud class from the WordCloud library. Finally, Matplotlib's pyplot (imported as plt) was used to display the generated word cloud visualization.

```
# Load spaCy model
nlp = spacy.load('en_core_web_sm')
```

Figure 4.6 Load spaCy Model

Here, the code loads the spaCy English language model en_core_web_sm. This model is used for various natural language processing tasks such as tokenization, part-of-speech tagging, and named entity recognition.

```
# Sample text from the dataset
sample_text = data['Tokenized_Review'].iloc[0]
```

Figure 4.7 Retrieves Tokenized Review

After load the csv file, this line retrieves the tokenized review text from the first row of the dataset and assigns it to the variable sample_text. This sample text will be used to demonstrate the aspect extraction process.

Table 4.2 Pre-defined List of Aspects

Aspects	Positve	Neural	Negative
Fabric	soft, breathable, smooth, silky, luxurious, plush, comfortable, cozy, warm	lightweight, stiff, heavy, airy, thin, thick, absorbent, easily stretched	itchy, irritating, scratchy, durable, terrible, seethrough, unsewn, overstretched
Design	stylish, elegant, trendy, unique, modern, beautiful, fashionable, cute, pretty	simple, classic, plain, basic, standard, ordinary, subtle	tacky, ugly, outdated, unattractive, boring, poorly designed, flaw
Size	perfect, comfy, flattering, fitting, roomy, slim, shapeless, proportional, nice oversized	loose, tight, bulky	baggy, inconsistent, awkward, oversized, undersized, snug, long, big, tight, small, short
Color	vibrant, beautiful, eye-catching, rich, vivid, multicolored, bright, pastel, gorgeous, nice	neutral, monochromatic, subdued	faded, dull, mismatched, off-color, true-to-color
Price	affordable, budget friendly, reasonably, bargain, inexpensive, economical	standard	expensive, overpriced, pricey, costly, steep, premium
Comfort	cozy, soft, breathable, comfortable, relaxing, cushioned, smooth, comfy, easy-to-wear,	adequate, acceptable, satisfactory, standard, tolerable	restrictive, itchy, tight, irritating, scratchy

	supportive, luxurious		
Quality	excellent, superior, premium, exceptional, reliable, top-notch, outstanding, durable, high grade, fine, good quality, high quality	average, standard, acceptable, satisfactory, typical	poor, inferior, substandard, low quality, flimsy, cheap, faulty, bad quality, tacky, deteriorating, unsewn, low quality
Durability	durable, sturdy, high-quality, long-lasting, lasting, strong, resilient, robust, solid	average, typical, standard, acceptable	low-quality, flimsy, easily damaged, wears-out-quickly, fragile, shoddy, rugged, tearing, ripped
Service	helpful, courteous, responsive, professional	average, standard, satisfactory	poor, unprofessional, slow, disappoint

The attributes dictionary contains pre-defined attributes related to clothing reviews, along with lists of positive, neutral, and negative words associated with each attribute. These words will be used to match aspects in the reviews. The aspects were obtained from the knowledge acquisition and analysis phase, which included fabric, design, size, color, price, comfort, quality, durability, and service. Descriptive words for these aspects were derived from ChatGPT to enrich the attribute dictionary for more comprehensive aspect matching during sentiment analysis.

Table 4.3 List of Affirmation and Negation Words

Affirmation	good, great, excellent, positive, high, amazing, awesome, fantastic, superb, wonderful, pleasing, satisfying, delightful, terrific, impressive
Negation	not, no, never, none, neither, low, cheap, too, poor, bad, negative, disappointing, inferior, unsatisfactory, lacking, subpar, unimpressive, terrible

Next, the lists affirmation_words and negation_words were defined. In natural language processing (NLP) tasks, these lists of words, which convey positive or negative emotion respectively, are frequently employed to determine the feelings present in text. These lists were expanded with additional words sourced from ChatGPT to enhance the sentiment analysis capabilities of the model.

```
# Create PhraseMatcher instances for clothing attributes
attribute_matchers = {key: PhraseMatcher(nlp.vocab) for key in attributes}
for key, labels in attributes.items():
    patterns = [nlp(text) for text in labels['positive'] + labels['neutral'] + labels['negative']]
    attribute_matchers[key].add(key, None, *patterns)
```

Figure 4.8 Attribute Matchers

This code created an attribute_matchers dictionary, where each key was associated with an attribute found in the attributes dictionary. Using the vocabulary from spaCy, a PhraseMatcher instance was made for every key. This loop added the positive, neutral, and negative patterns for each attribute to the matching PhraseMatcher instance by iterating through the attributes dictionary. The nlp pipeline was used to transform each pattern into a spaCy Doc object.

```
# Function to find aspects and check for affirmation/negation
def find_attributes(text):
    doc = nlp(text)
    matched_aspects = set()  # Use a set to store unique aspects
    matched_categories = set()  # Use a set to store unique categories
    for key, matcher in attribute_matchers.items():
        matches = matcher(doc)
        for _, start, end in matches:
            aspect_span = doc[start:end]
            # Normalize aspect word to its base form (lemma)
            aspect = aspect_span.lemma_
            # Look for preceding and following tokens
            prev_token = doc[start - 1] if start > 0 else None
            next_token = doc[end] if end < len(doc) else None
            # Check for affirmation or negation words
            if prev_token and prev_token.lower_ in affirmation_words:
                aspect = prev_token.text + ' ' + aspect
            if next_token and next_token.lower_ in affirmation_words:
                aspect = aspect + ' ' + next_token.text
            # Store the aspect and its category
            matched_aspects.add(aspect)  # Use set.add() to add unique aspects
            matched_categories.add(key.capitalize())  # Store the category in title case
    return ', '.join(matched_aspects), ', '.join(matched_categories)
```

Figure 4.9 Find Attributes Function

This code created a function called find_attributes that took a text input and processed it using spaCy's natural language processing capabilities. Inside this function, the input text was processed using a spaCy model (nlp) to identify aspects related to clothing attributes. The function then iterated through the matches found by a set of pre-defined patterns (attribute matchers) and extracted the aspects and their categories. Additionally, the function checked for affirmation or negation words that might modify the meaning of the aspects. If an affirmation word was found before or after an aspect, it was concatenated with the aspect to capture the modified sentiment. Finally, the function returned the matched aspects and their categories, taking into account any affirmations or negations present in the text.

```
# Convert NaN values to empty strings in the 'Tokenized_Review' column
data['Tokenized_Review'] = data['Tokenized_Review'].fillna('')

# Apply attribute matching to the dataset and add the 'Extracted_Aspects' column
tqdm.pandas(desc="Matching Attributes")
data['Extracted_Aspects'] = data['Tokenized_Review'].astype(str).progress_apply(find_attributes)
```
```
Matching Attributes: 100%|██████████| 32381/32381 [03:40<00:00, 146.63it/s]
```

Figure 4.10 Handle NaN Values

This line replaces any NaN (not a number) values in the 'Tokenized_Review' column with empty strings. This step ensures that the find_attributes function can process all rows without encountering NaN-related errors. Then the line applies the find_attributes function to each row in the 'Tokenized_Review' column of the DataFrame data. It creates a new column called 'Extracted_Aspects' to store the extracted aspects for each review.

```
# Generate word cloud for matched attributes
attributes_text = ' '.join(data['Extracted_Aspects'].explode().dropna())
wordcloud = WordCloud(width=800, height=400).generate(attributes_text)
```

Figure 4.11 Combined All of The Retrieved Aspects

These lines combined all of the retrieved aspects into a single string, preparing the text for the word cloud visualization. The retrieved attributes were then utilized to generate a word cloud object using the WordCloud class from the Wordcloud library.

```
# Display the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Figure 4.12 Generate Word Cloud

Finally, this code displayed the generated word cloud visualization using Matplotlib. The word cloud provided a visual representation of the frequency of different aspects extracted from the clothing reviews, with larger words indicating higher frequency.

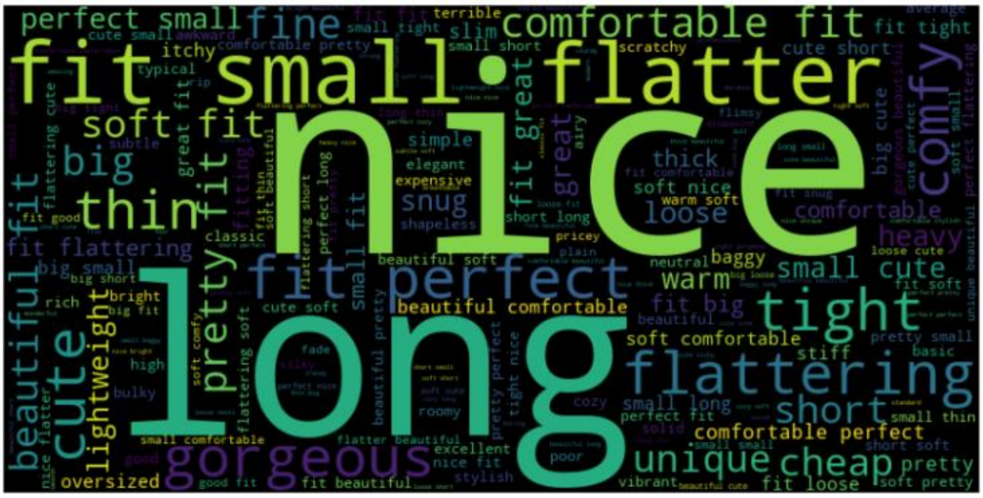


Figure 4.13 World Cloud

During the aspect's extraction process, the number of data points decreased from 32,385 to 28,665 because it was removed due to instances where reviews failed to mention the specified aspects. This reduction underscores the importance of enhancing the predefined list of aspects to improve future extraction accuracy.

4.4 Sentiment Analysis

The sentiment analysis process imported the necessary libraries for data manipulation (Pandas), plotting (Matplotlib), text sentiment analysis (TextBlob), and progress tracking (tqdm). Additionally, it imported the dataset from the previous process, which is the aspect extraction.

```
# Function to calculate sentiment scores for a review
def calculate_sentiment(review):
    analysis = TextBlob(review)
    sentiment_score = analysis.sentiment.polarity
    return sentiment_score
```

```
# Function to calculate sentiment scores for a review based on the "Extracted_Aspects"
def calculate_sentiment(review, aspects):
    sentiment_scores = {'positive': [], 'neutral': [], 'negative': []}
    for aspect in aspects:
        analysis = TextBlob(aspect)
        polarity_score = analysis.sentiment.polarity

        if polarity_score >= 0.1:
            sentiment_scores['positive'].append(polarity_score)
        elif -0.1 < polarity_score < 0.1:
            sentiment_scores['neutral'].append(polarity_score)
        else:
            sentiment_scores['negative'].append(polarity_score)

    return sentiment_scores
```

```python
# Function to calculate sentiment scores for tokenized reviews in list form
def calculate_sentiment_tokenized(review_tokens):
    analysis = TextBlob(' '.join(review_tokens))
    polarity_score = analysis.sentiment.polarity

    if polarity_score >= 0.1:
        return {'positive': [polarity_score], 'neutral': [], 'negative': []}
    elif -0.1 < polarity_score < 0.1:
        return {'positive': [], 'neutral': [polarity_score], 'negative': []}
    else:
        return {'positive': [], 'neutral': [], 'negative': [polarity_score]}
```

Figure 4.14 Functions for Sentiment Analysis

Three sentiment analysis functions were defined by these cells. Using TextBlob's sentiment analysis, the calculate_sentiment function determined the sentiment score (polarity) for a given review. The sentiment scores for a review were determined using the calculate_sentiment_with_aspects function using aspects as an argument. The sentiment ratings were then classified as positive, neutral, or negative depending on predetermined thresholds. For tokenized reviews, the calculate_sentiment_tokenized method computed sentiment scores in list form and sorted them according to polarity scores.

```python
# Function to determine the label based on sentiment scores
def determine_label(sentiment_scores):
    if sentiment_scores['positive']:
        return 'positive'
    elif sentiment_scores['negative']:
        return 'negative'
    else:
        return 'neutral'
```

Figure 4.15 Functions for Creating Sentiment Labels

There were also functions for creating sentiment labels defined in the code. A sentiment label (positive, negative, or neutral) was determined using the determine_label (sentiment_scores) function in conjunction with the sentiment scores. The created label and an already-existing label in the dataset were compared using the compare_labels (row) function.

```python
# Calculate sentiment scores and labels for each row
tqdm.pandas(desc="Calculating Sentiments and Labels")

def calculate_sentiment_and_label(row):
    extracted_aspects = str(row['Extracted_Aspects']).split(', ')

    if not extracted_aspects:
        sentiment_scores = calculate_sentiment_tokenized(row['Tokenized_Review'])
    else:
        sentiment_scores = calculate_sentiment(row['Review'], extracted_aspects)

    label = determine_label(sentiment_scores)

    return sentiment_scores, label
```

Figure 4.16 Calculate Sentiment and Label Function

Using progress_apply from tqdm, the code applied the calculate_sentiment_and_label function to each row in the dataset. Based on the extracted aspects or tokenized review, this function determined the sentiment ratings and labels for every review. The outcomes were entered into two new columns which is "Label" and "Sentiment Scores."

```
# Apply sentiment score and label calculation to the dataset
tqdm.pandas(desc="Calculating Sentiments, Labels, and Comparing")
dataset[['Sentiment_Scores', 'Label']] = dataset.progress_apply(calculate_sentiment_and_label, axis=1, result_type='expand')
```

```
Calculating Sentiments, Labels, and Comparing: 100%|██████████| 27816/27816 [00:09<00:00, 2890.35it/s]
```

```
# Apply label comparison
dataset['Label_Match'] = dataset.progress_apply(compare_labels, axis=1)
```

```
Calculating Sentiments, Labels, and Comparing: 100%|██████████| 27816/27816 [00:00<00:00, 121740.09it/s]
```

Figure 4.17 Compare Label Process

To apply the compare_labels function to every row in the dataset, another progress_apply was utilized. This function, probably based on additional criteria or annotations, compared the created sentiment label with an existing label in the dataset.

```
# Count the True and False occurrences
label_match_counts = dataset['Label_Match'].value_counts()
```

```
# Plotting the bar chart
plt.figure(figsize=(8, 6))
label_match_counts.plot(kind='bar', color=['blue', 'orange'])
plt.title('Comparison of Generated Labels and Existing Labels')
plt.xlabel('Label Match')
plt.ylabel('Frequency')
plt.xticks([0, 1], ['True', 'False'], rotation=0)
plt.show()
```

Figure 4.18 Generate Bar Chart

Lastly, the code used Matplotlib (plt) to build a bar chart by counting the instances of True and False label matches. The created sentiment labels and the preexisting labels were compared visually in a bar chart, which shed light on the sentiment analysis technique' accuracy.

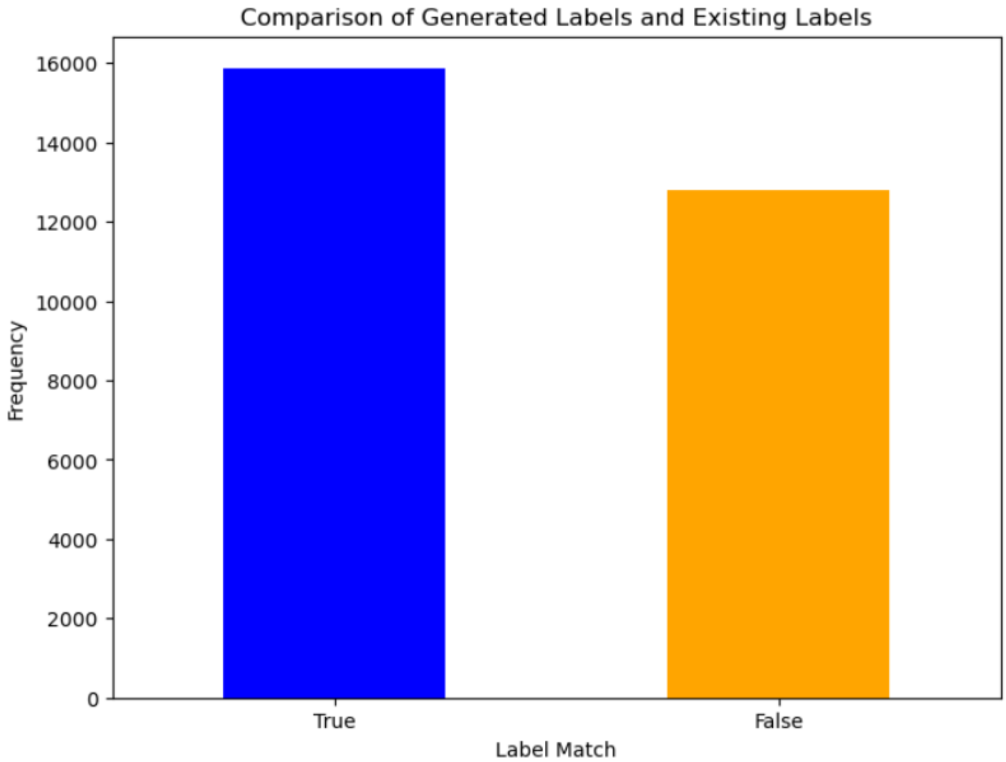


Figure 4.19 Graphs of Comparison between Labels and Existing Labels

The graph showed that there were over 15678 occurrences when the created sentiment label matched with the existing label (True) and 12788 occurrences where it did not (False). The reviews that had mismatched labels were eliminated in order to guarantee that the best reviews were used for modeling. The labels present in the dataset were obtained from the ratings of the clothing. Specifically, ratings of four or higher were classified as positive, ratings of three or lower as neutral, and ratings of two or lower as negative. The sentiment labels produced by TextBlob's sentiment analysis were then contrasted with these pre-defined labels. According to one of the problem statements (Almansour et al., 2022), ratings may not accurately reflect the sentiment expressed in the associated review texts. The difference between the sentiment labels that TextBlob produced from the reviews and the labels that came from the ratings could have been one way that this bias showed itself. The intention was to use more trustworthy data for modelling by eliminating the reviews with mismatched classifications.

4.5 Model Training

4.5.1 Naïve Bayes

Table 4.4 Classification Reports for Naïve Bayes

	Precision	**Recall**	**F1-Score**	**Support**
Negative	1.00	0.07	0.13	194
Neutral	0.00	0.00	0.00	135
Positive	0.90	1.00	0.95	2847
Accuracy			0.90	3176
Macro Average	0.63	0.36	0.36	3176
Weighted Average	0.87	0.90	0.86	3176
Accuracy (%)	90.05%			

The result of sentiment analysis using a Naive Bayes classifier on apparel reviews was shown in the figure. The categorization report included information on three sentiment categories which is negative, neutral, and positive, as well as precision, recall, f1-score, and support. The classifier demonstrated exceptional precision for negative sentiments, meaning that all reviews that were classified as negative were, in fact, negative. However, the poor recall rate implies that a large number of negative reviews were missed. Zeroes across precision and recall imply that neutral evaluations were neither correctly identified nor classified. However, positive ratings performed significantly better, with flawless recall and excellent precision, demonstrating the model's potent capacity to detect pleasant emotions. The model achieved an accuracy rate of 90.05% overall. However, this figure could be skewed by the dataset's imbalance, which was primarily composed of positive reviews.

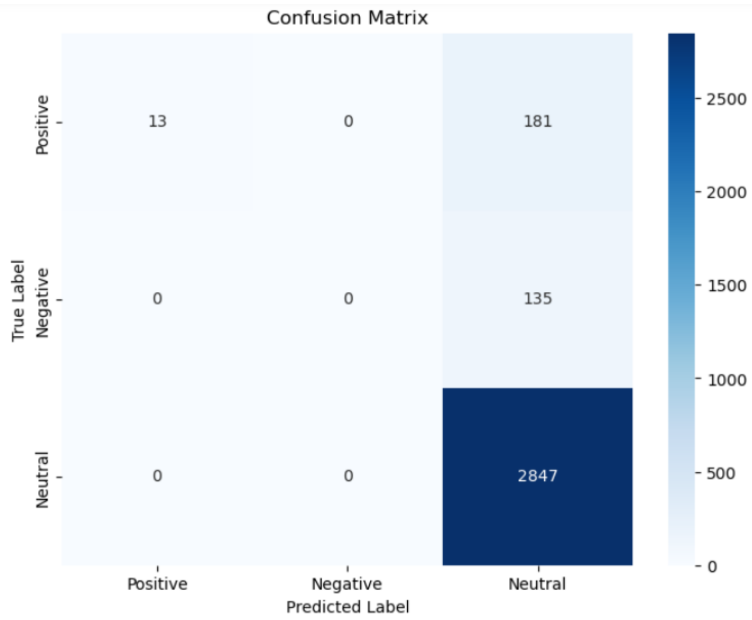


Figure 4.20 Confusion Matrix for Naïve Bayes

Additional insight into the model's performance can be gained from the visual representation of the confusion matrix, which shows the tendency to favour positive sentiment classification. The model does a good job at identifying a substantial portion of positive reviews, but it has trouble picking out neutral and negative opinions. It is noteworthy that it significantly overpredicts good ratings and misclassifies the majority of negative reviews as positive, failing to generate any accurate predictions for neutral evaluations.

4.5.2 Recurrent Neural Network (RNN)

Table 4.5 Classification Report for RNN

	Precision	Recall	F1-Score	Support
Positive	1.00	1.00	1.00	2847
Negative	0.95	0.96	0.96	194
Neutral	0.93	0.94	0.93	135
Accuracy			0.99	3176
Macro Average	0.96	0.97	0.96	3176
Weighted Average	0.99	0.99	0.99	3176
Accuracy (%)	99.15%			

The sentiment analysis Recurrent Neural Network (RNN) model performs exceptionally well when its batch size is 32 and it is trained over 10 epochs. The model has classified positive, negative, and neutral attitudes in the dataset with almost flawless precision, recall, and f1-scores. While the recall measure shows that the model missed very few real instances of each sentiment, the accuracy metric suggests that practically all predictions in each sentiment category were true. The remarkable overall accuracy of 99.15% indicates how well the model works for sentiment classification.

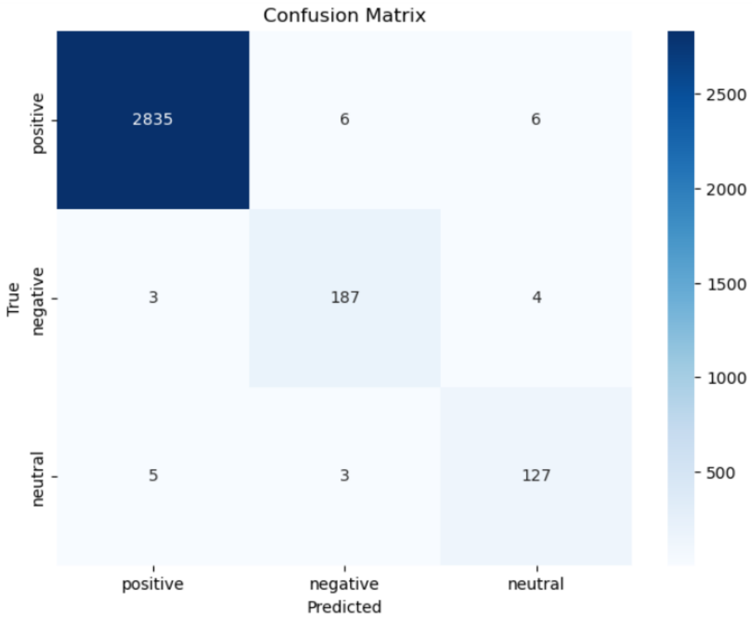


Figure 4.21 Confusion Matrix for RNN

The confusion matrix indicates minimal misclassification, highlighting the model's ability to distinguish between sentiment classes despite the challenges of text classification, such as understanding language nuances. The model demonstrates strong generalization across various sentiment categories, suggesting that the chosen combination of batch size and epochs was effective in enabling the model to learn without overfitting.

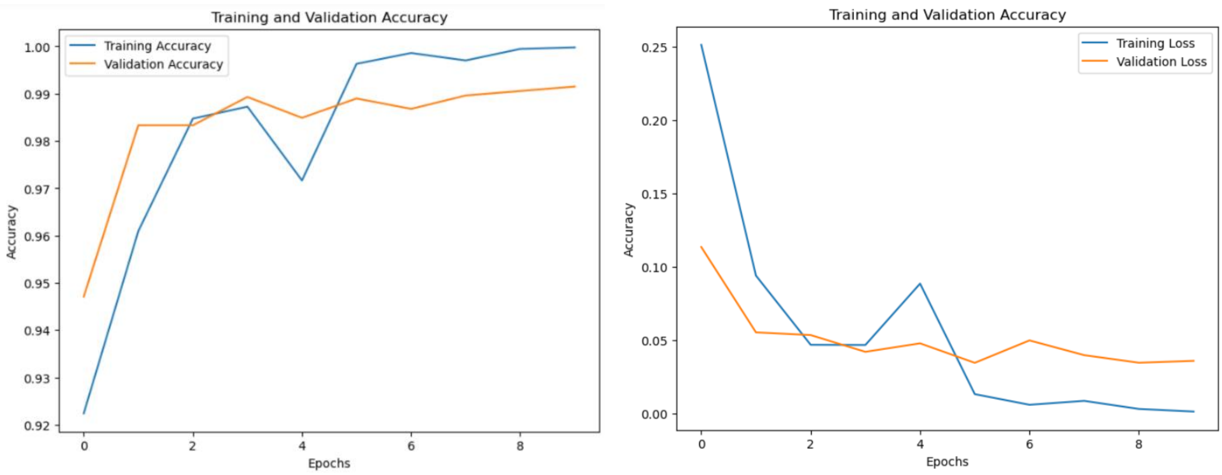


Figure 4.22 Graphs of Training Accuracy and Training Loss for RNN

The accuracy graph makes it clear that training accuracy starts out high and steadily increases over time, eventually stabilizing at 99%. As is typical during training, the validation accuracy starts out at a high level but exhibits some fluctuations across epochs due to differences in the validation data. It doesn't overfit to the training set, either, since it stays quite near to the training accuracy, showing that the model is doing an excellent job of generalizing.

The training loss decreased rapidly in the first epoch and then progressively decreases further, levelling off around the fourth epoch in the loss graph, which normally displays a decreasing trend as the model learns. On the other hand, the validation loss shows some fluctuation but reduces quickly, with a noteworthy peak around the sixth epoch before declining once more.

The model performs well overall with high accuracy and minimal loss, suggesting that the batch size and epoch count selected for this training were appropriate. The training and validation measures' close consistency indicates that the model is strong and ought to function reliably on untested data.

4.5.3 Long Short-term Memory (LSTM)

Table 4.6 Classification Report for LSTM

	Precision	Recall	F1-Score	Support
Positive	0.99	1.00	0.99	2847
Negative	0.95	0.93	0.94	194
Neutral	0.92	0.88	0.90	135
Accuracy			0.99	3176
Macro Average	0.95	0.94	0.95	3176
Weighted Average	0.99	0.99	0.99	3176
Accuracy (%)	98.77%			

The sentiment analysis performance of the LSTM model, trained across 10 epochs with a batch size of 32, shows 98.77% accuracy in classification, indicating extremely effective performance. Across all sentiment classes, the model has shown exceptional precision and recall. Recall for positive reviews is nearly perfect, and precision is almost perfect, suggesting that the model is quite capable in detecting positive sentiments. While scores are slightly lower for negative and neutral groups, they nevertheless show a high ability to appropriately classify sentiments.

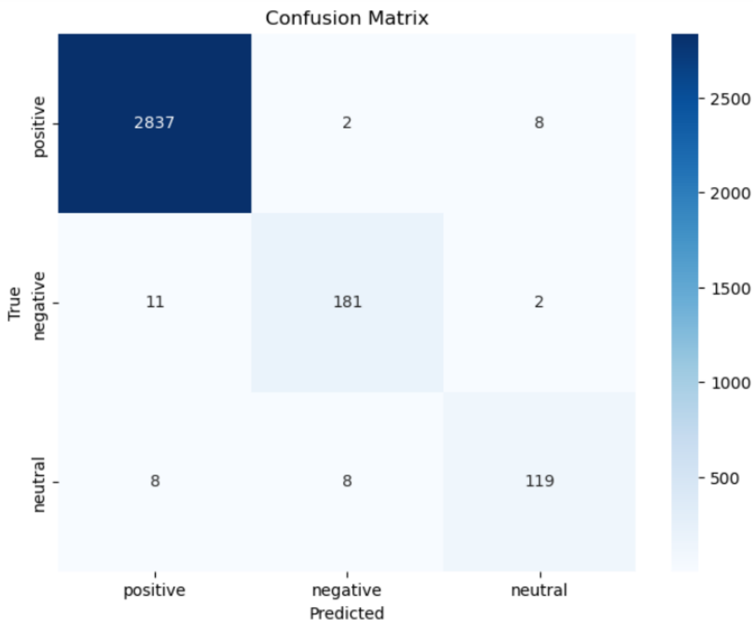


Figure 4.23 Confusion Matrix for LSTM

The model tended to misclassify positive sentiments as neutral, and the confusion matrix indicated that this type of error was more common than misclassifications between other sentiment classes. This pattern suggested a challenge in distinguishing between subtle differences in the expressions of neutral and positive sentiments in text.

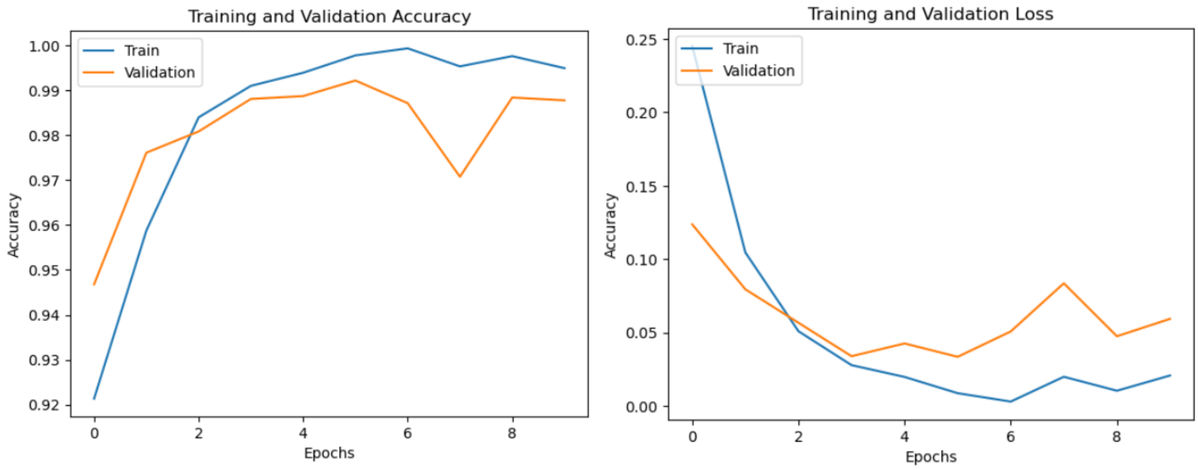


Figure 4.24 Graphs of Training Accuracy and Training Loss for LSTM

According to the accuracy graph, training accuracy increased significantly in the first few epochs before stabilizing, although with occasional variations, around the 99% mark. Despite having a slightly lower initial value, the validation accuracy quickly increased and closely tracked the training accuracy, suggesting strong generalization without appreciable overfitting.

The training loss initially decreased rapidly before stabilizing with more epochs, suggesting the model was converging. The validation loss showed some fluctuations, which is normal when the model encounters new validation data. However, occasional spikes in validation loss could indicate overfitting, where the model learns from noise in the training data rather than the actual pattern, or challenges with specific validation data batches.

The LSTM model appears to have trained efficiently and generalized well to the validation data overall, based on its learning curves. For this task, the selected training parameters of 10 epochs and a batch size of 32 appeared appropriate. The small variations in validation accuracy and loss may be resolved by optimizing the hyperparameters of the model or changing the learning rate to improve stability over time. The model's high accuracy and converging loss suggest a successful training outcome, despite these minor issues.

Table 4.7 Summary of Model’s Classification Report

Matrix	Naïve Bayes	RNN	LSTM
Precision	0.63	0.96	0.95
Recall	0.36	0.97	0.94
F1-Score	0.36	0.96	0.95
Accuracy (%)	90.05%	99.15%	98.77%

According to the total accuracy rates reported for the RNN, LSTM, and Naive Bayes models, the RNN model had the best accuracy, at 99.15%. This shows that the RNN did the greatest job of accurately categorizing sentiment out of the three models that were evaluated. Since the RNN's shown accuracy is higher than that of the LSTM and Naive Bayes models, it is the most promising option for additional testing to potentially enhance its sentiment analysis capabilities.

4.6 Experiments

4.6.1 Experiment 1

Table 4.8 Classification Report for Experiment 1

	Precision	Recall	F1-Score	Support
Positive	1.00	1.00	1.00	2847
Negative	0.98	0.88	0.93	194
Neutral	0.79	0.93	0.86	135
Accuracy			0.98	3176
Macro Average	0.92	0.94	0.93	3176
Weighted Average	0.99	0.99	0.99	3176
Accuracy (%)	98.68%			

Experiment one employed an RNN model with the Adamax optimizer, trained over ten epochs with a batch size of 32. The classification report showed excellent recall and precision metrics in every category, with the positive class displaying perfect scores. Remarkably, the model achieved a 98.68% total accuracy, demonstrating its ability to predict sentiments.

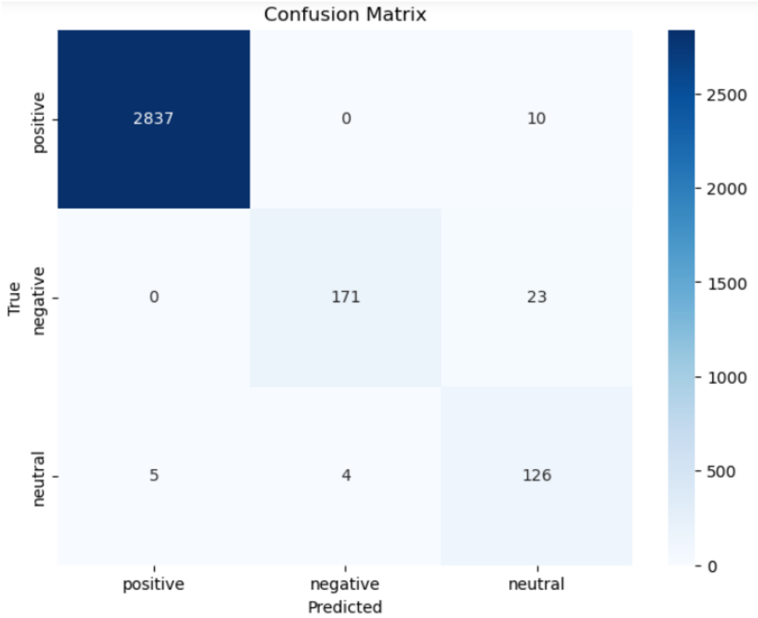


Figure 4.25 Confusion Matrix for Experiment 1

The model's performance is revealed by the confusion matrix. It illustrates how well the model can identify positive sentiment, with very few false positives and no false negatives observed. Although the recall of the model is slightly reduced, it still shows good precision in detecting negative attitudes, suggesting that some negative sentiments were misclassified. However, compared to the other categories, the model's precision and recall are poorer, making it difficult to reliably detect neutral attitudes. This implies that the model might struggle to discern between positive and negative sentiments and neutral sentiments.

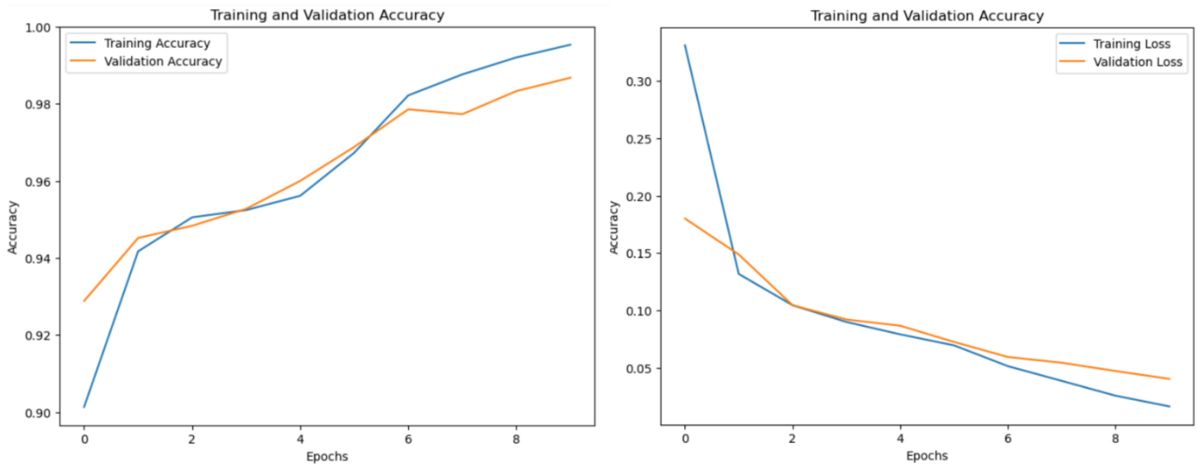


Figure 4.26 Graphs of Training Accuracy and Training Loss for Experiment 1

The accuracy graph shows that early in the training process, the model's training accuracy significantly improved, indicating quick learning from the training data. The training accuracy then increased progressively until hitting a plateau, suggesting that the model had almost reached its peak performance on the training dataset and that there had been little improvement in accuracy with more epochs. On the other hand, the validation accuracy first increased more slowly but thereafter closely followed the training accuracy. The model may have successfully generalized to the validation data by understanding the underlying patterns rather than simply memorizing the training set, based on this close alignment.

A close look at the loss graph reveals an early significant decrease in training loss, suggesting quick error correction in the early learning stage. This reduction slowed down as the epochs went on, suggesting that the model was getting close to its learning bounds with the current architecture and hyperparameters. In a similar vein, the validity loss dropped quickly at first before levelling off. However, there was considerable variation in the validation loss, which could indicate that the model's performance on the validation set was unstable. These fluctuations could be the result of early overfitting, in which the model learned certain features of the training data that did not transfer well to the validation data, or the model encountered validation data that was notably different from the training set.

The model successfully identified the underlying patterns in the data, as shown by the learning curves for accuracy and loss. The training parameters of ten epochs and a batch size of 32 were appropriate for the task. The validation loss's volatility, however, points to the possibility of more optimization. In general, the model exhibits good accuracy and consistent loss patterns, suggesting that more fine-tuning may be necessary to achieve maximum performance.

4.6.2 Experiment 2

Table 4.9 Classification Report for Experiment 2

	Precision	**Recall**	**F1-Score**	**Support**
Positive	1.00	0.99	1.00	2847
Negative	0.99	0.90	0.94	194
Neutral	0.80	0.96	0.87	135
Accuracy			0.99	3176
Macro Average	0.93	0.95	0.94	3176
Weighted Average	0.99	0.99	0.99	3176
Accuracy (%)	98.71%			

Experiment two employed an RNN model with the Adamax optimizer, trained over 20 epochs with a batch size of 64. With precision and F1-scores for positive ratings exceeding 1.00 and high scores for negative and neutral emotions as well, the model showed remarkable recall and precision across all sentiment classes positive, negative, and neutral. This suggests an excellent capacity to recognize positive sentiments and distinguish negative sentiments from others. The model performed well even if its accuracy with neutral reviews was slightly lower. This suggests that further data or adjustment could help the algorithm better understand more complex, neutral language. The model's total accuracy was a very high 98.71%, demonstrating its ability to effectively classify attitudes.

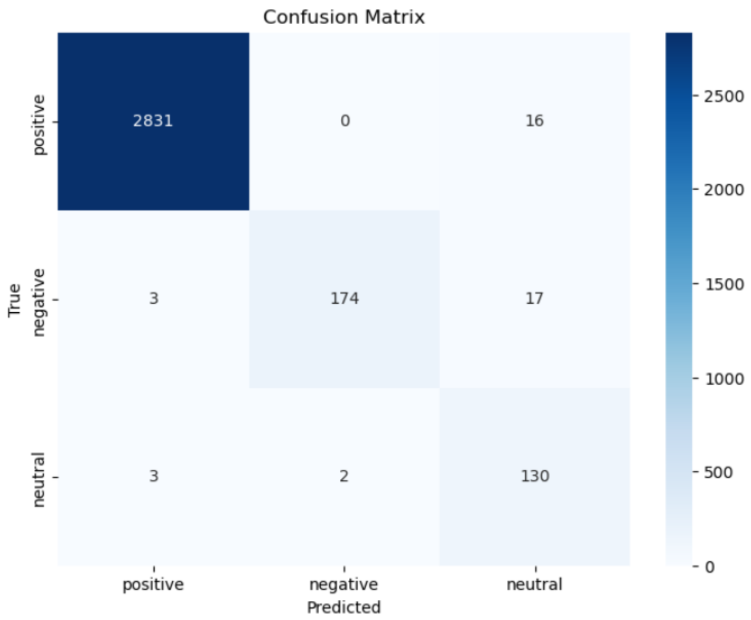


Figure 4.27 Confusion Matrix Experiment 2

The confusion matrix showed that the model had a significant bias in favor of accurate classifications, particularly for positive evaluations, and that there was very little confusion between the neutral and other classes. This high accuracy, especially in the case of few misclassifications, indicated that the model was well-suited and efficient for the task at hand, while it could have done a better job of differentiating between neutral feelings. Such performance was promising for practical applications, indicating the model's robustness and reliability in sentiment analysis tasks.

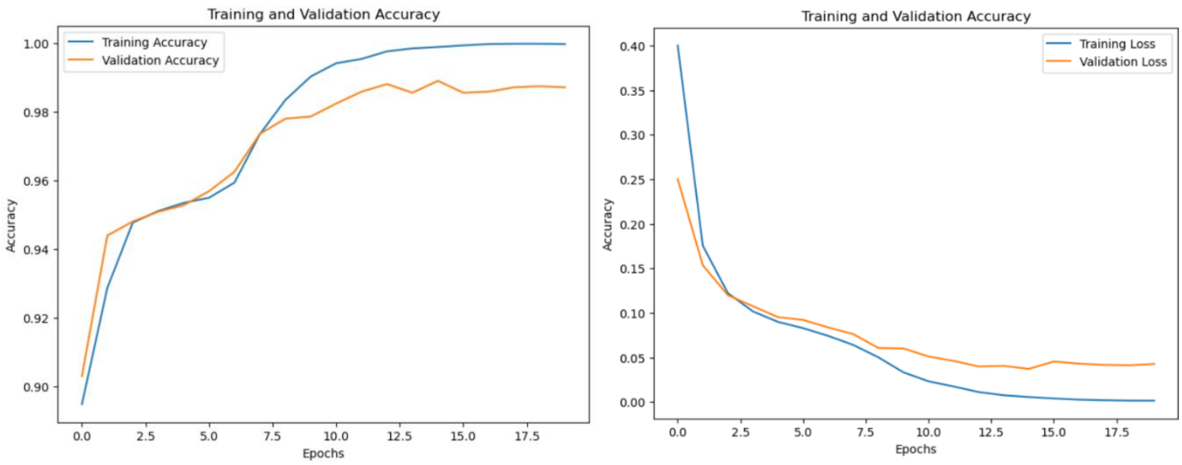


Figure 4.28 Graphs of Training Accuracy and Training Loss for Experiment 2

The training accuracy grew significantly in the first few epochs, as can be seen from the accuracy graph, indicating that the model picked up the training data quickly. The rate of training accuracy improvement decreased as the epochs carried on, suggesting that the model was reaching the limits of what could be achieved with the available data and network architecture. As the epochs went on, the validation accuracy nearly followed the training accuracy curve, having initially lagged behind the training accuracy but quickly catching up. This trend demonstrated that the model did not overfit, which is often indicated by an increasing gap between the training and validation accuracy, and that the model adapted well to new data.

Based on the loss graph, it shows the model had swiftly and effectively lowered its error rate due to a sharp decline in training loss from the beginning. It was to be expected that as the model optimized and the number of epochs increased, the training loss would decrease more gradually, and the weights would be adjusted less. Like the training loss, the validation loss decreased quickly at initially before levelling out and leaving an insignificant gap between the two curves. This small gap was another indication of good generalization, as a significant increase in validation loss compared to training loss would suggest overfitting.

Overall, the learning curves indicated a successful training procedure, and the model demonstrated strong task learning with excellent accuracy and continuously declining loss. Perfect for a machine learning model, the high performance was sustained across the epochs without any indications of overfitting.

4.6.3 Experiment 3

Table 4.10 Classification Report for Experiment 3

	Precision	Recall	F1-Score	Support
Positive	1.00	0.99	1.00	2847
Negative	0.97	0.92	0.95	194
Neutral	0.84	0.94	0.89	135
Accuracy			0.99	3176
Macro Average	0.93	0.95	0.94	3176
Weighted Average	0.99	0.99	0.99	3176
Accuracy (%)	98.77%			

Experiment three employed an RNN model with the Adamax optimizer, trained over 30 epochs with a batch size of 128. This experiment shows an astounding 98.77% overall accuracy. The model achieved perfect precision and an F1-score of 1.00, demonstrating outstanding ability to recognize positive reviews. The model also demonstrated good recall and precision for negative evaluations, with an F1-score of 0.95. While the model was still able to capture most neutral sentiments, it proved to be slightly more difficult to identify neutral evaluations, as seen by a lower precision of 0.84 and a high recall of 0.94.

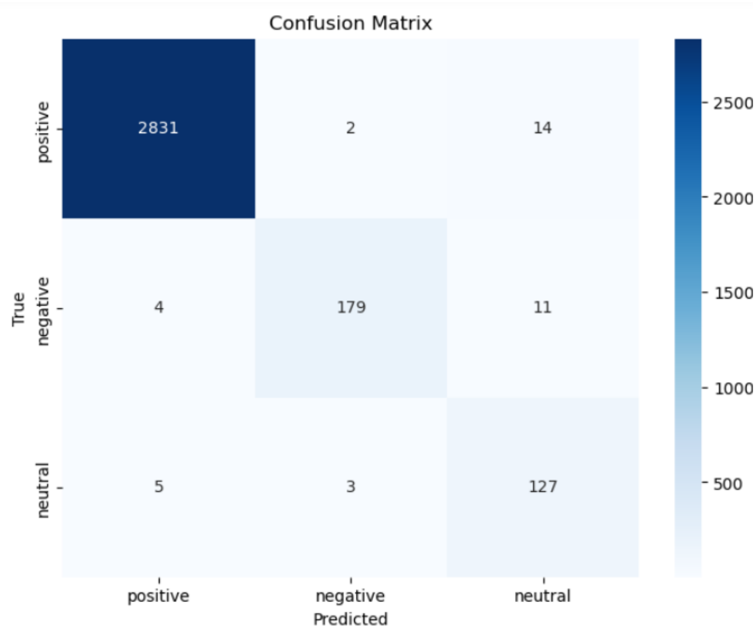


Figure 4.29 Confusion Matrix Experiment 3

The majority of misclassifications occurred between the positive and neutral categories, which is a common challenge in sentiment analysis due to the delicate nature of neutral language. This further demonstrated the confusion matrix's high predictive ability. In spite of this, the model demonstrated a well-tuned system that might be highly useful for real-world sentiment analysis applications, as seen by its overall high scores in precision, recall, and F1-score across all categories.

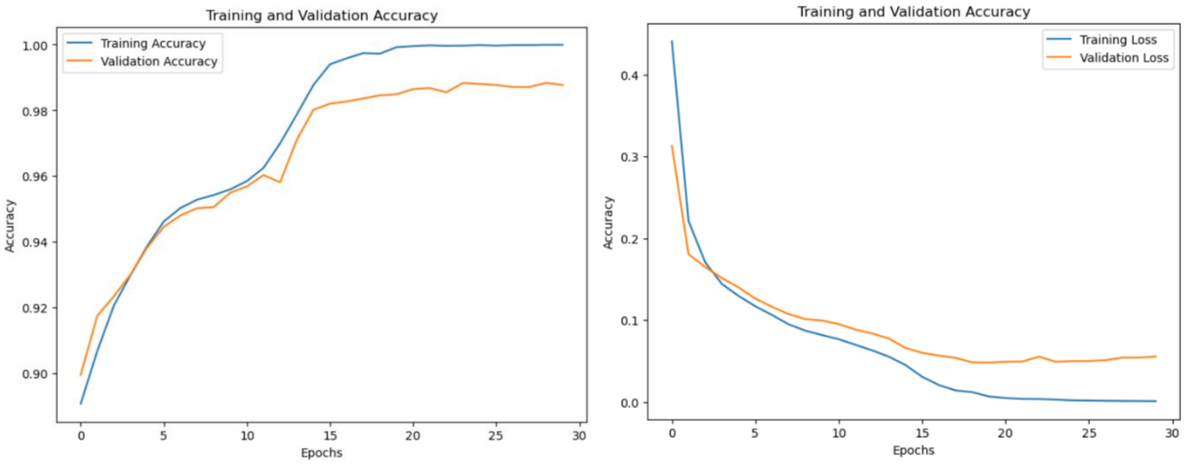


Figure 4.30 Graphs of Training Accuracy and Training Loss for Experiment 3

The training accuracy increased rapidly in the early epochs of the accuracy graph, suggesting that the model picked up the training dataset fast. Around the tenth epoch, this sharp rise started to level down, indicating that the model was getting close to operating at its best with the existing architecture and training data. The validation accuracy increased alongside the training accuracy but plateaued sooner, which is typical as the model began to reach the limits of what it could generalize from the training data.

At first in the training process, the loss graph demonstrated a significant decrease in training loss, indicating significant learning and improvements in the model's predictions. This loss decrease declined as the epochs went on, as is typical when the model adjusted its parameters. The model was not overfitting, as evidenced by the validation loss declining along with the training loss and both curves converging. The learning from the training data was successfully applied to the validation data by the model.

The model achieved high accuracy and low loss, indicating that it could successfully learn and generalize. These graphs overall reflected a well-conducted training process. As the training process came to a close, there was no apparent divergence between the training and validation lines, which was a positive indication of the model's robustness.

4.6.4 Experiment 4

Table 4.11 Classification Report for Experiment 4

	Precision	Recall	F1-Score	Support
Positive	1.00	0.99	1.00	2847
Negative	0.97	0.98	0.98	194
Neutral	0.94	0.98	0.96	135
Accuracy			0.99	3176
Macro Average	0.97	0.98	0.98	3176
Weighted Average	0.99	0.99	0.99	3176
Accuracy (%)	99.43%			

Experiment four employed an RNN model with the Adam optimizer, trained over ten epochs with a batch size of 32. Across all sentiment categories which is positive, negative, and neutral shows that the model showed exceptional ability to predict, obtaining nearly perfect precision, recall, and F1-scores. It demonstrated exceptional accuracy in identifying neutral and negative attitudes with very few errors, and it accurately recognized positive thoughts. An astounding 99.43% of the model was accurate overall.

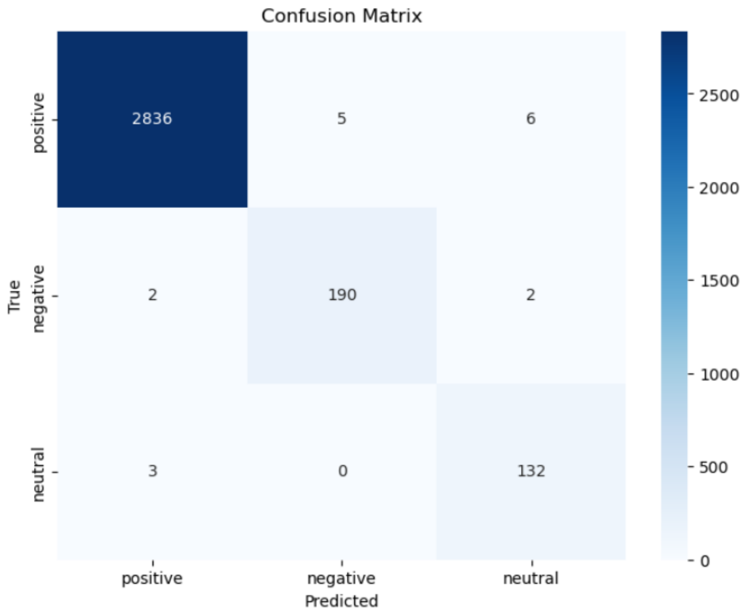


Figure 4.31 Confusion Matrix Experiment 4

The confusion matrix for the sentiment analysis model demonstrates strong performance, particularly in identifying positive sentiment in clothing reviews. The model is highly accurate with positives, while still performing reasonably well on negative and neutral categories, with only minor confusion between them. The relatively clear distinction along the diagonal of the matrix suggests that the model has learned to distinguish between the sentiments effectively, with most reviews being classified correctly according to their true sentiment.

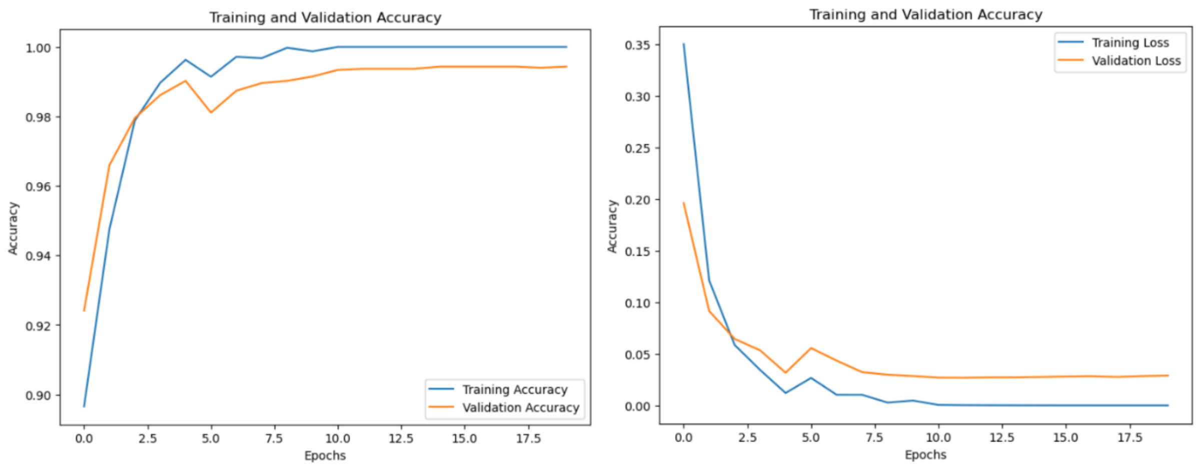


Figure 4.32 Graphs of Training Accuracy and Training Loss for Experiment 4

The training and validation accuracy and validation loss for an RNN model across around 20 epochs were displayed in the graphs. The training and validation accuracy both started high and rapidly converged, with the training accuracy eventually settling slightly above the validation accuracy, according to the accuracy graph. The model appeared to have achieved good generalization without overfitting, as evidenced by the tight convergence of training and validation accuracy, with no apparent divergence.

During the first few epochs, the training and validation losses on the loss graph decreased sharply before decreasing, suggesting that the model had gained experience and become more accurate in its predictions. Since the model learns directly from the training data, it was expected that the training loss would always be lower than the validation loss. Nonetheless, the validation loss stabilized and remained near the training loss, providing additional evidence that the model had not overfit and had successfully generalized to previously unseen data.

In general, the model demonstrated a robust learning performance, exhibiting high accuracy and minimal loss, signifying efficient learning during the training phase. The absence of overfitting indicated that the model architecture and selected hyperparameters were appropriate for the given sentiment analysis task.

4.6.5 Experiment 5

Table 4.12 Classification Report for Experiment 5

	Precision	**Recall**	**F1-Score**	**Support**
Positive	1.00	1.00	1.00	2847
Negative	0.96	0.95	0.96	194
Neutral	0.91	0.93	0.92	135
Accuracy			0.99	3176
Macro Average	0.96	0.96	0.96	3176
Weighted Average	0.99	0.99	0.99	3176
Accuracy (%)	99.09%			

Experiment five employed an RNN model with the Adam optimizer, trained over 30 epochs with a batch size of 128. The model performed remarkably well, with an overall accuracy of 99.09%. With a perfect score of 1.00 for recall, precision, and F1-score, it demonstrated perfect classification with no false positives. With an F1-score of 0.96, scores for negative sentiments were likewise high, demonstrating a good capacity to precisely identify negative sentiments. With precision at 0.91 and recall at 0.93, neutral evaluations had significantly lower scores. Despite this, the reviews' F1-score of 0.92 was still excellent and demonstrated the inherent difficulty of identifying more ambiguous neutral sentiments.

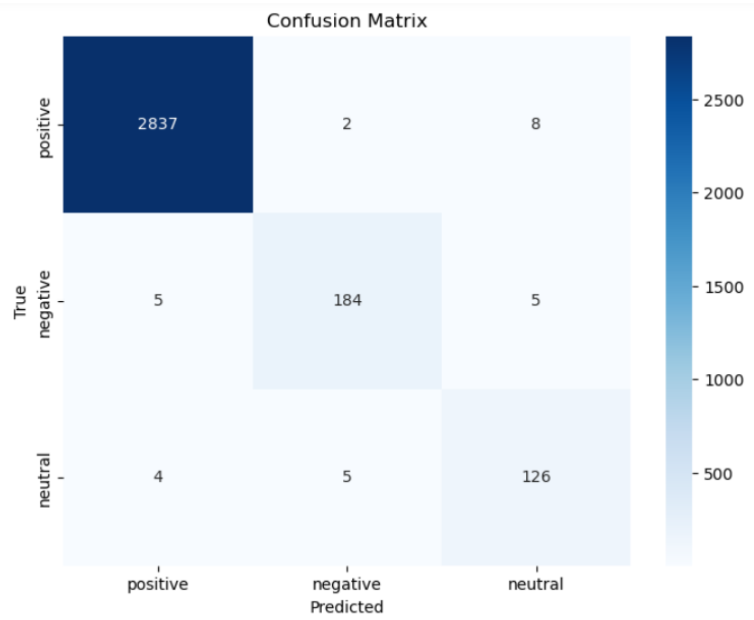


Figure 4.33 Confusion Matrix Experiment 5

The confusion matrix of this experiment indicates the sentiment analysis model is still performing effectively, especially in recognizing positive sentiments. There's a high rate of correctly identified sentiments, with the largest portion correctly classified as positive. The model also shows a good ability to distinguish between the different sentiments, although it has some instances where it incorrectly classified negative and neutral sentiments as positive. It also indicates a few instances of misclassification between negative and neutral sentiments, but these are relatively low. Overall, the model's capability to accurately identify the correct sentiment remains robust, with most of the classifications aligning well with the true sentiments.

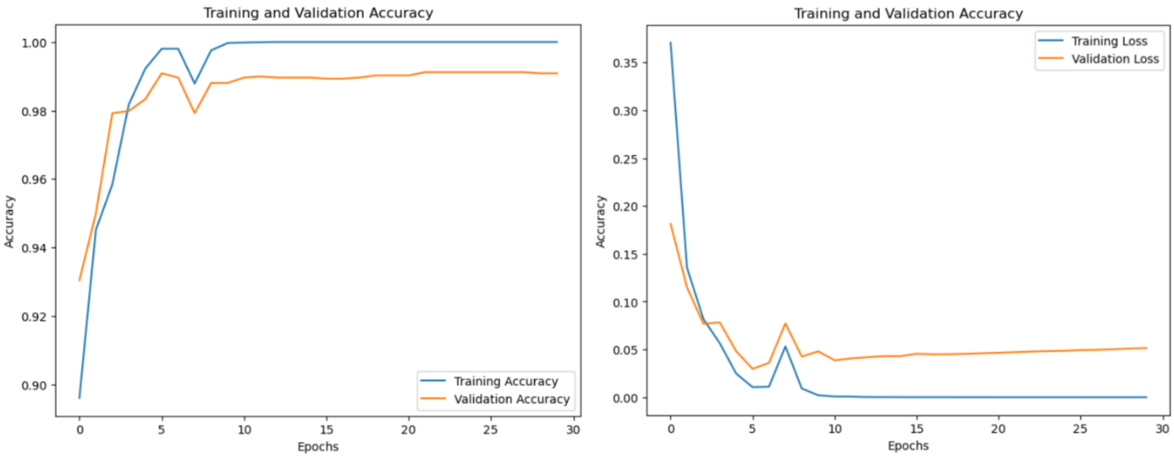


Figure 4.34 Graphs of Training Accuracy and Training Loss for Experiment 5

As the model started to learn from the training data, the accuracy graph on the left demonstrated a sharp increase in training accuracy throughout the early epochs. After roughly five epochs, the validation accuracy similarly increased quickly but showed some volatility, varying slightly above and below the training accuracy. This variation may have represented how the model responded to various characteristics in the validation dataset compared to the training set. Nonetheless, the fact that the validation accuracy stayed high and near the training accuracy indicated that the model did not overfit and had good generalization.

As the number of epochs increased, the training loss on the right-hand loss graph showed a steep early fall that levelled out. This was predicted and normal as the model got closer to an ideal weight set. Although it showed a rise around the fifth epoch, the validation loss declined in step with the training loss. This increase might have pointed to a set of validation data that was very difficult or maybe a learning rate that needed to be adjusted. Nonetheless, the validation loss promptly declined in tandem with the training loss, indicating that the model recovered effectively from the cause of the spike.

Overall, the graphs' performance of the model seemed to have been robust, with high accuracy and minimal loss, suggesting that it was capable of learning and generalizing. The fact that the training and validation metrics for the latter epochs showed a low gap suggested that the model was well-tuned and might have functioned well on fresh, untested data.

4.6.6 Summary of Experiments

Table 4.13 Summary of Experiments

Matrix	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6
Optimizer	Adamax	Adamax	Adamax	Adam	Adam	Adam
Epoch	10	20	30	10	20	30
Batch	32	64	128	32	64	128
Precision	0.92	0.93	0.93	0.96	0.97	0.96
Recall	0.94	0.95	0.95	0.97	0.98	0.96
F1-Score	0.93	0.94	0.94	0.96	0.98	0.96
Accuracy (%)	98.68%	98.71%	98.77%	99.15%	99.43%	99.09%

Table 4.13 displayed the results of six experiments that evaluated the performance of trained models under different parameters. Experiment four was included because it had already been conducted in the 4.5 model training section. Experiments one to three utilized the Adamax optimizer, while experiments four to six employed the Adam optimizer. The experiments varied in the number of epochs (10, 20, 30) and batch sizes (32, 64, 128) to observe their impact on performance metrics like precision, recall, F1-score, and accuracy. Across the experiments, precision and recall improved with an increase in the number of epochs and batch size. Experiment five demonstrated the best results, achieving the highest precision, recall, and F1-score (using the Adam optimizer with 20 epochs and a batch size of 64), with an accuracy of 99.43%. Based on these metrics, experiment five was the best model and was selected to run the prototype testing.

4.6.7 Prototype Testing

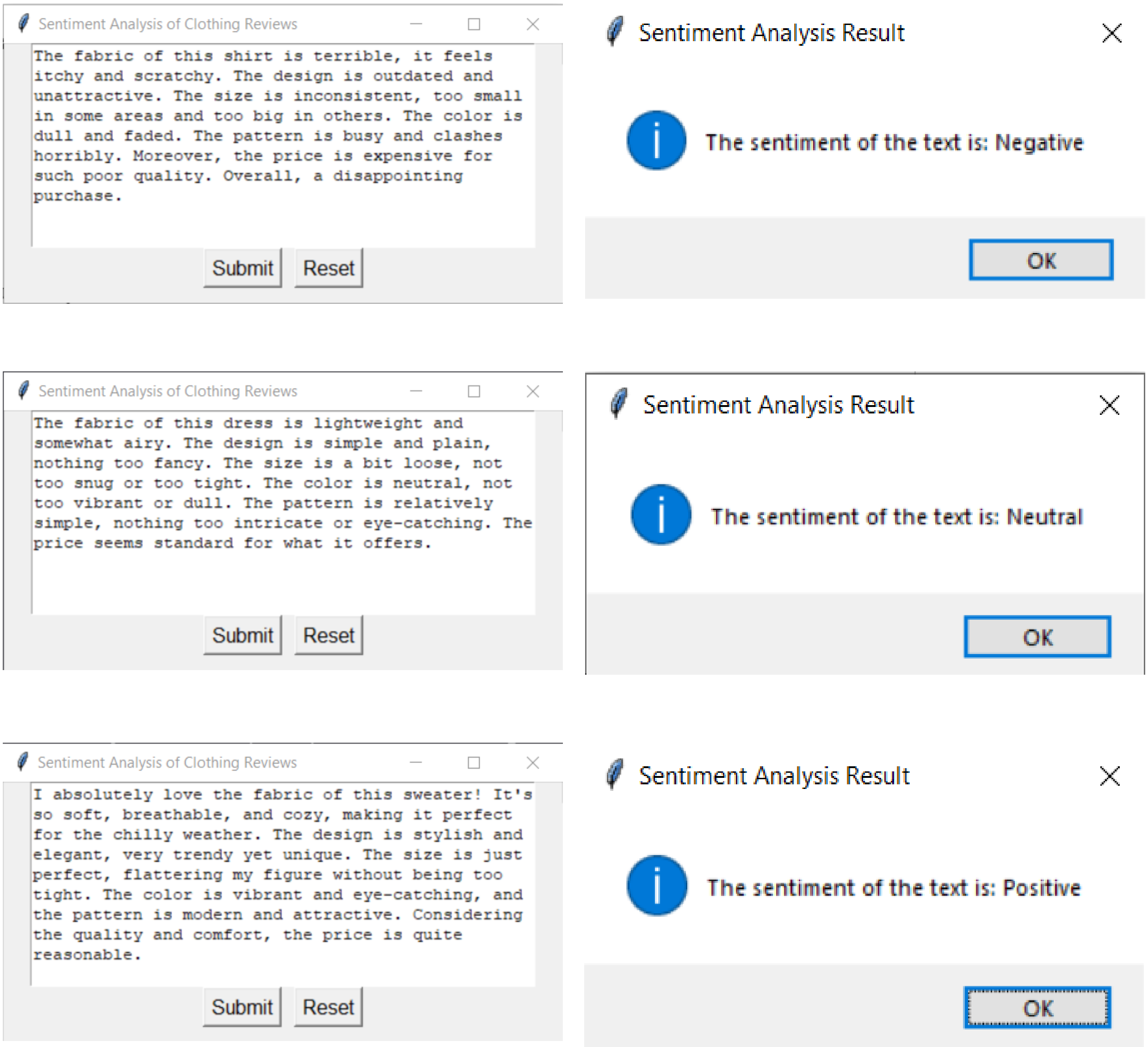


Figure 4.35 Simple User Interface for Sentiment Analysis

The prototype shown in the Figure 4.35 is a tool for sentiment analysis designed to categorize apparel reviews as neutral, negative, or positive. This prototype used the RNN model that was trained over 20 epochs with a batch size of 64 using the Adam optimizer. The examples demonstrate how the model can recognize and categorize sentiment in text. This user-friendly interface allows even non-technical users to easily test and interact with the model. It could be a valuable tool for customer service representatives, product managers, or marketers who need to quickly understand customer sentiment.

The model has been well-trained and shows promise as a reliable tool for analysing sentiment in text, as indicated by its excellent accuracy shown in previous graphs and reports. However, while the RNN model performs well in the provided examples, real-world applications may present more complex challenges, such as sarcasm, idioms, or conflicting emotions in a single review. Additionally, the prototype needs detailed mentions of aspects in reviews to accurately predict sentiment. Reviews that are too short or lack detail in aspects might lead to inaccurate predictions. Further improvements or advanced natural language processing methods may be needed to address these challenges.

4.7 Dashboard

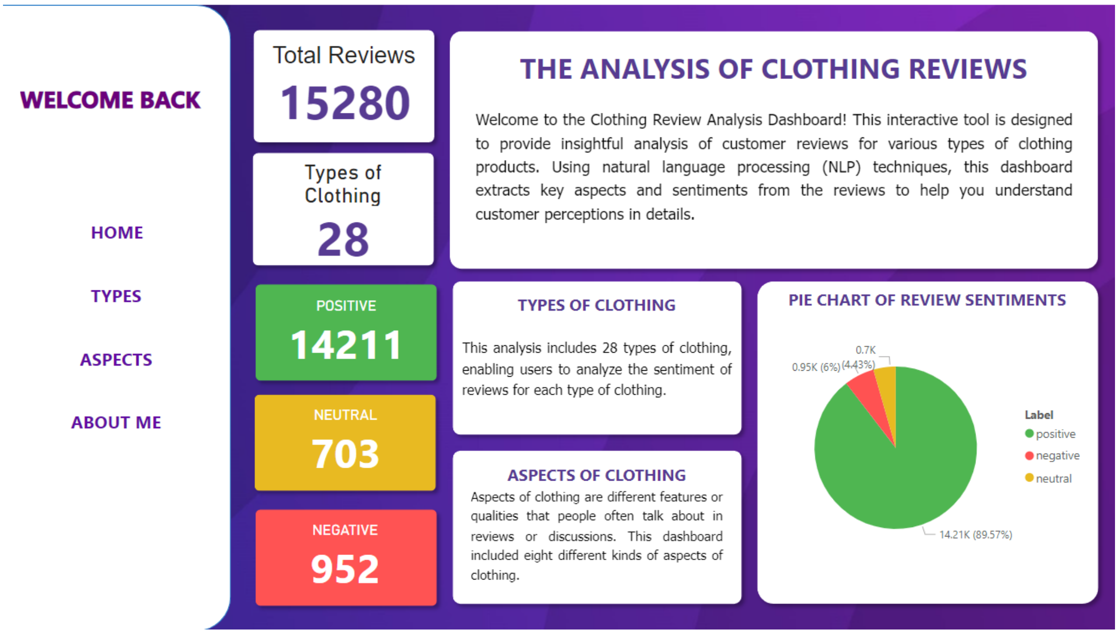


Figure 4.36 Main Page of Dashboard for Reviews Analysis

The page served as an introduction screen, introducing users with the analytical abilities of the tool and setting the basis for an in-depth analysis of customer sentiments and patterns in the clothing industry. The page captured the essence of the dashboard's capabilities. The page encouraged users to explore the types and aspects of clothing analyses in more detail on the following pages.

Figure 4.37 Types of Clothing Page

The second page of the Clothing Review Analysis Dashboard provided a focused view on the types of clothing. It retained the summary of total reviews, categorized sentiments, and the 28 types of clothing from the main page. A pie chart color-coded the sentiments of the reviews, further supporting the visual representation with a column chart that displayed the number of reviews per clothing type. This page allowed user to quickly figure out what types of clothing were being discussed about the most and the least. A stacked bar chart was also used to display the percentage of positive, neutral, and negative evaluations for each category of clothing, including dresses, knits, blouses, and skirts. This page served an important part in giving a thorough analysis of customers' sentiments by clothing category, giving insightful information about which clothing that have positive, neutral, or negative sentiments.

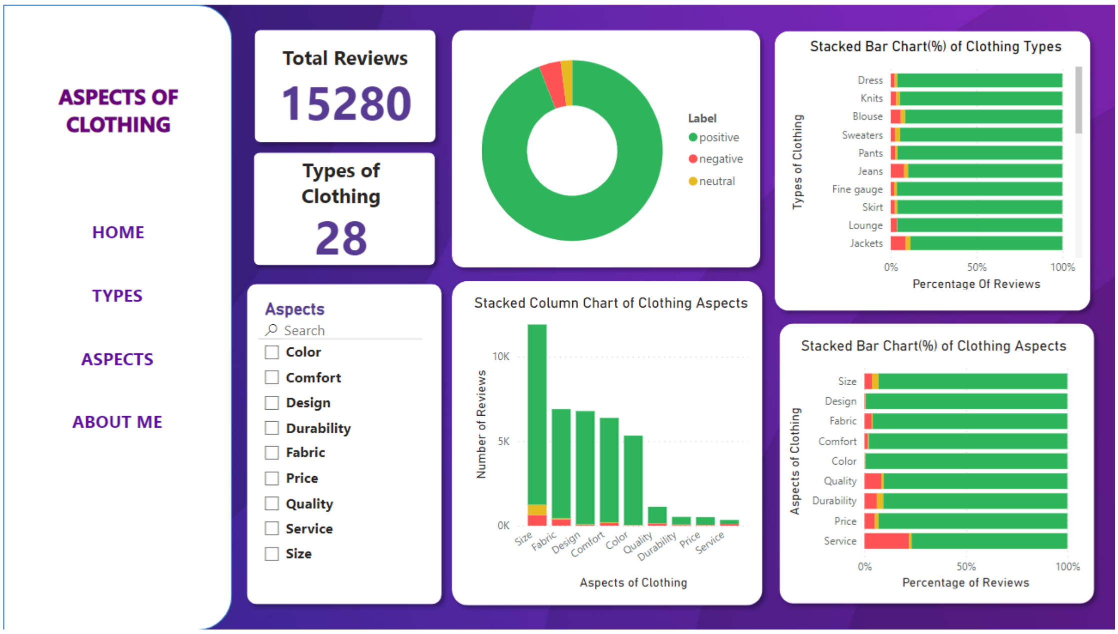


Figure 4.38 Aspects of Clothing Page

On the third page of the Clothing Review Analysis Dashboard, a comprehensive comparison between the aspects and types of clothing was presented. This page continued to feature the total review count and the sentiment breakdown but introduced a new layer of analysis with two stacked bar charts, one for clothing types and one for clothing aspects. The stacked bar chart for clothing types was included to offer a direct comparison with the aspects of clothing, providing a visual side-by-side analysis. This allowed users to compare how the sentiments were distributed across both the types of clothing, such as dresses, blouses, or suits, and the aspects like size, fabric, or design.

This comparison chart was very informative because it showed the sentiment trends within each category in addition to the number of reviews per category. A comprehensive view of the data was made possible by putting both clothing kinds and aspects on one page, which allowed for a thorough examination of customer sentiments. Through a comprehensive analysis of both types and aspects of clothing, users may identify connections and patterns that potentially impact customer sentiments and suggestions for improvements to the product.

Figure 4.39 About Me Page

The final page of the Clothing Review Analysis Dashboard, as depicted in the image, served as an 'About Me' section, introducing the academic background. The page also had a suggestion box for questions regarding the data and an input area for questions. In conclusion, the four pages provided a strong sentiment analysis tool. The dashboard began with a thorough summary of the data and then went into detail with parts dedicated to different types and aspects of clothing. Every page was carefully designed with visuals that offered insights into customer sentiments.

4.8 Summary

In the process of analysing sentiment in clothing reviews, a systematic approach was employed. A methodical methodology was used to analyse the sentiment found in apparel evaluations. To guarantee quality, data was initially collected and pre-processed. Aspect extraction was used to find key points in the reviews. Sentiment analysis revealed that over 15,678 instances of the TextBlob generated sentiment labels matched the pre-existing labels, whereas about 12,788 examples showed mismatches. Reviews containing mismatched labels were eliminated in order to preserve the integrity of the modeling process. Originally, these labels were based on the ratings for the clothing, however, current online reviews have shown a systematic bias that has been reducing over time.

Three models were trained which is Naïve Bayes, RNN, and LSTM, with accuracy rates of 90.05%, 99.15%, and 98.77%, respectively. Five separate experiments were conducted using different optimizers, epochs, and batch sizes in order to further test the selected RNN model. With a high accuracy of 99.09%, experiment 5 was chosen to next testing. The sentiment analysis was deployed on a basic user interface using experiment 5 model trained. Finally, developing a Power BI dashboard to illustrate the data and provide a thorough overview of the sentiment distribution across different apparel items and review. The insights were made available to stakeholders by the dashboard, which combined the analysis's findings.

CHAPTER FIVE
CONCLUSION

In Chapter Five, the results and findings obtained from the project, emphasizing the achievement of its objectives, the demonstrated strengths, limitations encountered, and recommendations for future work. This chapter provided a thorough analysis of the project's results, emphasizing both its achievements and its room for development. Important insights that could guide future research and applications in the field of sentiment analysis for apparel evaluations were obtained through a thorough study of the results.

5.1 Achievement of Objectives

Several significant achievements were made in accomplishing the study's objectives. First of all, the study was able to identify the key aspects that frequently showed up in reviews of clothing. A thorough examination of appropriate research and review datasets resulted in the identification of key themes in the evaluations, including fabric, design, size, color, prices, comfort, quality, durability, and service.

Second, a noteworthy achievement was the use of Aspect-Based Sentiment Analysis (ABSA) to assessments of apparel. About 44.92% (12788) of the reviews were difficult to process because they were excessively short or did not include enough aspect mentions, but the other 55.08% (15678) could be handled well with ABSA methods. Recurrent neural networks (RNN), long short-term memory (LSTM) networks, and naive bayes are among the ABSA models whose accuracy frequently exceeded 90%, demonstrating the method's robustness.

The final goal of the project was successfully completed with the development of a dashboard for aspect-based sentiment analysis of clothing reviews using Power BI. In order to give stakeholders an easy-to-use tool for gaining insights from the apparel reviews, the dashboard offered a simple platform for viewing and analyzing the sentiment analysis results.

All things considered, the study accomplished a great deal in reaching its goals and offered insightful information and useful instruments for assessing sentiment in apparel reviews. These successes open up new avenues for investigation and use in the field of sentiment analysis in consumer goods.

5.2 Strength of Project

The project's strength was its ability to train sentiment analysis models for clothing reviews with a high degree of accuracy. The models, which included Naive Bayes, Long Short-Term Memory (LSTM) networks, and Recurrent Neural Networks (RNN), regularly showed accuracy rates of over 90%, demonstrating their dependability in assessing the sentiment related to different aspects of clothing.

The project's strength was further increased by the creation of a dashboard utilizing Power BI to analyze the elements of clothing reviews. Through the use of this dashboard's intuitive layout, stakeholders were able to visualize and analyze the sentiment analysis results and obtain insightful knowledge about how customers feel about various aspects of clothing. Sentiment analysis for clothing evaluations was better understood and used in the project more effectively overall because of the combination of precise models and an easy-to-use dashboard.

5.3 Limitation of Project

Several limitations affected the project's scope and outcomes. First off, the dataset utilized for training and assessment had difficulties because several reviews were short or lacked information about specific aspects of clothing. This could make it difficult to adequately convey the feeling connected to each aspect of clothing products. Furthermore, the imbalance between positive, negative, and neutral sentiment instances in the dataset might have affected the model's ability to generalize well across different sentiment classes.

Another issue involved the aspect extraction process, as it's possible that some aspects were unsuccessfully taken out of the reviews. The difficulty in locating and obtaining subtle elements from unstructured text data may be the cause of this. Moreover, it's possible that not all aspects related to clothing reviews were covered by the predetermined list of aspects utilized for extraction, which would have resulted in partial aspect coverage in the analysis.

Overall, these limitations demonstrated the necessity of better aspect extraction methods, a more complete aspect list, and a more balanced dataset in order to increase the sentiment analysis model's precision and coverage.

5.4 Future Work and Recommendations

The performance of the algorithm could be greatly increased for upcoming projects by improving the dataset utilized for sentiment analysis of clothing assessments. It would be more beneficial to obtain a larger and more evenly balanced dataset that includes a wider range of reviews that go into greater detail about various aspects such as fabric, design, size, color, pricing, comfort, quality, durability, and service. A dataset with diverse reviews would enable the model to generalize better across different sentiments and aspects, enhancing its overall effectiveness in analyzing clothing reviews.

Enhancing the precision and coverage of extracted aspects also requires fine-tuning the aspect extraction process. This refinement could involve leveraging more advanced natural language processing (NLP) techniques, such as semantic analysis or deep learning-based approaches, to capture nuanced aspects more effectively. One potential solution to guarantee the extraction of pertinent and precise elements associated with the fashion industry is to employ domain-specific dictionaries specifically designed for clothing reviews. The algorithm can more accurately detect and evaluate the important features of clothing reviews by improving the aspect extraction procedure, which will produce more accurate sentiment analysis findings.

REFERENCES

AbdulNabi, I., & Yaseen, Q. (2021). Spam email detection using deep learning techniques. *Procedia Computer Science*, *184*, 853–858. https://doi.org/10.1016/j.procs.2021.03.107

Almansour, A., Alotaibi, R., & Alharbi, H. (2022). Text-rating review discrepancy (TRRD): an integrative review and implications for research. *Future Business Journal*, *8*(1). https://doi.org/10.1186/s43093-022-00114-y

Aqlan, A. A. Q., Manjula, B., & Lakshman Naik, R. (2019). A study of sentiment analysis: Concepts, techniques, and challenges. In *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 28, pp. 147–162). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-981-13-6459-4_16

Behdenna, S., Barigou, F., & Belalem, G. (2018). Document Level Sentiment Analysis: A survey. *EAI Endorsed Transactions on Context-Aware Systems and Applications*, *4*(13), 154339. https://doi.org/10.4108/eai.14-3-2018.154339

Chang, Y.-C. (2020). Precision Marketing Strategies for Small and Medium-sized Clothing E-commerce Enterprises Based on Big Data. *Business Prospects*, *1*(1), 128–139. https://doi.org/10.52288/bp.27089851.2020.12.11

Cheng, Y., Yao, L., Xiang, G., Zhang, G., Tang, T., & Zhong, L. (2020). Text Sentiment Orientation Analysis Based on Multi-Channel CNN and Bidirectional GRU with Attention Mechanism. *IEEE Access*, *8*, 134964–134975. https://doi.org/10.1109/ACCESS.2020.3005823

Dang, C. N., Moreno-García, M. N., & De La Prieta, F. (2021). Hybrid Deep Learning Models for Sentiment Analysis. *Complexity*, *2021*. https://doi.org/10.1155/2021/9986920

Dhankhar, A., Mohammad, S., Gupta, A., Dhankar, A., Dabas, S., Tech, M., & Delhi, M. (2018). SENTIMENT ANALYSIS USING MACHINE LEARNING: A REVIEW. In *Article in SSRN Electronic Journal* (Vol. 5, Issue 2). http://www.ijcrt.org/papers/IJCRT1134113.pdf

Eckman, M., Damhorst, M. L., & Kadolph, S. J. (2020). Toward a Model of the In-Store Purchase Decision Process: Consumer Use of Criteria for Evaluating Women's Apparel. *Clothing and Textiles Research Journal*, *8*(2), 13–22. https://doi.org/10.1177/0887302X9000800202

Ferrari, A., & Esuli, A. (2019). An NLP approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*. https://doi.org/10.1007/s10515-019-00261-7

Flek, L. (2020). *Returning the N to NLP: Towards Contextually Personalized Classification Models*.

Fresneda, J. E., & Gefen, D. (2019). A semantic measure of online review helpfulness and the importance of message entropy. *Decision Support Systems*, *125*, 113117. https://doi.org/10.1016/J.DSS.2019.113117

Fu, X., Yang, J., Li, J., Fang, M., & Wang, H. (2018). Lexicon-Enhanced LSTM with Attention for General Sentiment Analysis. *IEEE Access*, *6*, 71884–71891. https://doi.org/10.1109/ACCESS.2018.2878425

Hakkinen, M., Wong, F. A., Anggreainy, M. S., & Hidayat, W. R. (2021). Sentiment Analysis of E-commerce Review using Lexicon Sentiment Method. *Proceedings of 2021 1st International Conference on Computer Science and Artificial Intelligence, ICCSAI 2021*, 68–71. https://doi.org/10.1109/ICCSAI53272.2021.9609786

Hassonah, M. A., Al-Sayyed, R., Rodan, A., Al-Zoubi, A. M., Aljarah, I., & Faris, H. (2020). An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowledge-Based Systems*, *192*, 105353. https://doi.org/10.1016/J.KNOSYS.2019.105353

Jain, P. K., Yekun, E. A., Pamula, R., & Srivastava, G. (2021). Consumer recommendation prediction in online reviews using Cuckoo optimized machine learning models. *Computers and Electrical Engineering*, *95*, 107397. https://doi.org/10.1016/J.COMPELECENG.2021.107397

Kim, H. joon, Kim, J., Kim, J., & Lim, P. (2018). Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning. *Neurocomputing*, *315*, 128–134. https://doi.org/10.1016/J.NEUCOM.2018.07.002

Kit, B. W. S., & Joseph, M. H. (2023). Aspect-Based Sentiment Analysis on Movie Reviews. *Proceedings - International Conference on Developments in ESystems Engineering, DeSE*, *2023-January*, 237–243. https://doi.org/10.1109/DeSE58274.2023.10099815

Kumar, -Vivek, Khan, -Mohammed Naved, & Kesharwani, -Subodh. (2019). online customer Satisfaction and loyalty in apparel purchase: a review of Extant literature. *Global Journal of Enterprise Information System*.

https://doi.org/10.18311/gjeis

Li, X., Bing, L., Zhang, W., & Lam, W. (2019). *Exploiting BERT for End-to-End Aspect-based Sentiment Analysis *.*

Lu, B., Ma, B., Cheng, D., & Yang, J. (2023). An Investigation on Impact of Online Review Keywords on Consumers' Product Consideration of Clothing. *Journal of Theoretical and Applied Electronic Commerce Research*, *18*(1), 187–205. https://doi.org/10.3390/jtaer18010011

Mahmud, M., Mahmud, F. A. M., Rafit, S. B., Mullick, A., & Anas, T. C. M. (2023). *Sentiment Analysis of Women's Clothing Reviews on E-commerce Platforms: A Machine Learning Approach*. https://www.researchgate.net/publication/371608341

Marwat, M. I., Khan, J. A., Dahman Alshehri, M., Ali, M. A., Hizbullah, Ali, H., & Assam, M. (2022). Sentiment Analysis of Product Reviews to Identify Deceptive Rating Information in Social Media: A SentiDeceptive Approach. *KSII Transactions on Internet and Information Systems*, *16*(3), 830–860. https://doi.org/10.3837/tiis.2022.03.005

Meng, J., Long, Y., Yu, Y., Zhao, D., & Liu, S. (2019). Cross-domain text sentiment analysis based on CNN_FT method. *Information (Switzerland)*, *10*(5). https://doi.org/10.3390/info10050162

Mohammed Almosawi, M., & Abdulbaki Mahmood, S. (2022). *Lexicon-Based Approach For Sentiment Analysis To Student Feedback Arabic Processing View project Health DB View project*. http://www.webology.org

Nawaz, Z., Zhao, C., Nawaz, F., Safeer, A. A., & Irshad, W. (2021). Role of artificial neural networks techniques in development of market intelligence: a study of sentiment analysis of ewom of a women's clothing company. *Journal of Theoretical and Applied Electronic Commerce Research*, *16*(5), 1862–1876. https://doi.org/10.3390/jtaer16050104

Nguyen, J., & Chaturvedi, R. (2020). Quarantine Quibbles: A Sentiment Analysis of COVID-19 Tweets. *11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference, IEMCON 2020*, 346–350. https://doi.org/10.1109/IEMCON51383.2020.9284875

Permana, D., & Dewi, S. L. (2020). *PREFERENSI KONSUMEN PADA MEREK PAKAIAN ROPU MENGGUNAKAN ANALISA KONJOIN*.

Purwaningtyas, A., & Rahadi, R. A. (2021). THE AFFECTING FACTORS ON

ONLINE CLOTHING PURCHASE: A CONCEPTUAL MODEL. *Advanced International Journal of Business, Entrepreneurship and SMEs*, *3*(8), 86–97. https://doi.org/10.35631/aijbes.38006

Rahman, A. (2017). *Factors Influencing Clothing Interest and Purchase Intention: A Study of Generation Y Consumers in Malaysia*.

Rao, G., Huang, W., Feng, Z., & Cong, Q. (2018). LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*, *308*, 49–57. https://doi.org/10.1016/J.NEUCOM.2018.04.045

Sagarino, V. M. C., Montejo, J. I. M., & Ceniza-Canillo, A. M. (2022). Sentiment Analysis of Product Reviews as Customer Recommendations in Shopee Philippines Using Hybrid Approach. *Proceedings of the 2022 IEEE 7th International Conference on Information Technology and Digital Applications, ICITDA 2022*. https://doi.org/10.1109/ICITDA55840.2022.9971379

Sahar, A., Ayoub, M., Hussain, S., Yu, Y., & Khan, A. (2022). *Pakistan Journal of Engineering and Technology, PakJET Transfer Learning-Based Framework for Sentiment Classification of Cosmetics Products Reviews*.

Saunders, D., & Byrne, B. (2020). *Domain adaptation for Neural Machine Translation*.

Singh, M., Jakhar, A. K., & Pandey, S. (2021). Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Social Network Analysis and Mining*, *11*(1). https://doi.org/10.1007/s13278-021-00737-z

Tanvir Abir, Md Adnan Rahman, Muhammad Nur-A Yazdani, Rubaiyet Hasan Khan, Syeda Sabrina, & Abu Bakar Abdul Hamid. (2020). Electronic Word of Mouth (e-WOM) and consumers' purchase decisions: Evidences from Bangladesh. *JOURNAL OF XI'AN UNIVERSITY OF ARCHITECTURE & TECHNOLOGY*, *XII*(III). https://doi.org/10.37896/jxat12.04/782

Tran, T., Ba, H., & Huynh, V. N. (2019). Measuring hotel review sentiment: An aspect-based sentiment analysis approach. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11471 LNAI*, 393–405. https://doi.org/10.1007/978-3-030-14815-7_33

Urriza, I. M., & Antonette Clarino, M. A. (2021). *ASPECT-BASED SENTIMENT ANALYSIS OF USER CREATED GAME REVIEWS*.

Waghmare, K. A., & Bhala, S. k. (2020). *2020 International Conference on Computer*

Communication and Informatics : January 22-24, 2020, Coimbatore, India.

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022a). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, *55*(7), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022b). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, *55*(7), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

Yadav, H., Dwivedi, K., & Abirami, G. (2022). Sentiment Analysis of company reviews using Machine Learning. *2022 3rd International Conference for Emerging Technology, INCET 2022*. https://doi.org/10.1109/INCET54531.2022.9824505

Yousefi Dahka, Z., Hajiheydari, N., & Rouhani, S. (2019). User response to e-WOM in social networks: how to predict a content influence in Twitter. *International Journal of Internet Marketing and Advertising*, *13*(1), 1. https://doi.org/10.1504/ijima.2019.10024249

Yu, L., & Bai, X. (2021). Implicit Aspect Extraction from Online Clothing Reviews with Fine-tuning BERT Algorithm. *Journal of Physics: Conference Series*, *1995*(1). https://doi.org/10.1088/1742-6596/1995/1/012040

APENDICES

APPENDIX 1

Dataset

A	B	C	D	E	F	G	H	I	J	K
	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	767	33		Absolutely wonder	4	1	0	Initmates	Intimate	Intimates
1	1080	34		Love this dress! it's	5	1	4	General	Dresses	Dresses
2	1077	60	Some major design flaws	I had such high hop	3	0	0	General	Dresses	Dresses
3	1049	50	My favorite buy!	I love, love, love thi	5	1	0	General Petite	Bottoms	Pants
4	847	47	Flattering shirt	This shirt is very fla	5	1	6	General	Tops	Blouses
5	1080	49	Not for the very petite	I love tracy reese dı	2	0	4	General	Dresses	Dresses
6	858	39	Cagrcoal shimmer fun	I aded this in my ba	5	1	1	General Petite	Tops	Knits
7	858	39	Shimmer, surprisingly go	I ordered this in caı	4	1	4	General Petite	Tops	Knits
8	1077	24	Flattering	I love this dress. i u	5	1	0	General	Dresses	Dresses
9	1077	34	Such a fun dress!	I'm 5"5' and 125 lb	5	1	0	General	Dresses	Dresses
10	1077	53	Dress looks like it's made	Dress runs small es	3	0	14	General	Dresses	Dresses
11	1095	39		This dress is perfect	5	1	2	General Petite	Dresses	Dresses
12	1095	53	Perfect!!!	More and more i fiı	5	1	2	General Petite	Dresses	Dresses
				Bought the black xs to go under the larkspur midi dress because they didn't bother lining the skirt portion (grrrrrrrrrrr). my stats are 34a-						

APPENDIX 2

Business Model Canvas

Topic: Aspect-Based Sentiment Analysis of Clothing Reviews

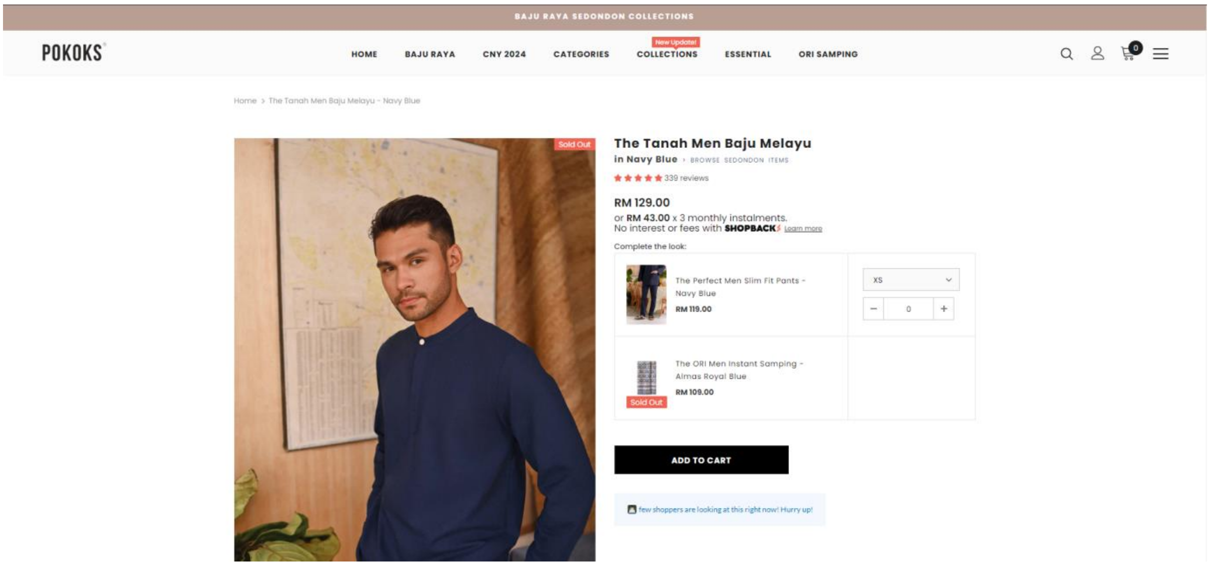
Problem	Solution	Unique Value Proposition	Unfair Advantage	Customer Segments
- There is a large quantity of data for analysis. - Manual review analysis takes a lot of time to complete. - Reviews may contain multiple aspects. - Ratings may not accurately reflect reviews.	- The model employs aspect-based sentiment analysis. - The model utilizes a lexicon-based approach. - The project features a user-friendly dashboard.	By providing hidden insights that help organizations better understand their customers and make data-driven decisions through the dashboard, the project also reveals the underlying sentiment behind ratings.	Detailed analysis and benchmarking are made possible by having access to a large database of evaluations and ratings from a variety of industries.	- Apparel Company - Fashion E-commerce Brands

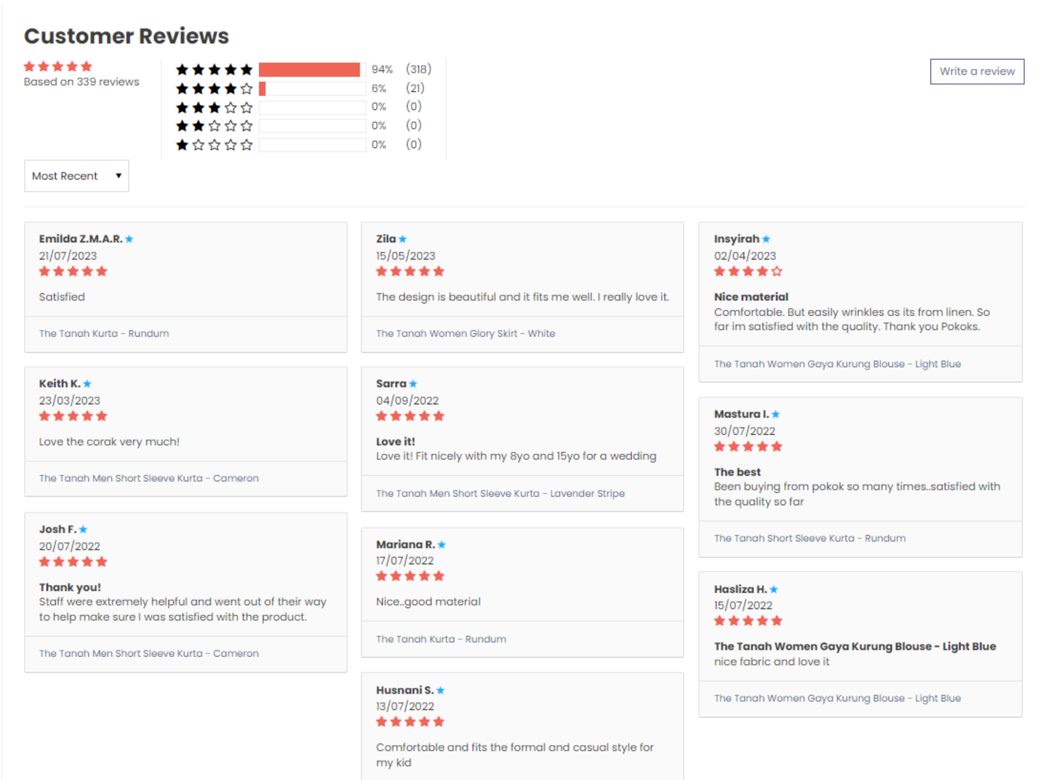
Existing Alternatives	Key Metrics	High-Level Concept	Channels	Early Adopters
- Sentiment Analysis with different techniques - Manual analysis of the reviews	- The revenue from the monthly subscription of the project. - Feedback of the customers.	FashionTrace	- Social Media - Industry events	- Small Boutique Brand - E-commerce Review Sites

Cost Structure	Revenue Structure
- Development Costs - Data Storage and Processing - Travel and Events - Utilities - Marketing and Sales	- Subscription model - Pay-Per-Use or Transaction Fees - Licensing and API Fees - Data Licensing

APPENDIX 3

POKOKS official website





APPENDIX 4

Naïve Bayes Code

In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Load your dataset
df = pd.read_csv('dataset_modelling.csv')
```

In [3]:
```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['Review'], df['Label'], test_size=0.2, random_state=42)
```

In [4]:
```
# Vectorize the text data using TF-IDF
vectorizer = TfidfVectorizer(max_features=1000)  # Adjust max_features as needed
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

In [5]:
```
# Train a Naive Bayes classifier
clf = MultinomialNB()
clf.fit(X_train_vectorized, y_train)
```

Out[5]:
```
▾ MultinomialNB
MultinomialNB()
```

In [6]:
```
# Make predictions
y_pred = clf.predict(X_test_vectorized)
```

In [7]:
```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print("Classification Report:\n", classification_rep)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

```
Classification Report:
              precision    recall  f1-score   support

    negative       1.00      0.07      0.13       194
     neutral       0.00      0.00      0.00       135
    positive       0.90      1.00      0.95      2847

    accuracy                           0.90      3176
   macro avg       0.63      0.36      0.36      3176
weighted avg       0.87      0.90      0.86      3176

Accuracy: 90.05%
```

In [8]:
```
# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Positive', 'Negative', 'Neutral'], yticklabels=['Positi
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

APPENDIX 5

RNN Code

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense
import matplotlib.pyplot as plt
import seaborn as sns
```

```
WARNING:tensorflow:From C:\Users\Sanchan\anaconda3\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softma
x_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

In [2]:
```
# Load the dataset
file_path = 'dataset_modelling.csv'
data = pd.read_csv(file_path)
```

In [3]:
```
# Prepare data
text = data['Review'].astype(str)
labels = data['Label']
```

In [4]:
```
# Convert labels to categorical format
label_dict = {'positive': 0, 'negative': 1, 'neutral': 2}
labels = labels.map(label_dict)
labels = np.array(labels)
```

In [5]:
```
# Tokenization and sequence padding
tokenizer = Tokenizer()
tokenizer.fit_on_texts(text)
sequences = tokenizer.texts_to_sequences(text)
word_index = tokenizer.word_index
max_len = max([len(seq) for seq in sequences])
data = pad_sequences(sequences, maxlen=max_len)
```

In [6]:
```
# Splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
```

In [7]:
```
# Model architecture
model = Sequential()
model.add(Embedding(len(word_index) + 1, 128, input_length=max_len))
model.add(LSTM(128))
model.add(Dense(3, activation='softmax'))

# Compile the model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
WARNING:tensorflow:From C:\Users\Sanchan\anaconda3\Lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is
deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\Sanchan\anaconda3\Lib\site-packages\keras\src\optimizers\__init__.py:309: The name tf.train.Op
timizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
```

In [8]:
```
# Model training
history = model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test), verbose=1)
```

In [9]:
```
# Model evaluation and prediction
y_pred_probabilities = model.predict(X_test)
y_pred = np.argmax(y_pred_probabilities, axis=1)
```

```
100/100 [==============================] - 6s 59ms/step
```

In [10]:
```
# Generate classification report
class_names = ['positive', 'negative', 'neutral']
print(classification_report(y_test, y_pred, target_names=class_names))


# Calculate and print accuracy score
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

In [11]:
```
# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label_dict.keys(), yticklabels=label_dict.keys())
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

In [12]:
```
# Plotting performance
plt.figure(figsize=(8, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()
plt.show()
```

In [13]:
```
# Plotting performance
plt.figure(figsize=(8, 6))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()
plt.show()
```

APPENDIX 6

LSTM Code

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, SpatialDropout1D, LSTM, Dense
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```
```
WARNING:tensorflow:From C:\Users\Sanchan\anaconda3\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softma
x_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

In [2]:
```
# Load the data
data = pd.read_csv('dataset_modelling.csv')
reviews = data['Review'].tolist()
labels = data['Label'].tolist()
```

In [3]:
```
# Mapping labels to numerical values
label_dict = {'positive': 0, 'negative': 1, 'neutral': 2}
labels = [label_dict[label] for label in labels]
```

In [4]:
```
# Splitting into training and validation sets
train_text, val_text, train_labels, val_labels = train_test_split(reviews, labels, test_size=0.2, random_state=42)
```

In [5]:
```
# Tokenization and sequence padding
max_features = 10000  # Set the maximum number of words to tokenize
max_len = 100  # Set the maximum sequence length
tokenizer = Tokenizer(num_words=max_features)
tokenizer.fit_on_texts(train_text)
train_sequences = tokenizer.texts_to_sequences(train_text)
val_sequences = tokenizer.texts_to_sequences(val_text)
train_sequences = pad_sequences(train_sequences, maxlen=max_len)
val_sequences = pad_sequences(val_sequences, maxlen=max_len)
```

In [6]:
```
# Convert labels to numpy arrays
train_labels = np.array(train_labels)
val_labels = np.array(val_labels)
```

In [7]:
```
# Define the LSTM model for multi-class classification
num_classes = 3  # Three classes: positive, negative, neutral
model = Sequential()
model.add(Embedding(max_features, 128, input_length=max_len))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(num_classes, activation='softmax'))  # Use softmax for multi-class classification
```
```
WARNING:tensorflow:From C:\Users\Sanchan\anaconda3\Lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is
deprecated. Please use tf.compat.v1.get_default_graph instead.
```

In [8]:
```
# Compile the model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```
```
WARNING:tensorflow:From C:\Users\Sanchan\anaconda3\Lib\site-packages\keras\src\optimizers\__init__.py:309: The name tf.train.Op
timizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
```

In [9]:
```
# Train the model
history = model.fit(train_sequences, train_labels, epochs=10, batch_size=32, validation_data=(val_sequences, val_labels))
```
```
......
```

In [10]:
```
# Make predictions on the validation set
val_predictions_probs = model.predict(val_sequences)
val_predictions = np.argmax(val_predictions_probs, axis=1)

# Generate classification report
report = classification_report(val_labels, val_predictions, target_names=label_dict.keys())

# Calculate accuracy
accuracy = accuracy_score(val_labels, val_predictions)
```
```
100/100 [==============================] - 3s 21ms/step
```

In [11]:
```
# Display the classification report and accuracy
print("Classification Report:")
print(report)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

In [12]:
```
# Confusion matrix
conf_matrix = confusion_matrix(val_labels, val_predictions)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label_dict.keys(), yticklabels=label_dict.keys())
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

In [13]:
```
# Plotting Training and Validation Accuracy and Loss
epochs = range(1, len(history.history['accuracy']) + 1)
plt.figure(figsize=(12, 5))
```

Out[13]:
```
<Figure size 1200x500 with 0 Axes>

<Figure size 1200x500 with 0 Axes>
```

In [14]:
```
# Plotting Training and Validation Accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

In [15]:
```
# Plotting Training and Validation Accuracy
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

APPENDIX 7

Prototype Code

In [14]:
```
import tkinter as tk
from tkinter import font
from tkinter import messagebox
import re
import numpy as np
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Assume tokenizer and max_len are already defined and trained model is available

def preprocess_input(text):
    # Text preprocessing steps
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    # Tokenization and sequence padding
    sequence = tokenizer.texts_to_sequences([text])
    sequence_padded = pad_sequences(sequence, maxlen=max_len)
    return sequence_padded

def predict_sentiment(text):
    preprocessed_text = preprocess_input(text)
    prediction = model.predict(preprocessed_text)
    predicted_class = int(np.argmax(prediction, axis=1))
    if predicted_class == 0:
        return 'Positive'
    elif predicted_class == 1:
        return 'Negative'
    elif predicted_class == 2:
        return 'Neutral'
```

```
def submit_action():
    text = text_entry.get("1.0", "end-1c")
    if text.strip() == "":
        messagebox.showerror("Error", "Please enter some text.")
    else:
        sentiment = predict_sentiment(text)
        messagebox.showinfo("Sentiment Analysis Result", f"The sentiment of the text is: {sentiment}")

def reset_action():
    text_entry.delete("1.0", "end")
```

In [15]:
```
import tkinter as tk
from tkinter import font
from tkinter import messagebox

def submit_action():
    text = text_entry.get("1.0", "end-1c")
    if text.strip() == "":
        messagebox.showerror("Error", "Please enter some text.")
    else:
        sentiment = predict_sentiment(text)
        messagebox.showinfo("Sentiment Analysis Result", f"The sentiment of the text is: {sentiment}")

def reset_action():
    text_entry.delete("1.0", "end")

# Create the main window
root = tk.Tk()
root.title("Sentiment Analysis of Clothing Reviews")

# Create a text input field
text_entry = tk.Text(root, height=10, width=50, wrap=tk.WORD)  # Set wrap option to tk.WORD
text_entry.pack()

# Create a frame to hold the buttons
button_frame = tk.Frame(root)
button_frame.pack()
```

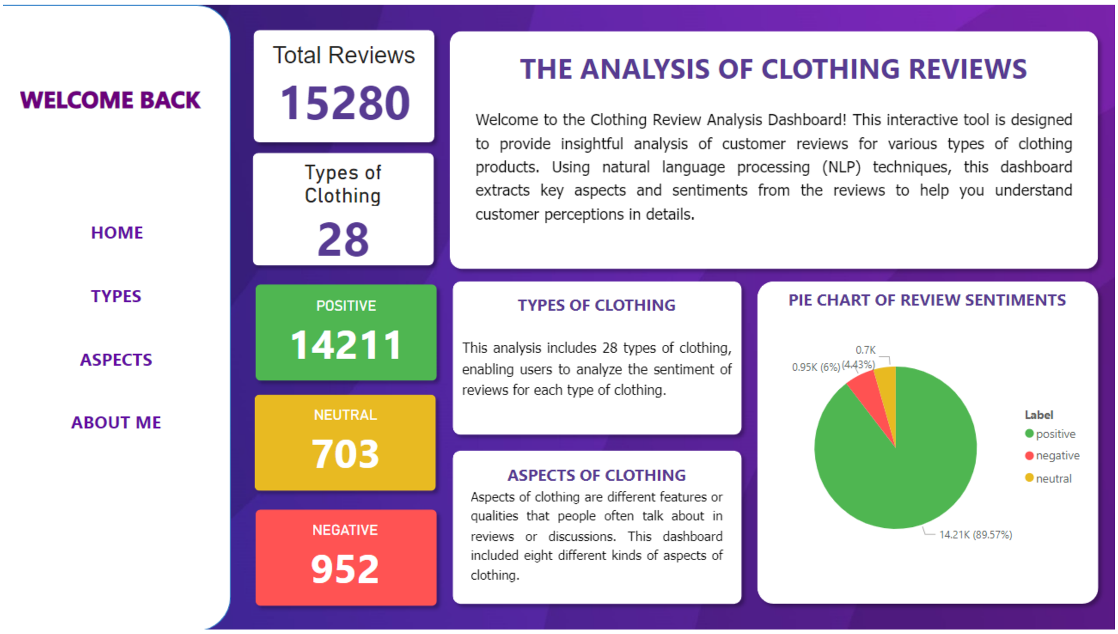
```
# Create a submit button
submit_button = tk.Button(button_frame, text="Submit", command=submit_action, font=("Space Grotesk", 12))
submit_button.pack(side=tk.LEFT, padx=5)

# Create a reset button
reset_button = tk.Button(button_frame, text="Reset", command=reset_action, font=("Space Grotesk", 12))
reset_button.pack(side=tk.LEFT, padx=5)

# Run the main loop
root.mainloop()
```

APPENDIX 8

Dashboard



Link:

https://app.powerbi.com/view?r=eyJrIjoiNGJiODk3ZTgtODY1NS00NmQ4LThmYTItY2U2ODNmMWY2Nzk0IiwidCI6ImNkY2JiMGUyLTlmZWEtNGY1NC04NjcwLTY3MjcwNzc5N2FkYSIsImMiOjEwfQ%3D%3D&pageName=ReportSectiond8c927cd92d2a000ad30

APPENDIX 9

F5 Document

CSP600/650	College of Computing, Informatics and Mathematics

F5 – PROPOSAL/PROJECT IN-PROGRESS FORM

STUDENT NAME	MUHAMMAD EHSAN BIN MAT SUKURY	**STUDENT ID**	2022930715
PROGRAM	CS259: BACHELOR OF INFORMATION SYSTEMS (HONS.) INTELLIGENT SYSTEM ENGINEERING		
SUPERVISOR	MADAM NURZEATUL HAMIMAH BINTI ABDUL HAMID		
TITLE	ASPECT-BASED SENTIMENT ANALYSIS OF CLOTHING REVIEWS		

DATE OF MEETING	COMPLETED ACTIVITY	SUPERVISOR/CO-SUPERVISOR	
		NEXT ACTIVITY/COMMENT	SIGNATURE
24/4/2023	Online meeting and presentation of SCIPB and final year project idea with the supervisor	Decide topic and find article related to final year project.	
4/4/2023	Request sign from supervisor for F1 document	None	
13/4/2023	Meeting with supervisor regarding F2 document	Give a guide how to make problem statement and find article related to the problems	
19/5/2023	Meeting with supervisor regarding mind map	None	
9/6/2023	Online meeting with supervisor regarding Chapter 1 and Research Methodology table	Make a correction of problem statement, research objectives and research methodology table.	
19/10/2023	Update progress of final year project for the new semester	Supervisor asked to make a document that contains steps and update the progress of each step	
30/10/2023	I had a brief meeting with my supervisor after her class to inform her that I had changed the method and algorithm for my final year project.	I was asked to proceed with my changed method and algorithm.	
11/1/2023	I showed my supervisor the whole progress of my final year project and she checked my draft report and pointed out areas that needed improvement and correction.	Make the corrections to the report based on her feedback and submitted it to her again.	

APPENDIX 10

F6 Document

CSP600/CSP650 College of Computing, Informatics and Mathematics

UNIVERSITI TEKNOLOGI MARA

F6 -REPORT SUBMISSION FORM

Instructions to student:

1. Ensure that the information needed in the form is complete before submission to CSP650 lecturer.
2. Obtain certification by Supervisor that the report has been screened for plagiarism.
3. Please attach the Originality of plagiarism Report.
4. Only complete form will be processed.

STUDENT NAME	MUHAMMAD EHSAN BIN MAT SUKURY	**STUDENT ID**	2022930715
PROGRAM	CS259: BACHELOR OF INFORMATION SYSTEMS (HONS.) INTELLIGENT SYSTEM ENGINEERING		
SUPERVISOR	MADAM NURZEATUL HAMIMAH BINTI ABDUL HAMID		
CO- SUPERVISOR (IF ANY)			
PROJECT TITLE	ASPECT-BASED SENTIMENT ANALYSIS OF CLOTHING REVIEWS		
HANDOVER DATE	23/1/2024		
STUDENT'S SIGNATURE			

Student is required to get endorsement from the following parties:

I certify that the final year project report has been screened for plagiarism and the originality of report is enclosed.

Similarity index: 1%

Approved by:

SUPERVISOR: ____________________ (Signature)   DATE: 24/1/24