SENTIMENT ANALYSIS OF AMAZON PRODUCT REVIEWS USING MACHINE LEARNING

OMAR MOHAMMED ALI ALBAAGARI

UNIVERSITI TEKNOLOGI MALAYSIA

ABSTRACT

A large number of customer reviews are available for online shopping; however, the majority of these reviews are not being utilized because they are not categorized. In order to make sense of this data, you will need to make use of sophisticated computer algorithms that are able to comprehend feelings and transform them into information that can be utilized. This research makes use of machine learning algorithms to analyze reviews from Amazon's Cell Phones and Accessories category. The objective of the study is to categorize comments as either positive, neutral, or negative, and to present the findings in a manner that is suitable for business users. Two classifiers, namely a Multinomial Naïve Bayes classifier and a Linear Support Vector Classifier, were successfully created and evaluated by our team. In order to improve the performance of the models, they were trained with a feature set that included TF-IDF, Chi-Square selection, and various other pieces of linguistic and structural information. These included part-of-speech counts, review metadata, and polarity lexicons.Obtaining a score of 0.816 on the F1 scale and a correctness rate of 95%, the Linear SVC performed the best. With an F1-score of 0.816 and an accuracy of 82%, the Naïve Bayes model performed the second best out of all the models presented. The Naïve Bayes algorithm was a great starting point; nevertheless, it did not perform well when it came to making unbiased assessments. The SVC, on the other hand, had outcomes that were more evenly distributed among all classes. In order to build interactive dashboards that corresponded with the classification, power BI dashboards were deployed. These dashboards made it simple for stakeholders to see what consumers were saying by displaying general sentiment trends, comparisons at the brand level, and words that were often used.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Background

Online purchase and online shopping of various commodities have seen a tremendous boost since there is no time in this fast age. There appears to be a sunrise of various e-shopping websites such as Amazon for meeting the needs of customers from almost any location, home/office/vacation etc. Customers always look for quality products at the lowest cost. Because the product or physical product cannot be inspected for quality within an e-commerce transaction, thereby customers rely on other customers' feedback. Such feedback does, however, influence the final choice of a customer to buy or not. More effectively, analyse this, the need for sentiment analysis has been met so that a product's popularity can be easily notified among customers.

Sentiment Analysis comes under natural language processing (NLP) and also known as opinion mining where several methods have been applied in Sentiment Analysis, such as Recurrent Neural Networks, hybrid methods, Ensembled methods, deep learning (Singhal et al., 2025). A method operates on the text data to attain the emotional tone (positive, negative, or neutral) of the writer provided in the text (Sanagar & Gupta, 2020). On an online shopping website, there are websites where the product review is done by the customers. Consumers and vendors of new goods read these reviews to understand the quality, popularity and other comments of the product. Sentiment analysis of the customers has become extremely crucial for the smooth running of any business. Customers can express their views and feelings about a product and service much more openly more than ever. It is highly unrealistic for a human to verify every single line to comprehend the user experience. So, the customer feedback analysis is very easily possible automatically with the newer methods and technology (Singhal et al., 2025).

Almost all consumable products are now available for purchase over the internet. For current research, it has been chosen mobile phones & accessories and reviews. The inevitability and omnipresence of mobile phone use among almost all human adults which to choose it as our study. During the purchase of a mobile, customers like to view the opinions of other customers. reviews. Not only do reviews help other customers in purchasing, rather they provide extremely helpful information to the product sellers too. The salespeople of the products would have to read hundreds or thousands of product reviews of similar products. The job is extremely complicated and tiring to accomplish manually. At times the score rating is good but the sentiment of the review is seen to be negative. In this scenario, the correct sentiment can only be known by reading the review better. Therefore, the aim of this paper is to conduct a sentiment analysis of reviews on mobile phones, where this complex process can be simplified by using machine learning.

1.2 Problem Statement

Customers feedback throws back the customer belief about the products. It can be whether positive, negative, or neutral. It's have been affected on the purchasing process to the customer and the product innovation to the firm. In Amazon platform offers the facility to the customer for giving feedback on the product that is being bought which reflects the real-time customer experience allowing companies to gauge the product quality and performance. Processing bulk of textual dataset, however, becomes useless, extremely likely to produce human error, and time-consuming. here where the sentiment analysis can solve this problem by classify the customer reviews into three broad categories which are positive, negative, and neutral categories. With this the businesses can keep watch on the product and make precious inference from the customer. Additionally, the businesses can learn the customer behavior from the sentiment analysis which can enable the businesses to predict the next buy of the customer.

Although several existing studies have already achieved over 90% accuracy in sentiment classification using machine learning models, many of these works focus

solely on the performance of the classifier itself and some of the feature engineering that been used such as POS. In this project five feature engineering being considered which are class sampling, review – based features, helpfulness features, POS tag features, and TF-IDF with N-grams feature. All of these features will help the model to capture and bringing higher accuracy. Each of these features will be discussed in detailed in the feature engineering section. Furthermore, the previous studies often overlook how the results can be communicated to stakeholders in an interpretable and actionable manner. This research aims to bridge that gap by not only applying Naïve Bayes and Support Vector Machine (SVM) models to classify sentiment in Amazon product reviews specifically in the cell phones and accessories category but also by integrating the results into an interactive Power BI dashboard. This added layer of visualization supports better decision making for the supplier in the Amazon website and the customer themselves.

From the given problem, the purpose of this research is to conduct a sentiment analysis on the Amazon cell phones and accessories products dataset with the objective of classifying the opinion of customers. The actual dataset from Amazon will be used by this research through the supported vector machine and Naïve Bayes models as a method of conducting the sentiment analysis.

1.3 Research Questions

This thesis seeks to illustrate consumer behaviour trends based on their purchases and reviews. Utilising the two distinct models reveals the consumers' reviews. A series of procedures must be undertaken to visualise the pattern of customer evaluations. Consequently, several research questions have been delineated for the experiment.

The research questions are:

a) What insights can be revealed about the distribution and characteristics of customer sentiments in Amazon cell phone and accessories reviews?

b) What preprocessing steps are needed for sentiment analysis of cell phone and accessories reviews, and how can SVM and Naïve Bayes improve sentiment classification

c) What conclusions may be derived from the customers purchases?

1.4 Research Objectives

a) To perform exploratory data analysis on the review dataset to uncover patterns, sentiment distribution, frequent keywords, and review length variations that inform the sentiment classification process.

b) To train a machine learning model to classify reviews into positive, neutral, or negative sentiments and compare model performance to identify the most accurate one.

c) To develop an interactive dashboard that summarize the analysis and making conclusion of the customer behaviour.

1.5 Scope of the Research

The scopes of the research are:

a) The data being collected from Amazon reviews prediction of cell phones and accessories Repository.

b) The programming language that is chosen is Python.

c) Implementing the Supported Vector Machine and Naïve Bayes models for the sentiment analysis.

d) Sorting the customer reviews into positive, neutral, and negative categories.

e) Building an interactive dashboard using PowerBi for the behaviour of the customers.

CHAPTER 2

LITERATURE REVIEW

2.1 Summary

The objective of this chapter is to introduce recent sentiment analysis literature reviews and research work. Introduction to certain levels of sentiment analysis is followed, which comprises data collection, data pre-processing, sentiment analysis techniques, and lastly, issues in the field. This chapter is an apt lead-in for the sentiment analysis as it explains the same in detail.

2.2 Trends of sentiment Analysis over Years

Sentiment analysis is becoming popular among scholars, businesses, governments, and organisations (Sánchez-Rada & Iglesias, 2019). With more individuals using the Internet, the World Wide Web has become the most important and widespread source of information. Millions of individuals use forums, blogs, wikis, social networks, and other online platforms to communicate their opinions and emotions (Jain et al.2019). The thoughts and feelings expressed by these individuals are important to our daily lives, thus user data must be investigated to automatically follow public opinion and help decision-making. Election outcomes have been predicted using Twitter tweets. Since the previous 15 years, research groups have been increasingly interested in sentiment analysis (Birjali et al. 2021). Since 2004, sentiment analysis has become the most dynamic intellectual field. This is because opinion mining and sentiment analysis articles have increased dramatically (Mantyla et al., 2018). Google Trends shows sentiment analysis popularity in Figure 2.1.

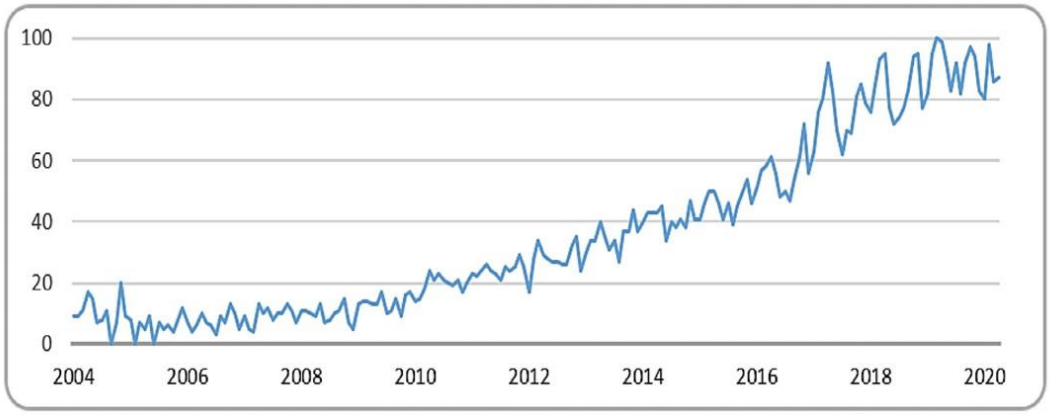

Figure 2.1 Curiosity in "Sentiment Analysis" since 2004 based on Google Trends.

(Mäntylä et al., 2018) utilized Google Scholar combined with Scopus when searching for most cited publications. That is owing to the fact that Scopus does not contain the publications issued in the initial years when there was no sentiment analysis. From table 2.1, as denoted shows the outcome of the (Mäntylä et al., 2018) work.

Table 2.1 In Scopus, search strings and the percentage of search hits are shown.

Search term	percentage of total hits
Sentiment analysis	68.5
Opinion mining	29.1
Sentiment classification	18
Opinion analysis	5.6

2.3 Sentiment Analysis Levels

A number of different degrees of investigation have been conducted on the task of sentiment analysis. On the other hand, feelings and views may be identified primarily at the level of the document, the level of the sentence, or the level of the aspect (Sánchez-Rada & Iglesias, 2019). The many levels of sentiment analysis are shown in Figure 2.2. Apart from being very difficult, the first two levels are also intriguing. On the other hand, the third level is more challenging than the first two since it does an in-depth study (Birjali et al., 2021). To provide a brief overview of each level, the following is a summary of each level:

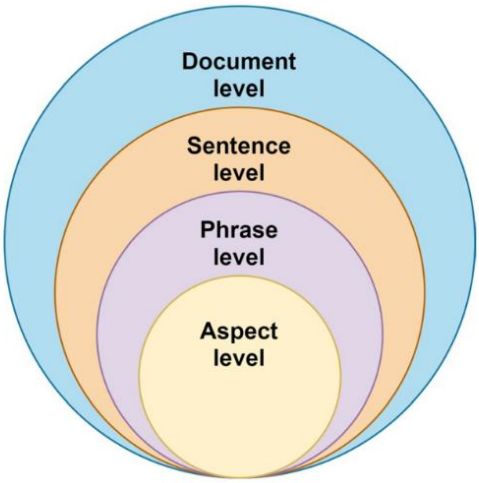

Figure 2.2 Levels of Sentiment Analysis (Wankhade et al., 2022).

2.3.1 Aspect level sentiment analysis

This level does fine-grained analysis to determine attitudes about certain characteristics. Example: "The camera of iPhone 11 is awesome." The review of the "iPhone 11," camera, is positive. This makes this activity useful for determining what people enjoy and hate. Instead of creating phrase or paragraph meaning, it focusses on objects, such a product's attributes. Aspect extraction—implicit or explicit—is the most important job in sentiment analysis (Wankhade et al., 2022).

There are numerous applications based in the real world that need this extent of in-depth research. Businesses, for example, identify which features or components

of the product are attractive to consumers in order to make improvements to the product (Birjali et al., 2021). There are a number of reasons for this.

2.3.2 Phrase level sentiment Analysis

There is also room to perform sentiment analysis at the phrase level, which involves extracting and that belongs to the class of decision-making words. There is room for one item or more items for each phrase. It is to be noted that one thing is expressed in a sentence (Thet et al., 2010), which can be helpful for product reviews that consist of multiple lines. This has been a popular topic of study being conducted lately. With that in mind. It is more useful to conduct sentence-level analysis compared to text-level analysis in analysing a document with positive and negative sentences. Text-level analysis deals with the classification of the entire text as subjectively positive or negative. Word is the basic unit of language, and the word polarity has a strong relationship with the subjectivity of the sentence or text that contains it. It is very probable that a sentence that contains an adjective in it is subjective (Fredriksen-Goldsen and Kim, 2017). Moreover, the chosen word that was ready to be spoken is typical of individuals' demographic characteristics like gender and age, interests, social status, and personality, among a myriad of other social and psychological factors. (Flek, 2020). Here, the word forms the basis for text sentiment translation.

2.3.3 Sentence level sentiment analysis

The sentence matters here. Whether the language conveys positive, negative, or neutral feelings is most crucial (B. Liu, 2012). Otherwise, the phrase must be objective, expressing facts, or subjective, expressing thoughts and views, to achieve this purpose. Multiple views were used for this investigation. Sentence type technique improved sentence-level sentiment analysis (Chen et al., 2017). They originally classified phrases into three kinds using a sequence model using neural networks. These were divided into sentences with zero targets, one targets, and phrases with more than one targets. Their classifier was a one-dimensional convolutional neural network

that was fed each text category in its own form. Document- and sentence-level sentiment analysis is important, but it doesn't cover all aspects of the entity (Medhat et al., 2014). Because it does not specify what people like or hate about the item.

2.3.4 Document level sentiment analysis

In this step, the goal is to identify whether a document has a favorable or negative impact (Alqaryouti et al., 2024). Each document is classed by the person's overall opinion on a product. Document-level categorization works best with single-author documents. It is ineffective for comparison or contrast publications comparing several objects on many degrees of difference. Many approaches have been suggested for document-level sentiment analysis. (Zhao et al., 2017) suggested a Domain-Independent Framework for Document-Level Sentiment Analysis using RST weighting principles. The approach was suggested for document-level sentiment analysis. The authors used two well-known lexicons to estimate phrase sentiment ratings after parsing the text into rhetorical structure trees. They added sentence ratings to assess the paper's mood polarity using weightage criteria. Sentiment analysis is useful for many applications, however sometimes a text contains contradictory emotions that may impact the outcome.

2.4 Sentiment analysis pre-processing

Sentiment analysis isn’t an issue; rather, it is a research challenge that may be described as a "suitcase," which means that it involves the completion of a number of natural language processing tasks (Cambria et al., 2017), as shown in Figure 2.3

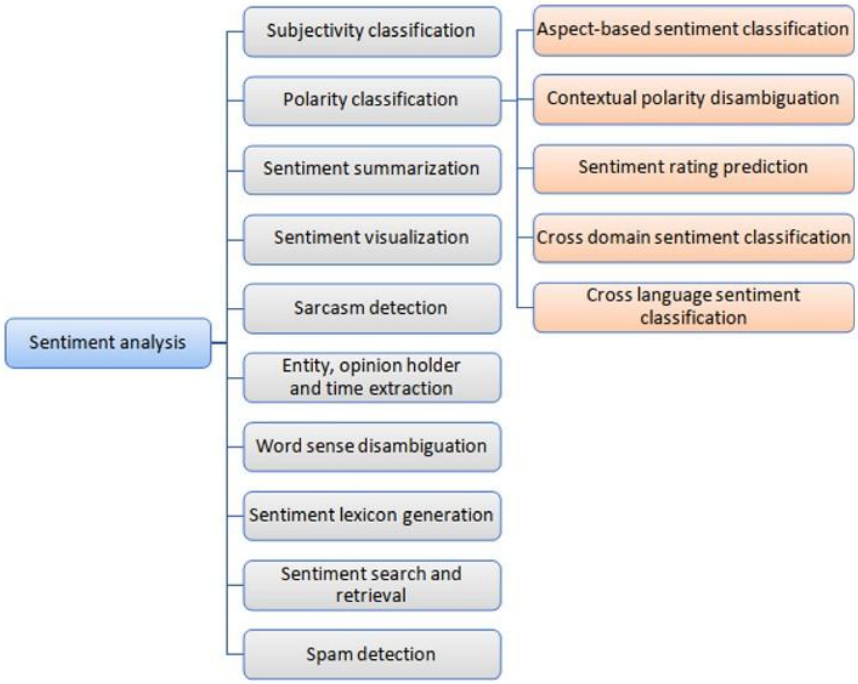

Figure 2.3 The most typical subtasks for sentiment analysis (Cambria et al., 2017)

It is needed to undertake many operations and find solutions to many natural language processing issues in order to elicit sentiments from some text. Therefore, sentiment analysis as an Information Retrieval field never ceases to be plagued by the unresolved NLPs' issues (Serrano-Guerrero et al., 2015), e.g., negation and sarcasm (Diamantini et al., 2017). In addition, more are inherent/extracted feature (Alqaryouti et al., 2024) and opinion summarizing (Moussa et al., 2018).

Figures 2.4 show the general sentiment analysis procedure. Data is turned into text and handled using NLP techniques after collection and extraction from many sources in different forms. Particularly the processing stage consists of feature selection, feature extraction, and text pre-processing. Different sentiment analysis techniques (e.g., machine learning, deep learning) may be used in the categorisation stage; the result can be shown in many ways. To manage all these stages and do sentiment analysis, many instruments are at hand.

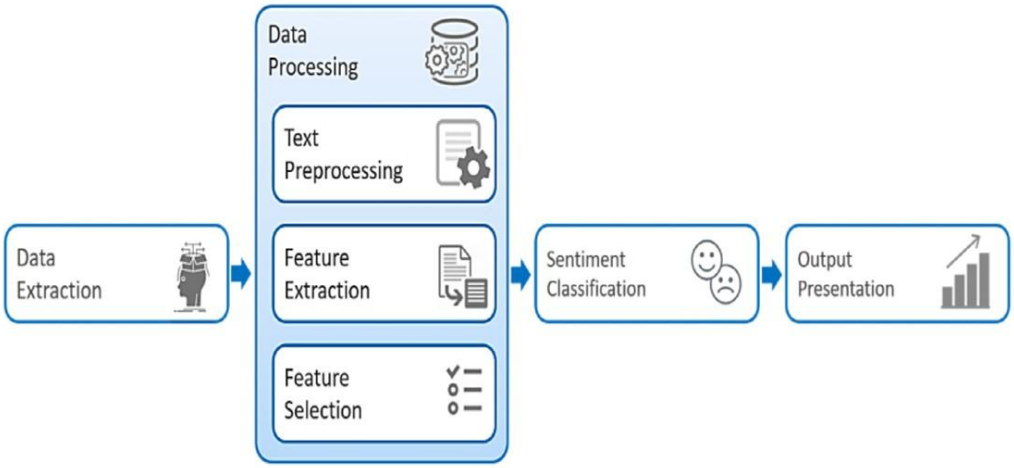

Figure 2.4 Procedures are often used in sentiment analysis (Wankhade et al., 2022)

2.4.1 Data Collection and Extraction

Sentiment analysis starts with textual data; so, many tools, many sources, are accessible to help to get it. Generally speaking, text data may be generated or gathered for study (O'Dea et al., 2015). via third-party (Ranjan et al., 2018) or by web scraping and crawling (Scrivens et al., 2019). Thus, improving textual data with other kinds of data (e.g., telephony data, geolocation data, and video data) to do sentiment analysis might provide fascinating findings.

To get sentiment analysis data, one has many choices. Using public datasets is one of these choices; they may be somewhat cheap, but sometimes it is challenging to obtain pertinent data fit for the study goal. Although various tools exist to produce new datasets with pertinent data (Daly & Huang, 2011), this approach may sometimes be expensive and time-consuming. some of these methods are web scrapping, APIs, and free available datasets.

2.4.2 Data Pre-processing

Usually, the information gathered from multiple sources—mostly through social media—is unstructured. Raw forms like these can have much noise and all kinds of spelling and grammatical mistakes (B. Liu, 2012). Thus, before anything is studied, text has to be cleaned and preprocessed. Apart from enhancing Table of Contents analysis, preparation also aims at decreasing the Table of Contents dimension of input data that includes most words that are meaningless and should be eliminated since they do not affect the text polarity. Table 2.2 shows some of the publicly available tools applied in various Table of Contents preprocessing and NLPs tasks. (Daly & Huang, 2011) compared several Table of Contents NLP toolkits for formal and social media text. The whole procedure is made up of some applied operations that contain tokenization, removing stop words, part-of-Speech (Post) tagging, and lemmatization.

The structure of the supplied data will affect the pre-processing stage. Some formats, for instance, extending abbreviations and removing repeated letters like the "i" in "liiiiiiike," call for more processing and cleaning actions. As was already established, textual data may be rather noisy; consequently, two basic steps—features extraction and features selection—are required to improve sentiment analysis (Alqaryouti et al., 2024).

Table 2.2 Text preprocessing and NLP task available toolset.

Toolkit	Language
NLTK	Python

CoreNLP	**Java**
OpenNLP	**Java**
MADAMIRA	**Java**
TextBlob	**Python**

2.4.3 Feature Extraction

Basic sentiment analysis tasks like feature extraction (FE) or feature engineering might alter sentiment classification outcomes (Liang et al., 2017). In this work, sentiment-expressing words are extracted to characterise important textual features. Social media texts are tricky, therefore Venugopalan and Gupta's analysis may incorporate additional factors (Parashar, 2015). Not always; text is frequently lower case and punctuation is removed. The following are major sentiment study elements:

The easiest way to express features, it is utilised for information retrieval and sentiment analysis. Word presence and frequency: the simplest feature representation. The frequency of single words or a list of n consecutive words in a unigram, bigram, or trigram is computed as a characteristic. Each word has a binary value of zero or one depending on term occurrence. Term frequency is an integer measurement that indicates how often a word occurs in the text. The phrase's textual relevance may be measured using TF–IDF weighting.

Parts-of-Speech tags or PoS tags refer to tags or annotations that tag the part of speech of a word in the target language. Generally speaking, words will be classified as belonging to some of a whole range of parts of speech classes, which include nouns, verbs, articles, adjectives, prepositions, pronouns, adverbs, conjunctions, and interjections. To illustrate, sentence "This camera is good" will be of type Stanford Log-linear sentence. Descriptor of Parts of Speech (Alqaryouti et al., 2024): The camera (noun NN), which is (determiner DT), is (verb VBZ), and is (adjective JJ) nice. Because adjectives are such good indicators of opinion, many methods for sentiment analysis depend on them (Hatzivassiloglou & Wiebe, 2000).

Words and phrases that convey opinions: words that are often used to communicate positive or negative sentiments (for example, nice and amazing for positive emotion, and horrible and dreadful for negative sentiment) [105] are examples of opinion terms. Not only are there a great number of adjectives, but there are also nouns, verbs, frequent phrases, and idioms that are capable of expressing ideas and feelings without the need of opinion words (Alqaryouti et al., 2024).

Negation words, also known as shifters opinion or shifters valence (Polanyi & Zaenen, 2005), are words that have the potential to shift or modify the orientations of opinions and reverse the polarity of sentiments. For instance, the most frequent negatives are not, never, none, nobody, nowhere, neither, and cannot (Polanyi & Zaenen, 2005). Other positives include neither and neither. Nevertheless, these terms are often included in stop-word lists and eliminated from the examination of the text during the preparation stage throughout the process. It is important to handle negation words with caution because of the influence they have, and it is also important to note that not every instance of negated words results in negation.

In most cases, the data that will be utilised for sentiment analysis involves the use of text form. Therefore, the text input should be converted into a fixed-size feature vector that will be appropriate for classification techniques. Reply to this message The bag-of-words model (BoW) and the vector space model (VSM) are the foundation upon which the sentence is constructed. In text representation algorithms, a collection of keywords is typically utilized. The feature extraction calculates over the list of input

keywords to try and figure out the word weights in a piece of text. It then generates a numerical vector that represents the feature vector of a text. There are some common algorithms of text representation discussed in the following paragraphs.

2.5 Text Preparation

2.5.1 Bag of words (BoW)

Among the most basic and often used techniques for text to numerical representation (vector) (Birjali et al., 2021), is the BoW model. But it does not consider syntactic information of the text as it just considers if there is a word that is applicable or not (Chamola et al., 2018) without being concerned about grammatical structure, sentence structure, or syntactic order of words. For instance, looking at the following lines:

S1: ‘‘The camera of this phone is awesome’’.

S2: ‘‘I want this phone; it is all about the camera. I love it’’.

First, the BoW model creates a vocabulary (V in Table 2-3) of all the unique words in the document and then approximates any given sentence (S1 and S2 in the case above) to a vector of fixed dimensionality equal to the vocabulary size of known terms, where each cell in an index contains the frequency of each word in the training corpus. TF–IDF, the second BoW extension, is simple and effective.

Table 2.3 Example of BoW representation of a sentence.

V	The	camera	of	this	phone	is	awesome	I	want	It	all	About	Love
S1	1	1	1	1	1	1	1	0	0	0	0	0	0

S2	0	1	0	1	1	1	0	2	1	2	1	1	1

2.6 Sentiment analysis techniques

A vivid and attractive field of research, sentiment analysis has applications in a variety of areas. Scholars hence always recommend, compare, and analyse multiple ways for that end. The intent is to maximize the performance of sentiment analysis and seeks answers to this arena of problems. Furthermore, adapting sentiment analysis into new areas is extremely beneficial and highlights this work further. However, the choice of the right way to sentiment analysis is really critical. This section is to present a brief introduction of the most common method practice conduct sentiment analysis as shown in figure 2.5. Different viewpoints (e.g., some view of text, level of depth of text analysis) allow one to classify the recent approaches to sentiment analysis (Collomb et al., n.d.).

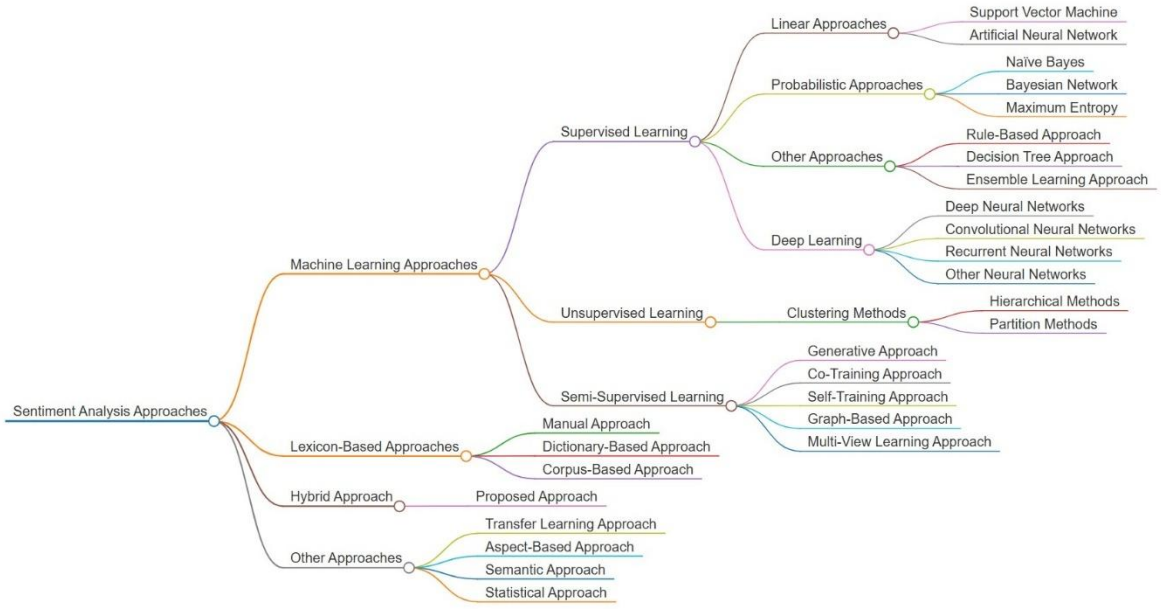

Figure 2.5 Sentiment analysis approaches

Most research, however, often separates the analysis of the sentiment techniques into 3 categories: hybrid methods, Lexicon-Based methods, and machine

learning methods (Madhoushi et al., 2015). Most often utilised method is machine learning. Sentiment classification is accomplished using language characteristics and machine learning techniques. The lexicon-based method makes use of sentiment lexicon, a set of frequently used words and phrases meant to convey either good or negative emotions (Jurek et al., 2015) Conversely, hybrid methods integrate lexicon-based techniques with machine learning to improve sentiment analysis accuracy.

2.6.1 Machine Learning Approaches

According to (Yusof et al., 2015), machine learning categorises sentiment polarity as negative, positive, or neutral from test and train sets. (Y. Li et al., 2017), (Hussain & Cambria, 2018), and (Alqaryouti,2024) distinguish supervised, unsupervised, semi-supervised, and reinforcement learning. Supervised categorisation is used for fixed-class jobs. The unsupervised strategy may be best when labelled data is unavailable to determine this set. Semi-supervised learning may be used for unlabelled datasets with a specific number of labelled samples. Reinforcement learning algorithms employ trial-and-error methods to help agents interact with their surroundings and maximise cumulative rewards.

The issue with machine learning approaches is that they have a tendency to need huge training sets in an attempt to get acceptable performance. This is because such systems are able to learn patterns in the text that are domain-specific, which ultimately leads to improved classification results. On the other hand, a classifier that has been trained on one particular dataset does not classify as well as it would on another domain (Islam et al., 2023).

2.6.1.1 Supervised Learning

For supervised methods, the training files must be tagged with the tags biased towards defining the classes (e.g., positive, neutral, and negative sentiments) as shown in figure 2.6. Linear, probabilistic, rule-based, and decision tree classification methods

are the four types of supervised classification methods (Yusof et al., 2015). Brief review and comparison of the most prominent supervised classification methods usually applied in sentiment analysis are presented in the next subsection.

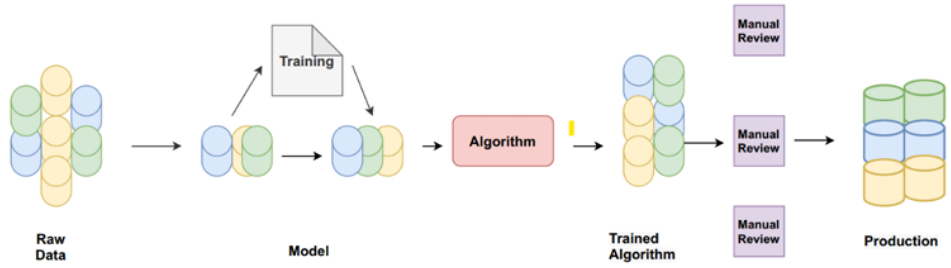

Figure 2-6 Sentiment procedure for supervised machine learning (Wankhade et al., 2022)

There is a statistical method known as the linear approach, which is used to categorise feelings by using decision boundaries that are either linear or hyperplane (Joulin et al., n.d.). When there are more than two classes, the word "hyperplane" is almost always used to describe the situation. This classification is carried out via a linear predictor, denoted by the equation 1 below

$$p = A.X + b \tag{2.1}$$

which makes use of the characteristics of the document which lead to determine the category to which it belongs. X and A are the vectors that represent the linear coefficients (weights) and the document frequency of the words, respectively. A is the vector of linear coefficients. The dot product between A and X, in addition to the bias b, is also represented by the forecasts. When the appropriate features are used, linear classifiers, which are often referred to as deterministic classifiers, are not only straightforward but also frequently achieve state-of-the-art performance characteristics. The benefits and drawbacks of SVM and ANN are outlined in the table 2.4, which provides a summary as well.

The Support Vector Machine, or SVM, is a non-probabilistic classifier which can be used to discriminate data linearly or non-linearly and can also deal with discrete and continuous variables.

It's characterized by its solid theoretical backing and the ability to make more precise classifications than most of the other algorithms in a great variety of applications (Cortes et al., 1995) In (Joachims, n.d.), it was stated that support vector machines are appropriate for text classification and therefore they find extensive application in sentiment classification. To find the hyperplane that maximally separates classes is the aim of the support vector machine (SVM) classifier. As a larger margin decreases the generalisation error of the classifier, a good separation is then defined as the hyperplane with the largest margin to the closest training point to one of the two classes. Support vector machines (SVM) were utilized for performing sentiment analysis in some research works. (Rana & Singh, 2017) used linear SVM and Naïve Bayes for movie review analysis. They found that the linear support vector machine (SVM) technique was most accurate. The research of (Al Amrani et al., 2018) indicates that support vector machines (SVM) with other techniques give promising results. They suggested a hybrid approach using the Support Vector Machine (SVM) and the Random Forest approach. They demonstrated through their study that the hybrid approach is superior to the result of standalone usage of other algorithms.

ANN, which stands for artificial neural networks, have been a popular categorisation method in recent years (Vinodhini & Chandrasekaran, 2016). It does this by extracting features from linear combinations of input data and modelling the output as a nonlinear function of these characteristics (Rana & Singh, 2017). Normal neural networks have 3 layers: input, output, and hidden. Each layer has several organised neurones. Neuronal networks link layers. Each connection's weight is computed by minimising a global error function during gradient descent training (Al Amrani et al., 2018). This determines the actual weight. Chen et al. proposed a neural network-based solution that blends machine learning with information retrieval. They provide semantic orientation indexes to a back-propagation artificial neural network. They found that the technique increased sentiment classification and reduced training

time. Many studies have examined multi-hidden-layer artificial neural networks (ANN).

Table 2.4 Advantages and disadvantages of SVM and ANN.

Classifier	Advantages	Disadvantages
SVM	• High-dimensionally stable and effective • High accuracy and simple training compared to other machine learning techniques. • Efficient memory use owing to kernel mapping in high-dimensional feature spaces.	• Low performance when feature count exceeds sample count. • Select the right kernel function. • Poor interpretability due to no probabilistic categorisation explanation.
ANN	• Proficient in handling complicated variable relationships and generalising well in noisy data. • Works well with high dimensionality issues. • Quick execution time.	• theoretically complicated and difficult to apply. • Demands high usage of RAM. • Training time is longer than other algorithms and some demand a huge dataset.

Probabilistic method Probabilistic classifiers use Bayes' theorem to predict a probability distribution across a collection of classes, whereas linear classifiers predict the most likely class of input (positive or negative). Mix models allow the classifier to classify each class as a mixture component. They are termed generative classifiers because every mixture component is a generative model. Probabilistic classifiers are popular because they are simple, computationally efficient, and need little training

data. If the data do not (at least nearly) follow the distribution assumptions (Birjali et al., 2021), class performance might be poor. Table 2.5 summarises the pros and cons of Maximum Entropy, Bayesian Network, and Naïve Bayes.

In the realm of text categorisation, naïve bayes (NB) is a straightforward classifier among the most often employed methods. Depend on BoW feature extraction, the model is built on Bayes Theorem. Consequently, the location of a word in the text is disregarded and the existence of a given word is independent of the existence of any other words. By using Bayes' rule, naïve Bayes assigns a document d to the category c, hence optimising P(c/d).

$$P(c|d) = \frac{P(c)p(d|c)}{p(d)} \tag{2.2}$$

The previous probabilities of document d being categorised under category c, category c, and document d in p(d|c). Naïve Bayes calculates class posterior probabilities based on document word distribution and independent feature requirements:

$$P(c|d) = \frac{P(c)p(\omega_1|c) * \ldots\ldots * P(\omega_n|c)}{p(d)} \tag{2.3}$$

Naïve Bayes is a classifier used in the majority of research. Hasan et al. attain high accuracy by developing a classifier based on the Naïve Bayes method to categorize attitudes in English and Bangla languages. Based on their classifiers, they also tested certain random tweets and reviews and attained outstanding results in most cases.

Bayesian Network (BN) is a directed acyclic network with random variables at each node and edges showing their effect (Gutiérrez et al., 2019). Although the model recognises that certain nodes are totally reliant due to conditional dependencies, it also recognises that all nodes are dependent since they are random. It is a framework to represent joint probability distribution interaction among variables and is extensible to

include new variables. Bayesian networks seek trillions of word dependency in text categorisation. (Wan & Gao,2016) immediately employed Bayesian nets as senti-classifiers. Naive Bayes, SVM, Bayesian Network, C4.5, Decision Tree, and Random Forest were employed for ensemble sentiment classification. According to their study, Bayesian Network outperformed all six classifiers in one test.

Maximum Entropy (ME) may also be referred to as a Maxent classifier or a conditional exponential classifier; it makes no assumption of interaction between features. According to the following exponential function, it estimates the class label c conditional distribution over document d in an attempt to maximize the system entropy (Zou et al., 2016).

$$P_{ME}(c|d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_i, cf_{i,c}\,(d,c)\right) \tag{2.4}$$

Using Z(d) as a normalisation function, fi, c as a feature function, and λi, c as a feature weight parameter ensures that observed features match projected features in the collection.

$$f_{i,c}(d,c') = \begin{cases} 1, & n_i(d) > 0 \; and \; c' = c \\ 0, & otherwise \end{cases} \tag{2.5}$$

Table 2.5 Advantages and disadvantages of NB, BN, and ME.

Classifier	Advantages	Disadvantages
NB	• Easy to use and interpret. • Computationally efficient.	• Usually, characteristics are not independent. • Data scarcity limits since every value

	• Uses less data and training time than other approaches.	requires a probability estimate.
BN	• Even in complicated domains, it is straightforward to grasp, therefore model construction takes less time. • Handles missing data and achieves high accuracy with minimal training. • Avoids overfitting	• Computationally costly, hence it's not generally used. • Unsuitable for multi-feature issues
ME	• Fits unknown preceding distributions. • Efficiency of textual data acquisition and big data handling.	• Tends to overfit

Rule-based methodology. Any classification process using IF-THEN rules in class prediction is also referred to as rule-based classification (Tung, 2009). Through this method, the classifiers used in such a process are reliant on a guideline set to perform sentiment categorisation. LHS → RHS in which LHS is a rule precursor or a set of requisites over feature space in DNF (Disjunctive Normal Form) and RHS is an outcome or conclusion (class label) of the rule provided LHS conditions are fulfilled (Tung, 2009). Efficient rule-based classifiers, which are utilized in order to categorize fresh occurrences quickly, compare in functionality to decision trees. Rule-based is also helpful in avoiding overfitting. But when the size of the rules becomes too large, it is hard to interpret and is very time-consuming. Its performance is also impacted with noisy data.

Using a rule-based method aided with past polarity lexicon, (Tan et al., 2016) categorised financial news items. They followed sentiment composition guidelines to ascertain the polarity of a phrase; to get the general sentiment of an article they employed a mathematical method known as P/N ratio to average the sentiment values of all the included sentences in the financial news item. Using a rule-based method, (Gao et al., 2015) investigated the emotional causes in a Chinese micro-blog. If a set of criteria are satisfied to identify the related cause, their system sets off feelings depending on an emotion model.

Decision tree approach. The approach makes use of application of a condition upon the at tribute value in an effort to hierarchically divide training data space thereby classifying input data into a finite set of pre-defined categories. Status of attribute value is one or more words' absence or presence (Medhat et al., 2014). Every test on a characteristic of this based tree method is an internal node; every test result is a branch; and child node or class distribution is a leaf node (Birjali et al., 2021). Similar to flowcharts in nature, this based tree method is readable. Also able to handle noisy data, decision tree classifiers are interpretable and comprehensible classifiers. But unstable and overfitting-vulnerable they are (Birjali et al., 2021). Decision tree method works exceptionally well with large datasets; hence it is not advisable to use it with small datasets.

To undertake document-level sentiment categorisation, (Ngoc et al., 2019) developed a novel model based on the C4.5 algorithms of the decision tree, thereby falling within the domain of data mining. On the testing set, the suggested model registered with 60.3% correctness. Sentiment may also be categorised as in using many additional decisions tree algorithms as CART, C5.0, C4.0 (Basti et al., 2015).

An ensemble learning method. This method mostly aims to integrate numerous distinct classifiers to get a classifier that surpasses every one of them (Rokach, 2010). People use this idea when they have to make a significant choice; they evaluate many points of view. This method uses the all-used classifiers to guide decisions by means of their advantages. Generally speaking, the final choice is reached using a set of guidelines including (Wan & Gao, 2016) Majority Vote approach. The fundamental

drawback of an ensemble system is it needs more computation and training time than a single algorithm; yet, the classifiers' cooperation helps to provide a greater generalisation and excellent accuracy. Thus, it is advisable to choose algorithms deliberately (e.g., fast algorithms like decision trees). (Ankit & Saleena, 2018) presented an ensemble classification system using logistic regression, Random for Est, Support Vector Machine, and Naïve Bayes algorithms. Several research (Wan & Gao, 2016) (Sultana & Islam, 2019) used ensemble-based sentiment categorisation technique.

Another intriguing sequential ensemble approach is boosting. Weak classifiers are trained to enhance prediction performance. Each classifier learns from samples that its predecessors mislabelled (Ankit & Saleena, 2018). This approach has several advantages: Lastly, a team of classifiers learns to precisely predict all data. (2) The bad performance of one classifier is inconsequential since other team members will fix it. Many boosting models include AdaBoost, GBM, and Boosted SVM. This sentiment analysis approach has been utilised in many studies (Khalid et al., 2020; Araque, 2017). GBSVM, a voting classifier combining SVM with gradient boosting, was proposed by Khalid et al. (2020). Boosting strategies outperform state-of-the-art models on different datasets.

2.6.1.2 Unsupervised learning

Most of the current methods for sentiment analysis are supervised learning models that have been trained on labelled corpora, such as that each document was previously labelled prior to the training process (Han et al., 2020). But in others, it is hard to collect and aggregate labelled datasets (Mohbey et al., 2022), particularly textual data ones, since the latter most of the time exists in unstructured form. This is due to the fact that they entail a large process of humans labeling the data employed in their development (Fernández-Gavilanes et al., 2016). In return, it becomes simpler to procure unlabeled sets of data and then proceed to classify the datasets with the aid of unsupervised learning algorithms. For this purpose, the techniques use statistical properties of texts such as co-occurrence of words, natural language processing, and

prevailing lexicons comprised of emotionally driven or polar words (Sankar & Subramaniyaswamy, 2017). On the other hand, in machine learning, unsupervised sentiment analysis will tend to use clustering. Data may be divided into numerous groups using clustering without the requirement for each group to be precisely labeled with the sentiment it is supposed to have. That is, the clustering operation separates the data into groups, or clusters, so that the data of a cluster are highly similar from one viewpoint with respect to the data of any other cluster. (Ma et al., 2017) studied the effectiveness of some well-known clustering algorithms with respect to the goal of sentiment analysis. According to (Ma et al., 2017) techniques used for clustering can be categorized into two: partition clustering and hierarchical clustering.

Hierarchical techniques decompose a dataset into nested clusters (sub-groups with groupings) like a tree. Agglomerative and divisive clustering categorize hierarchical techniques (Suresh & Gladston Raj, 2017). Top-down technique determines divided clustering. Starting with one cluster containing all data, this method assigns them to sub-clusters based on similarity using a recursive process. This method was used by (Tsagkalidou et al.2011) to cluster blog entries regarding emotions by proximity. In the bottom-up technique, or agglomerative clustering, each data point starts in its own cluster and merges with related data until one or more clusters remain. (Archambault et al.2013) analyzed microblogging topics and sentiment using agglomerative clustering.

Partition methods try to divide data into a collection of non-overlapping sets such that every element is present in one (Cui et al., n.d.). This is based on a measure of similarity usually Euclidean distances between the objects. Even though with maximum distance from other clusters' data, a cluster's data possesses a fairly short distance to each other. Table 2.6 depicts frequently advantages and disadvantages of the clustering methods used in sentiment analysis.

K-means and its variations are the most used partition algorithms (Al-Harbi & Rayward-Smith, 2006). The K-means method assigns each database sample to a cluster based on the similarity of the data item and cluster centers from a specified starting point. This continues until all data items are mapped to cluster centroids.

Algorithms stop when convergence criteria are met. Either an iteration count is used or the result does not change. K-means clustering was preferred for huge data sentiment analysis (Riaz et al., 2019).

Table 2.6 Advantages and disadvantages of clustering approaches.

Method	Advantages	Disadvantages
Hierarchical	• Simple to apply. • Handle noisy data effectively. • No need to set cluster numbers.	• High compute costs make it unsuitable for huge datasets. • High outlier sensitivity. • Once allocated, an item cannot be moved to another cluster.
Partition	• Perfect for sentiment analysis. • Simple, scalable. • Low computational needs make it suitable for huge datasets.	• Data noise sensitivity. • Handling non-convex clusters and identifying initial cluster centroids. • Inaccurate and unstable.

2.6.1.3 Semi – Supervised Learning

Generative approach. This method supposes that data in many categories follow various distributions and that, in case at least one labelled data per category (Han et al., 2020), the parameters of every distribution may be approximated. Stated differently, a generative model specifies distributions on the inputs and uses Bayes rule to project, after training this model, the label (class) of a test input. Combining tree complementary and theoretically baseline models, (Mesnil et al., 2014) suggested

a rather basic and effective ensemble approach for sentiment analysis. Among these was established a generative method based on IMDB movie review dataset 4 the whole system achieves a new state of the art performance. This shows that one may use ensemble learning while using unsupervised or semi-supervised methods. Table 2.7 Advantages and disadvantages of semi-supervised approaches

The co-training approach. The approach, developed by (Blum ,1998), assumes data may be presented in two perspectives with knowledge of all data points (Biyani et al., 2013). Teaching each other will be based on two classifiers' shared knowledge. Every classifier for the two data perspectives is trained with two feature sets. Co-training adds the most confident examples of each classifier to the labelled set in each training cycle, expanding the dataset. After consuming all unlabeled data or a set number of iterations, the iteration stops (Da Silva et al., 2016). This sentiment analysis method has been utilized in many studies. (Xia et al. n.d.) proposed a bi-view co-training technique using Blum and Mitchell's method to solve the negation issue and improve semi-supervised sentiment classification bootstrapping.

self-training approach. Semi-supervised learning uses it extensively. Self-training is two-step. First, the classifier is trained on little labelled data. In the second stage, a trained classifier labels unlabeled data to add the most confident examples to the training set (Gao et al., 2014). Using the newly tagged data, the last step is repeated. The induced model is evaluated using test data. This approach is widely used in sentiment analysis (Gao et al., 2014). (He & Zhou, 2011) suggested a framework learning from named characteristics rather than labelled examples after self-training. The experimental findings showed that their strategy improved certain previous methods.

Graph based method. Under this method, an architectural. The data is shown using a graph. Vertices in the graph show in-stances—that is, sentences—while edges explain the consistency between events. Usually, the tightly related events fall into the same class (Hajmohammadi et al., 2015). numerous research using this method demonstrated its efficiency in numerous NLP tasks including sentiment analysis (Hajmohammadi et al., 2015) Using graph-based Word Sense Disambiguation (WSD)

and a multiple meaning sentiment lexicon, (Jalilvand & Salim, 2012) presented a sense level sentiment categorization algorithm. Using a baseline technique, they evaluated their strategy against another usually falling into the same class (Hajmohammadi et al., 2015). This technique has been widely used in several papers, demonstrating its efficacy in various NLP tasks, including sentiment analysis (Hajmohammadi et al., 2015). Inspired by graph-based Word Sense Disambiguation (WSD) and a multiple meaning sentiment lexicon, (Jalilvand & Salim, 2012) proposed a sense level sentiment categorization algorithm. Using two subjective lexicons, they evaluated their methodology against a baseline method to show its efficacy for sentiment categorization. two subjectivity lexicons to demonstrate their success for sentiment categorization.

Multi-view learning method, in this technique, multiple views are considered in order to find the solution to the problem and the overall performance is derived using them agreeing on each other (Su et al., 2012). All the classifiers will be learned on one view; afterwards, these classifiers are used to assign labels to unlabeled instances, which will then be included into the training set if they happen to be extremely reliable. Principally, the approach is practiced on instances where there are plenty of feature sets. In congruence with the concepts in multi-view learning, (Lazarova & Koychev, 2015) suggested an approach to sentiment analysis of movie critiques in terms of Bulgarian.

Table 2.7 Advantages and disadvantages of semi-supervised approaches.

Approach	Advantages	Disadvantages
Generative	• Few marked examples provide great accuracy. • Effective if the model is almost accurate.	• In-flexibility • Performs poorly in categorisation tasks.
Co-training	• Performs well with few marked occurrences.	• Unsuitable for datasets with single features.

	• Reduces error spread. • Works with most popular classifiers.	• Sensitive to outliers and noise data.
Self-training	• Simple • Suitable for large tagged datasets. • Applicable to most common classifiers.	• Spread of wrong results. • Does not provide sufficient convergence data.
Graph-based	• Performance is good when graphs match tasks. • Simple to understand	• Performance depends on graph structure and edge weights. • Poor performance if graphs don't match tasks.
Multi-view learning	• Explores the issue from several perspectives.	• Considers feature conditional independence.

2.6.1.4 Reinforcement learning

In reinforcement learning (RL), an agent gets rewards in the following time step based on its prior action. The agent interacts with the environment to maximize cumulative rewards using trial-and-error RL algorithms (Y. Li et al., 2020). Reinforcement learning, usually used in games, is being used for robot control. This method can handle challenging tasks, especially with neural networks, however sentiment analysis is seldom used. This method's similarity to human learning is useful for sentiment analysis. Reinforcement learning corrects training faults by using previous experiences to improve judgement and approach perfection. However,

defining the reinforcement learning model may be difficult. Reinforcement learning is data-intensive and computationally expensive.

(W. Liu et al., 2018) proposed a reinforcement online learning approach for real-time emotion state prediction using physiological inputs. The authors substituted application of the incentive concept with. The predictor. In every round of the online learning, the. The efficiency of their proposed approach was compared to support vector regression (SVR) and least squares (LS) models. Experiment outcomes show the excellent performance of the suggested approach and the significant reduction in time. As in (Broekens et al., 2015), emotions can be used as a motivation to maximize agent action. Starting from the reinforcement learning primitives—i.e., reward—the authors suggested a computational model of four emotions: pleasure, sorrow, hope, and fear. In a design as mapping of affect labels onto RL primitives, the model is instrumented to probe the mapping of adaptive behavior to emotion. Agent-based simulation experiment demonstrates that there can be a mapping of emotions onto reinforcement learning so that there would be a signal for feedback for the agent to change its behavior that will maximize the communication among adaptive agents and human agents.

2.6.1.5 Deep Learning

Recent sentiment analysis has also favored ANN-based deep learning (DL). DL introduces supervised or unsupervised feature representment learning methods based on machine learning (Rojas-Barahona, 2016). Deep learning for reference neural networks with many layers of perceptron inspired by the human brain (Vateekul & Koomsubha, 2016) allows more complex models to be trained on a larger dataset and produce state-of-the-art results in many fields, from computer vision and speech recognition to NLP. CNN, RNN, and DBN are DL neural network models. They can learn sophisticated traits from data (Moubayed et al., 2020), but engineers don't need to choose them. Conversely, they are computationally expensive and complicated. Several research examined deep learning sentiment analysis approaches (Dang et al., 2020a). However, the next sections briefly summarize the most popular deep learning models for sentiment analysis.

Deep networks (DNN). This model has hidden layers in its Artificial Neural Network (ANN) between the input and output layers (Schmidhuber, 2015). The input layer contains input data, the hidden layer contains neurons, and the output layer has one or more neurons that create network outputs (Dang et al., 2020b). It uses complex math modelling and ANN learning to determine the input-output connection. ANN feedforward and partial DNN flows may be separated. Feedforward ANNs are basic, perfect networks. Most natural language processing applications like sentiment analysis employ DNN and its variants (e.g., CNN and RNN). BowTie, based on the deep feedforward neural network, has one encoding layer, hidden levels, and an output layer (Vassilev, 2019). Model analysis yields good results compared to other methods.

CNN—convolutional neural networks. This variant feedforward neural network, first utilized in computer vision (Ouyang et al., 2015), has shown success in NLPs and recommender systems. An input layer, output layer, and hidden layer with convolution, pooling, normalization, and fully linked layers comprise a CNN. Convolutional layers filter inputs (e.g., word embedding in text sentiment analysis) to create features, whereas pooling layers lower feature resolution to make feature identification noisy and small change independent. Fully connected classification layers and a normalizing layer to normalize a preceding layer's output improve convergence during training. CNNs have become popular for sentiment analysis. According to (Kim, 2014), one of the most popular CNN models for sentiment analysis is sentence-level sentiment categorization. The author evaluated a CNN model based on pre-trained word2vec. The model outperformed earlier techniques, suggesting pre-trained word-embedding may be useful for deep learning NLP applications.

Recurrent neural networks. Memory cells are used to address input sequences. RNNs are widely employed in NLP applications like sentiment analysis because they can store and recall extended sequences (Sharfuddin et al., 2018). RNN output relies on prior computations. For instance, the model predicts the next word in a phrase using all prior word states and their relationships (Chen et al., 2019). Standard RNN has a problem with vanishing gradient, hence (Hochreiter & Schmidhuber, 1997) invented Long-Short Term Memory (LSTM), which is used in many industries. This

paradigm is being used for sentiment categorization by academics. A bidirectional LSTM model (W. Li et al., 2020) may exploit the revelation between target words and sentiment polarity keywords in a sentence without a sentiment lexicon. Experimentally, our model outperformed numerous state-of-the-art methods.

alternative neural networks, Fewer deep neural networks are utilized in sentiment analysis than the three above. (C. Li et al., 2014) introduced an RNN-based Recursive Neural Deep Model. This model outperforms Naïve Bayes, Maximum Entropy, and SVM in Chinese social data binary sentiment categorization accuracy. Unsupervised deep neural networks like Autoencoders and its derivatives for sentiment analysis are difficult to employ (Sagha et al., 2017). Classifiers should collect features from the encoder layer (Zhou et al., 2014). hybrid models combine two or more deep learning models (Rehman et al. 2019). The authors present a Hybrid CNN-LSTM Model for sentiment categorization utilizing LSTM and an extremely deep CNN model.

2.6.1.6 Lexicon – Based Approach

In this scenario, sentiment analysis uses a predefined dictionary of words, opinion lexicon, to score words as negative or positive (Hu & Liu, 2004). Scores may be a polarity value (+1, -1, or 0) for positive, negative, or neutral words, or a number indicating sensation strength or power. Finding the semantic orientation values of incoming text terms helps determine its direction. Each lexicon element is allocated sentiment values after tokenizing a document into words or microphases. Sum and average or other methods may determine the document's formula, algorithm, or attitude.

Lexicon-based sentiment analysis excels in phrase and feature sentiment. Unsupervised because no training data is needed. A nice phrase in one domain may not be in another since words have hundreds of meanings and senses. In the first sentence, "small" is undesirable because most people like large screens, but in the second, it's beneficial since smaller cameras are easy to carry. Create or modify a

domain-specific sentiment lexicon to prevent this issue. (Sanagar & Gupta, 2020) Genre-level sentiment lexicon adaption. This novel adaption technique learns target and source domain sentiment lexicons from unlabeled data. Domain-specific lexicons may be learnt by transfer learning (Sanagar & Gupta, 2020). A revolutionary unsupervised sentiment lexicon learning approach may be used in new genres. Genre-level information from source domain corpora is transferred to destination domains. Compared to machine learning with huge training sets, lexicon-based approaches perform badly. García-Hernández et al. (2013) outline three methods for developing and annotating sentiment lexicons.

In order to label the lexicon, hand method requires the intervention of people. Two steps characterize the emergence of sentiment lexicons: to produce the list of words having sentiment carrying firstly; to secondly assign labels of sentiment onto these words. Generally, a little time-consuming, costly, and labour-intensive in nature, hand method can help deliver a homogenous and stable vocabulary. The process might find one automated option associated with streamlining this procedure. Here, to avoid the errors or as a benchmarking activity, a manual approach is taken. Several lexicons were created hand-made. Based on a hand list of intensifiers and negators, (Wilson et al., 2005) constructed MPQA Subjectivity Lexicon; (Taboada et al., 2011) constructed Semantic Orientation CALculator (SO-CAL).

(Mohammad & Turney, 2013) Gamification and crowdsourcing provide researchers another tool. On the Internet, crowdsourcing is the method of involving a community towards a shared objective. Using Amazon Mechanical (Mohammad & Turney, 2013) developed word emotion and word polarity association lexicon. Rather, gamification is the use of game techniques to non-game challenges. Tower of Babel was created (Hong et al., 2013) to include players in assigning emotion polarity to words thereby creating a sentiment lexicon.

Dictionary-based technique Unlike antonyms, synonymous phrases share the same emotional polarity, according to this technique. This technique develops emotion lexicons using trusted dictionaries like WordNet9 (Miller et al., 1990) or thesauri (S. Mohammad et al., 2009). First, beginning seed words with known

orientation are hand-collected. Using many lexical resources to find synonyms and antonyms expands the list. The latest words are added to the previous list until no more are found (Yusof et al., 2015). Later, a manual examination may find and rectify errors. Using automatic annotation of all WordNet 3.0 synsets, (Baccianella et al.2010) created SentiWordNet 3.0, a popular lexicon. (Song, 2016) suggested building a thesaurus lexicon from three online dictionaries. Some available lexicons for seed growth are included in Table 2.8. (Sanagar & Gupta, 2016) discussed polarity lexicon learning. Authors addressed the polarity terminology in two ways. The first polarity lexicon building approaches were provided. Second, the authors insightfully analysed the open-source polarity lexicon. The endeavour concluded with open research questions and polarity lexicon building directions.

All dictionary-based Lexicon-based approaches cannot detect domain-specific sentiment terms, making them unsuitable for context and domain-specific categorisation. Creating dependence rules is difficult and time-consuming, but this method is computationally cheap without dataset training and can rapidly construct a lexicon with numerous emotion terms and their orientation.

Table 2.8 List of common lexicons.

Lexicon Name	Description	Lexicon size	Output
WordNet (Miller et al., 1990)	a database of English words that groups verbs and nouns by their semantic and lexical meanings.	117000 synsets	Synsets of words that mean the same thing.
SentiWordNet (Esuli & Sebastiani, n.d.)	Opinion mining using WordNet. Synsets are sentence components (e.g., nouns, verbs) with polarity ratings and same meaning.	117000 words	Positive, negative, and objective polarity scores between 0.0 and 1.0..

SenticNet (Cambria et al., 2020)	conceptual primitive-based SA semantic resource. The semantic polarity of common-sense concepts is ended by dimensionality reduction.	200000 concepts	Negative, Positive
MPQA (Wilson et al., 2005)	The University of Pittsburg's Opinion Finder contextual polarity and subjectivity clues.	20611 words	Negative, Objective, Positive

2.6.1.7 Corpus – Based Approach

Corpus-based techniques use a seed set of pre-tagged orientation sentiment words and syntactic or co-occurrence patterns to find more sentiment words and direction in a large corpus than dictionary-based approaches. Language rules like AND, OR, and BUT help reveal emotion words. The same direction causes two adjectives connected by a conjunction (like "simple AND easy") to drift. Even if consistency of emotion is not always used, these connectives may have laws. After this procedure, clustering may be utilized to produce sets of impact words—positive and negative (Liu, 2011).

The technique outlined in (Hatzivassiloglou & McKeown, 1997) was to list frequently occurring adjectives with their orientations and add words that co-occur in the pattern W1 and W2 share the same orientation. They grouped positive and negative words by labelling whether two consecutive adjectives had the same or opposite polarity after developing a log-linear model network with words in vertices and their pairs in edges. The corpus-based method is straightforward but takes a large dataset to assess word polarity and text mood (Agarwal & Mittal, 2016). Statistical and semantic strategies include corpus-based approaches (Vyas & Uma, 2019).

Statistics-based technique for word emotion orientation. This technique states that identical emotion phrases frequently have the same sentiment if they appear often together. Therefore, a word's unknown polarity is decided by its frequency of co-occurrence with other words in the same context. The (Turney & Littman, 2003) mutual information calculation method determines co-occurrence frequency. This approach is often used to create sentiment lexicons and analyze sentiment. Han et al. (2018) proposed a domain-specific lexicon generation method for review sentiment analysis. They lexiconically allocated PoS tags to words using mutual information. The authors did well with the proposed method.

A semantic view. This ontology-based technique assigns word similarity using novel criteria, unlike earlier approaches. (Araque et al., 2019) found the semantically nearby phrases' emotion value similar. This method is used in emotion dictionaries for synonyms, antonyms, and related phrases to extend vocabulary and analyze sentiment (Zhang et al., 2012). The authors suggested Weakness Finder, a semantic and statistical expert system that finds product faults in Chinese reviews. Word similarity was calculated using the Chinese Hownet lexicon (Dong, 2006). The recommended expert system performed well in experiments.

2.6.1.8 Hybrid Approach

Lexical and machine learning approaches are used in the hybrid method. It eliminates ambiguity by combining sentiment word context with machine learning flexibility and lexical analysis (Gupta & Joshi, 2020). The hybrid technique is driven by machine learning's accuracy and lexicon-based method's stability.

The combination of strategies of both above strategies makes it leverage on their shortcomings and maximize their capabilities. A few hundred lexical solutions are consequently utilized as feed for sentiment classifiers. Hence, emotion lexicons are reasonably imperative in the hybrid strategy traditionally accustomed to the pursuit of extra performance.

Few sentiment analysis models used hybrid methods. Most popular lexicon-based approaches to identify word polarity for sentiment analysis classifiers. An early study by (Devi et al., 2019) used machine learning classifiers, dictionaries, and HARN's method for lexicon-based document categorization. First, they labelled reviews across domains using Naïve Bayes and SVM, then estimated document-level polarity using HARN's approach. In addition, the hybrid technique was 80%–85% accurate compared to HARN's. Deep Learning may be utilized with lexicons for sentiment analysis. (Shin et al., 2016) used lexical embeddings and attention in a convolutional neural network. Word scores were aggregated to create lexical embeddings from several sources. Naïve concatenation, separable, and multichannel convolution incorporate these embeddings in a CNN model. Lexicon integration may increase CNN model accuracy, stability, and efficiency. (Elshakankery & Ahmed, 2019) propose a hybrid tweet polarity classification technique employing lexicon-based and machine learning methods. For Arabic tweet sentiment analysis, the authors recommended HILATSA, their hybrid incremental learning technique. Using SVM, Logistic Regression, and Recurrent Neural Network (RNN) classifiers, they created words, emoticons, idioms, and general lexicons.

(Shin et al., 2016) utilized an attention mechanism and lexical embeddings in a convolutional neural network. It generated lexicon embeddings from diverse sources of lexicons based on word score aggregations. Three approaches—Naïve concatenation, separable and multichannel convolution—integrate these embeddings into a CNN model. It illustrates that accuracy, stability, and efficiency in CNN models can possibly be improved with lexicon integration. In (Elshakankery & Ahmed, 2019), they suggest a hybrid solution that leverages both lexicon based and machine learning algorithms for tweets' polarity classification. The authors suggested their hybrid incremental learning method, HILATSA, for Arabic tweets sentiment analysis. They developed terms lexicon, emoticon lexicon, idioms lexicon and various general lexicons based on SVM, Logistic Regression and Recurrent Neural Network (RNN) classifiers for classification.

To treat the different types of words and misspelling, they also experimented with the Levenshtein distance approach to opinion analysis. Six datasets have been

placed under trial of the HILATSA algorithm. Five lexicons are preprocessed for testing, verification, and training the classifier model; the sixth dataset is for hybrid system testing and simulation.

2.7 Related Work

Repeatedly buying products from shoppers on a web shop and then giving ratings and reviews as feedback is an activity that is carried out repeatedly. Data availability and the usage of natural language processing techniques have attracted a vast majority of researchers towards the area of sentiment analysis. It is our intention to perform a sentiment analysis of Amazon.com reviews of cell phones. In this current study, studies that have previously been done on the subject of sentiment analysis and that are connected to Amazon reviews or reviews for phones have received a thorough examination.

In the field of mobile phone reviews, there is a substantial amount of research (1-2), (4-5) that pertains to sentiment analysis. The data that were pertinent were gathered, and this was followed by the commencement of data preparation. Sentiment analysis was done (1 - 5) on initially unlocked mobile phones.

At the beginning of the data preparation stage, (Mukherjee et al., 2021) transformed all reviews into lowercase, removed all punctuation, removed stop words, and lemmatized and tokenized the reviews. Following rating, these data were categorized into three forms: positive, neutral, and negative attitudes. But for training, positive and negative data only were used. The one- and two-stars negative ratings were labeled as negative, three-stars ratings were labeled as neutral, and four stars and five stars ratings were labeled as positive emotion. After the data processing was completed, 30,000 negative and 30,000 positive data were selected. During the course of comparing the performance of machine learning and deep learning-based classifiers, it was observed that the best performance was obtained by the deep learning model with a score of 98.51%, while the Decision Tree model performed 95.1% in machine learning. Tokenization, removal of stop words, lemmatization, lowercasing, and

removal of punctuation were all performed during the pre-processing of (K. Baktha & B. K.Tripathy, 2017). They went through each review and classified it as good, negative, or neutral according to the rating. The four-star and five-star ratings were assigned a positive rating, the three-star ratings were assigned a neutral rating, and the one-star and two-star ratings were assigned a negative rating.

Because the minimum number of evaluations that were neutral was 21000, the researchers opted to utilize this number so as to have even data about the sentiments of various groups.The performance of various machine learning and deep learning models was tested and compared on the basis of a variety of different feature extraction methods, including TF-IDF, bag-of-words, word2vec, and Glove, among many others. In this particular instance, the CNN model performed a decent level of performance with 92.72% accuracy when using the word2vec feature extraction approach. The mobile review dataset has also been subjected to similar nature work. Text data have been converted into numeric data by applying the TF-IDF approach. After this, the model's performance was tested with mobile phone data by utilizing different machine learning algorithms, such as Naive Bayes, RNN, ANN, and SVM. The accuracy of RNN was 95.67% (S.Tammina, 2020).

Once the process of negation marking was completed with it and the effect of negation in sentences was realized, it was at its best. On the other hand, when TF-IDF vectorizer was utilized along with logistic regression, it showed a satisfactory performance, having an accuracy rate of 92% compared to both Naïve Bayes and Random Forest algorithm (Aljuhani & Alghamdi, 2019). For the purpose of conducting a sentiment analysis on three various mobile phone brands' data, i.e., the REDMI Note 3, APPLE IPHONE 5S, and SAMSUNG J7, three machine learning algorithms were developed: Logistic Regression, Naive Bayes, and SentiWordNet model.

Apart from Word2Vec, BoW, and TF-IDF were used at the time of feature extraction by Fang and Zhan (2015). Additionally, MLP classifier, SVM, Random Forest, Naive Bayes, and Decision Tree were also utilized for classification. The

researcher compared its performance and found that the MLP classifier worked well with BoW as a feature extraction method, thus giving 92% accuracy.

A summary of the research paper on sentiment analysis is included in table below 2.9, which provides an overview of the information.

Table 2.9 Tabular Representation of literature review

Reference	**Publisher**	**Papers Type**	**Model Used**	**Datasets**	**Evaluation Criteria for Classification**
1. (Mukherjee et al., 2021)	IEEE	Comparison	• Decision Tree • BERT	Mobile	• Accuracy: 95.1% • Accuracy: 98.51%
2.(S.Tammina , 2020)	ScienceDirect	New Technology	• RNN	Cell Phone	• Accuracy: 95.67%
3.(Aljuhani & Alghamdi, 2019)	Springer	Comparison	• Random Forest • Naïve Bayes • Logistic Regression	Mobile	• Accuracy: 92% • Accuracy: 91% • Accuracy: 93%
4.(Almjawel et al.,2019)	IJACSA	Comparison	• word2vec+CNN	Mobile	• Accuracy: 92.72%
5.(K. Baktha & B. K.Tripathy, 2017)	IEEE	Comparison	• SVM	Mobile phones, accessories, and musical instruments reviews	• Accuracy: 94.02%
6.(Korovkinas & Danėnas, 2017)	IEEE	Comparison	• LTSM • GRU	Health and Personal Care product	• Accuracy: 78.1% • Accuracy: 83.9%
7.(Kumar, Desai, & Majumdar, 2016)	University of Latvia	New Technology	• SVM • NB • New Introduced Method	Product Review	• Accuracy: 78.08% • Accuracy: 84.35% • 89.19

8.(Hu, Hu, Ding, & Zheng, 2015)	IEEE	Comparison	• NB • Logistic Regression	APPLE IPHONE 5S, SAMSUNG J7, REDMI NOTE 3	**NB** • Recall: 0.87 • Precision: 0.675 • F-Measure: 0.76 **Logistic Regression** • Recall: 0.778 • Precision: 0.713 • F-Measure: 0.744
9.(Fang & Zhan, 2015)	Springer	Comparison	• SentiME	Product Reviews	• Recall: 0.905 • Precision: 0.857 • F-Measure: 0.88
10.(Bhatt et al., 2015)	IEEE	Comparison	• DNN • SVM	Electronic products	• Accuracy: 90% • Accuracy: 85%
11.(Bhaskar, Sruthi, & Nedungadi, 2014)	Springer	Comparison	• SVM • LTSM	beauty, book, electronic, and home products	• Accuracy: 80.95% • Accuracy: 87.6%
12.(Sharma & Dey, 2013)	IEEE	New Technology	• Existing Method • Proposed Method	Digital camera	• Accuracy: 74.2% • Accuracy: 76.02%

2.8 Data Visualization and Dashboards in Sentiment Analysis

Visualizing data plays a crucial role in social media sentiment analysis due to the large and diverse nature of the data it produces. Interactive dashboards allow researchers and decision-makers to quickly grasp public sentiment trends, detect emerging topics, and make informed choices. According to Chowdhury et al. (2025), effective visual representation significantly enhances the interpretation of sentiment data and supports the development of more focused communication strategies.

Data visualization serves as a powerful and universal method not only for explaining complex datasets but also for effectively communicating analysis results to stakeholders—particularly those without technical expertise. According to Plecto

(2025), key trends in data visualization for 2025 include the use of real-time dashboards, AI-powered insights, and mobile-optimized interfaces, all aimed at enhancing accessibility and clarity of data.

An effective dashboard should follow fundamental design principles like clarity, consistency, and emphasis on the most important information. As noted by UXPin (2025), an ideal dashboard presents data in a clear and intuitive way, enabling users to easily interpret insights and make well-informed decisions.

Sentiment analysis visualization techniques are especially useful for detecting emotional patterns, such as a rising wave of negative sentiment toward a specific policy or product. These insights are crucial for crafting more targeted and responsive communication strategies. According to IRJMETS (2025), real-time sentiment dashboards support ongoing monitoring of public opinion, allowing timely responses to emerging issues.

2.8.1 Related Studies on the Use of the Dashboard in Sentiments

The use of dashboards to monitor public opinion has become standard practice across various industries, including political campaign management, government project oversight, and public policy polling. WebLyzard (2024) highlights this trend in its report, noting that during the 2024 U.S. presidential election, dashboards were effectively utilized to track shifts in public sentiment by analyzing media interactions and the influence of news stories on how candidates were perceived.

In addition, Chowdhury et al. (2025) emphasize that social media sentiment analysis offers valuable insights into customer behavior and emotional patterns, which can support more informed communication and business strategies. Effectively designed dashboards serve as a powerful means of consolidating and presenting these insights, making them easier for decision-makers to understand and act upon.

IRJMETS (2025) also underscores the value of real-time sentiment analysis dashboards for tracking public opinion and quickly addressing emerging concerns.

These dashboards enable users to observe public sentiment as it unfolds, detect the emergence of new trends, and take timely actions in response to changes in public perception.

Below is a literature review table that summarizes studies on the application of dashboards for monitoring public opinion.

Table 2.10 Literature review related to the use of dashboards

Reference	Area	Mechanism	Insights
WebLyzard (2024)	Public opinion monitoring in elections	Real-time dashboard	Enables media association analysis for political campaign monitoring, though limited to media data only.
Chowdhury et al. (2025)	Consumer sentiment analysis	Interactive dashboard	Provides insights into consumer behaviour for communication strategy but is restricted to specific platforms.
IRJMETS (2025)	Public opinion monitoring on policies	Real-time dashboard	Allows rapid response to emerging issues in government programs, though limited to social media data.

2.9 Sentiment Analysis Challenges

2.9.1 Sarcasm detection

Sarcasm is described by Macmillan English dictionary as saying something nasty to make someone laugh or speaking the reverse of what someone intends. Make him seem angry or nasty (Rundell, 2007). Sarcasm makes sentiment analysis harder since one might write something nice but with a bad motive or something negative but positive. Daily, we hear caustic comments. The necessity to recognise sarcasm to automatically classify sarcastic lines in a text and achieve false sensations is developing. Sarcasm identification is a difficult NLP task due to its intricacy and uncertainty (Ren et al., 2020). There are several ways to detect sarcasm (Ren et al.,

2020). (Jain et al., 2020) performed real-time sarcasm detection using deep learning and Hinglish. Softmax attention layer, convolutional tensional neural network, and bidirectional LSTM were used in their model. CNN received semantic context vector for English features from softmax attention layer from GloVe word representation. Additional punctuation-based HindiSenti (Hindi Senti WordNet) feature vectors enriched the CNN model. Its categorisation rate is 92.71%, surpassing baseline deep learning systems.

2.9.2 Negation handling

Since typically flip text polarity, negative terms like not, neither, nor, etc. are crucial for sentiment analysis. Though "The phone's battery is not good," "This phone's battery is good" is positive. Unfortunately, some systems eliminate negation words since they're on Stop-Word lists or their dictionary's neutral emotion value doesn't alter polarity. Negation phrases are common in sentences without altering mood, making polarity reversal difficult. (Jain et al., 2020) propose a syntactic path-based bidirectional neural hybrid network for negation scope identification. Bi-LSTM is trained to learn context representation from sentence direction (forward and backward), however this study combined CNN and Bi-LSTM to find token-cue syntactic characteristics in dependency and constituency parse trees. Model F-score: 90.82%.

2.9.3 Spam detection

Spam detection is vital in sentiment analysis. Since online postings influence consumer purchases, spam and spam reviews may damage corporate reputation and intentionally modify customers' opinions about goods, services, organisations, and others (Cardoso et al., 2018). Since reviews seem the same, building a spam technique to recognise phoney reviews in a large number of reviews is tough. (Saumya & Singh, 2018) developed a spam-reducing system using review and comment sentiment, content-based characteristics, and rating variance. This method classifies reviews as spam or not based on comment data.

2.10 Evaluation Metrics for Sentiment Analysis.

2.10.1 Accuracy

The most common categorisation metric is accuracy, a ratio of right to incorrect. Relationship between predicted instances and total cases is shown in equation 2.6 (He & Zhou, 2011). Accuracy is also good for machine learning sentiment classification if the data classes are balanced. Accuracy is also great.

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{2.6}$$

2.10.2 Precision

Precision is accuracy, the percentage of anticipated positive samples that are right. Accuracy considers excellent precision. Additionally, the metric indicates challenges with reliable predictions (Han et al., 2018).

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{2.7}$$

2.10.3 Recall

The recall (or sensitivity) is the percentage of positive samples accurately labelled as positive. Recall examines model misclassification. Recall, unlike precision, may be used to forecast depression when class identification should be prevalent. This forecast should be uncertain (Bastı et al., 2015).

$$\text{Recall} = \frac{T_P}{T_P + F_N} \tag{2.8}$$

2.10.4 F-measure

F1-score may help overcome the challenge of comparing classifiers with varying recall and accuracy, as it balances specific predictions with extensive class coverage (Bastı et al., 2015).

$$\mathrm{F-Measure} = \frac{2\,(\mathrm{Precision})(Recall)}{(\mathrm{Precision}) + (Recall)} \tag{2.9}$$

2.10.5 Specificity

The inverse of recall metric is specificity. One may do a confident accuracy using this measure (Bastı et al., 2015).

$$Specificity = \frac{T_N}{T_N + F_P} \tag{2.10}$$

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Model designing, the chapter provides an overview of the overall framework adopted in the research process for carrying out sentiment analysis on Amazon review ratings. Right from the initial question of the sentiment analysis to comparing and testing the models, the research process has been covered. In this chapter, information used and models applied will be explained and demonstrated.

3.2 Research Framework

In an attempt to attain the sentiment analysis as a whole, research was conducted in 8 steps. The accomplishment of each step resulted in the attainment of a very significant milestone. Following are the processes that are provided in the order of their occurrence: research gab and identification, data collection, data preprocessing, exploratory data analysis, feature engineering, machine learning implementation, model comparison, and development of an interactive a Power bi dashboard. There is a diagram of the work process in figure 3.1. There is an explanation of each step in the following sub section below.

3.3 Experimental Setup

The project was done in Python, and the scikit-learn module was used for machine learning tasks like vectorization, feature selection, and classification. Jupyter Notebook has been done for development and testing since it was an interactive space

for scripting, visualization, and documentation. It has been utilized other tools like NumPy and pandas to change the data, and matplotlib and seaborn to do statistical analysis and make graphs. Table below shows the computational environment setup.

Table 3.1 computational environment setup.

Component	**Specification**
Operating System	Windows 11
Processor	Intel® Core™ i9-14900HX
RAM	16 GB DDR5
GPU	NVIDIA GeForce RTX 5070
Specifications	
Language	Python v3.11.5
Browser	Microsoft Edge
Environment	Jupyter Notebook
Python Packages	pandas, numpy, matplotlib, scikit-learn

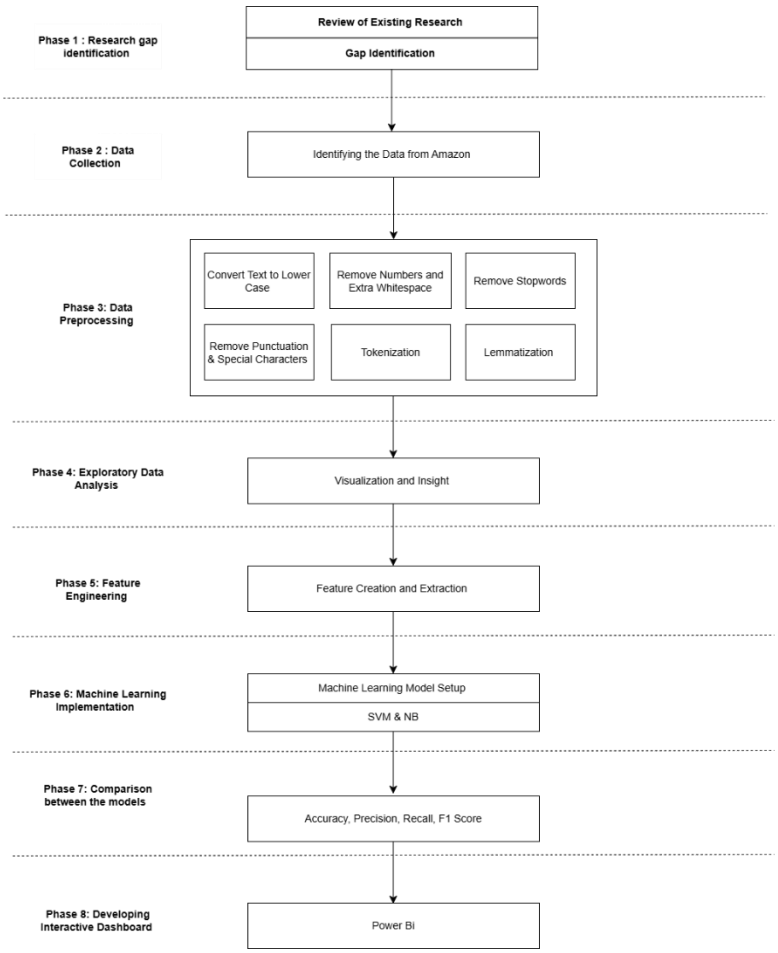

Figure 3.1 Overall research methodology

3.4 Phase 1 Research gap identification

The research starts with getting to know the topic. It begins with a review of the literature of recent research, reading research papers, examining previous models and methodologies, and studying what similar projects have found. The goal is to understand what has been done, what are the outstanding issues, and where do the current methodologies fall short. Once this information is collected, gap identification is a process of critically examining these results to reveal the areas that remain largely unaddressed. The gap may be methodological (e.g., absence of sophisticated preprocessing), contextual (e.g., domain-specific constraints), or performance-based (e.g., poor model accuracy).

Using a sentiment analysis approach to public reactions on Amazon reviews with machine learning techniques such as supported vector machine and Naive Bayes classification, the primary objective of this study is to provide valuable data in order to ensure accurate and reliable analysis. However, in order to accomplish this, a number of issues need to be resolved.

1. Identifying public sentiment regarding the Amazon reviews

2. Comparing the performance of Naive Bayes and SVM models in sentiment classification based on Amazon cell phone and accessories reviews.

3.5 Phase 2: Data Collection

the dataset consisting of reviews of cell phone and accessories reviews was collected from the Amazon Reviews Repository collection [Amazon Reviews'23]. There are a total of 928,301 rows of data that represent each unique review, and there are nine columns that include the following information: overall, verified, reviewerID, asin, reviewerName, summary, title, and brand as shown in figure 3.2. It offered a valuable perspective on the total pleasure of purchasers.

	overall	verified	reviewerID	asin	reviewerName	reviewText	summary	title	brand
0	5.0	True	A236WRQL1MB9HM	7391002801	Morningstar	Beautiful item; received timely. Thank you.	Five Stars	Silver Elegant Butterfly Foot Ankle Chain Summ...	Accessory
1	1.0	True	AN04BLRG7BD8I	7391002801	J. Inman	Had this for 2 weeks. Had to replace screen p...	Outer ring very flimsy.	Silver Elegant Butterfly Foot Ankle Chain Summ...	Accessory
2	1.0	True	A3PHYA8A965CYU	7391002801	Morgan Epperson	The apple is not centered in the hole on the b...	Pretty, but doesnt fit well.	Silver Elegant Butterfly Foot Ankle Chain Summ...	Accessory
3	1.0	True	A3N778P1L4YH9Y	7391002801	McKenna Clark	Case is cheaply made. If you aren't using an a...	Case is cheaply made. If you aren't using an ...	Silver Elegant Butterfly Foot Ankle Chain Summ...	Accessory
4	5.0	True	A3PHJYND753HBC	7391002801	Amazon Customer	This case is a really good thing. When you're ...	Very low price, unbelievably high quality	Silver Elegant Butterfly Foot Ankle Chain Summ...	Accessory

Figure 3.2 Review Dataset

Table 3.2 gave a description of each characteristic in the dataset, along with an explanation of the significance of those attributes as shown below.

Table 3.2 Description of Each Attribute

attributes	Description
Overall	The rating or the overall column is given by the customer from 1 to 5 express the scale of the satisfaction with the product.
Verified	Verified column is a Boolean value which is a True or False that shows that the reviewer comes from a verified purchase or not.
reviewerID	A unique number for each reviewer which can distinguish between the reviewers
Asin	A unique number for each product which stand for Amazon Standard Identification Number.
reviewerName	The display name of the customer usually it's not unique or reliable
ReviewText	The main text of the review part containing textual data which reflect the opinion of the customer. Based on this column going to construct the this project.
Summary	A short summary given by the customer.
Title	The product title
Brand	The name of the brand which helpful of analyzing the brand data.

3.6 Phase 3: Data Preprocessing

When carrying out any kind of data analysis with text that is accompanied, the very first thing to do is to clean the text data. This is done so that some basic steps are watched and can be used to clean and pre-process text data for modelling as well as machine learning. The following steps are the preprocessing steps conducted on the dataset of Amazon reviews. Figure 3-3 shows the coding steps of the cleaning and preprocessing step.

(a) Convert reviews into lowercase
(b) Remove Punctuation & Special Characters
(c) Remove Numbers and Extra Whitespace
(d) Tokenization
(e) Remove Stop-words

Prior to doing the function of the data preparation, before applying it. Standard English stopwords, negation, degree and comparison, and other stopwords, as indicated in the table below, have been included in the stopword list that has been compiled by a custom stopword list.

Table 3.3 Stopword Customization

Category	stopword
Personal pronouns	'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves',
Question words	'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'when', 'where', 'why', 'how',
Common verbs	'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'can', 'will', 'shall', 'would', 'should', 'could', 'may', 'might', 'must'
Conjunctions & prepositions	'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under',
Determiners & quantifiers	'a', 'an', 'the', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such',
Negations	'no', 'nor', 'not', "don't", "doesn't", "didn't", "isn't", "aren't", "wasn't", "weren't", "haven't", "hasn't", "hadn't", "won't", "wouldn't", "can't", "couldn't", "shouldn't", "mightn't", "mustn't", "shan't",
Time & sequence	'again', 'further', 'then', 'once', 'now',
Degree / comparison	'only', 'own', 'same', 'so', 'than', 'too', 'very',
Contractions & variants	'd', 'll', 'm', 'o', 're', 've', 'y', 's', 't',
Polite / filler words	'hi', 'hello', 'hey', 'ok', 'okay', 'yes', 'no', 'thanks', 'thank', 'welcome', 'please', 'etc', 'etc.', 'via', 'regards', 'dear', 'best', 'sir', 'madam',
Extended contractions (common English forms)	"aren’t", "can’t", "could’ve", "couldn’t", "didn’t", "doesn’t", "don’t", "hadn’t", "hasn’t", "haven’t", "he’d", "he’ll", "he’s", "how’d", "how’ll", "how’s", "i’d", "i’ll", "i’m", "i’ve", "it’d", "it’ll", "it’s", "let’s", "might’ve", "mightn’t", "must’ve", "mustn’t", "shan’t", "she’d", "she’ll", "she’s", "should’ve", "shouldn’t", "that’d", "that’s", "there’d", "there’s", "they’d", "they’ll", "they’re", "they’ve", "we’d", "we’ll", "we’re", "we’ve", "what’d", "what’s", "when’d", "when’s", "where’d", "where’s", "who’d", "who’ll", "who’s", "why’d", "why’s", "would’ve",
Slang / short forms	'lol', 'omg', 'btw', 'fyi', 'pls', 'thx', 'u', 'ur', 'r', 'im', 'ya', 'yup', 'nah'

```python
#--- Data Preprocessing ---#

import re
import string

# Convert your list to a set for faster lookups
ENGLISH_STOPWORDS = set(english_stopwords)

def preprocess(text):
    text = str(text).lower()  # lowercase
    text = re.sub(r'\d+', '', text)  # remove numbers
    text = text.translate(str.maketrans('', '', string.punctuation))  # remove punctuation
    text = re.sub(r'[^\x00-\x7F]+', ' ', text)  # remove non-ASCII
    text = re.sub(r'\s+', ' ', text).strip()  # remove extra spaces

    # Remove stopwords
    tokens = [word for word in text.split() if word not in ENGLISH_STOPWORDS]
    return ' '.join(tokens)

# Apply preprocessing
df_agree['preprocessed_text'] = df_agree['cleaned_text'].astype(str).apply(preprocess)
```

Figure 3.3 Data Cleaning and preprocessing

The difference between the text taken before and after the processing processes is shown in table 3.1. It is being prepared for additional analysis.

Table 3.4 The Text Before and After the processing

The Text Before the Processing	The Text After the Processing
“Awesome! stays on, and looks great. can be used multiple times”	“awesome stay look great use multiple times”

So, in figure 3.4 shows the steps of the cleaning and preprocessing steps. First all letters in the review are converted into lowercase. Makes the analysis more consistent because uppercase and lowercase are treated the same. For example, "PHONE" and "phone" are considered the same.

Then, Removes numbers from text. Numbers usually have no meaning in the context of sentiment, unless specifically relevant. Furthermore, removes punctuation and special characters such as ".", ",", "?", "#", "&", etc. Punctuation and special

characters don't contribute directly to sentiment analysis. Also, the whitespace is being removed to make the text neat and easy to read.

Then tokenize the text. In this process is going to split the text into words such as 'this is good' to 'this', 'is', 'good'. Tokenization helps to analysing the text as it helps to separate the text into individual words making it easier to analyse.

Lastly lemmatization is the process where the words is normalized such as 'arrived' become to 'arrive' this helps for modelling the machine learning to understand and getting a high accuracy.

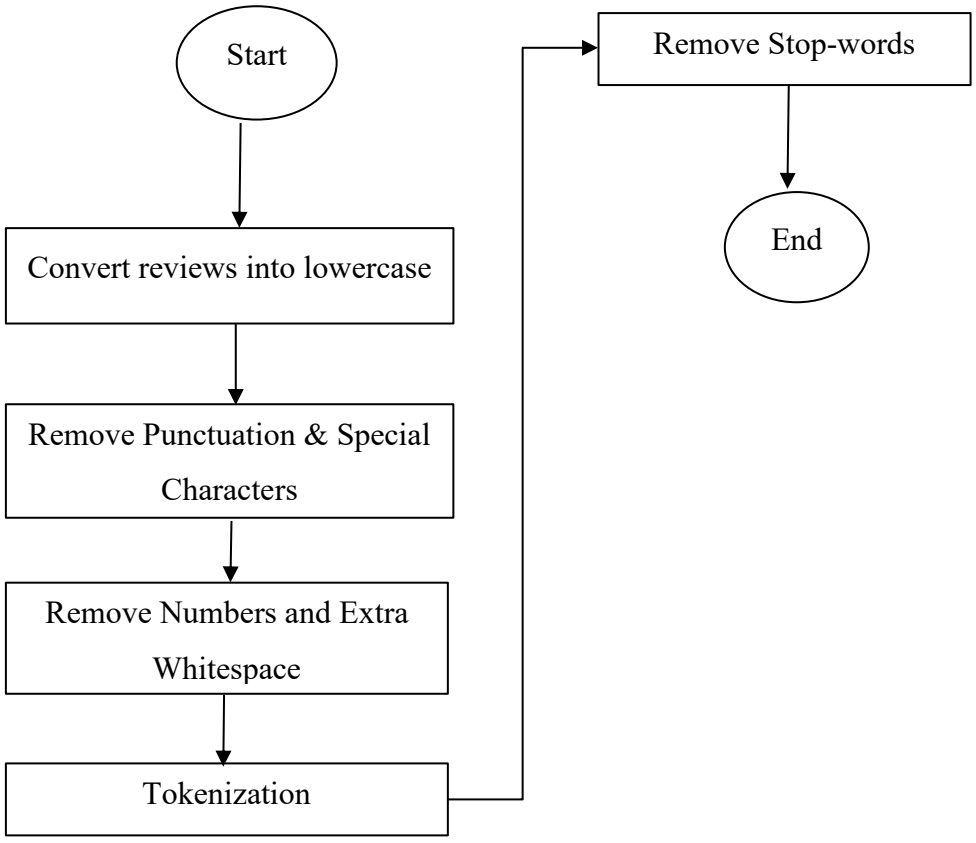

Figure 3.4 Flowchart of Data Cleaning and Preparation

3.7 Phase 4: Exploratory Data Analysis

EDA, or exploratory data analysis, is one of the techniques that can be used to learn all the information about the dataset. The trends, patterns, and relationships present in the data are all included here. It is extremely helpful to have a clear idea of the data structure already set before beginning to construct the machine learning methodology.

A number of essential stages have been carried out in this exploratory data analysis (EDA) in order to get a deeper comprehension of the review dataset's structure and content.

The ratings of the distribution indicate the degree to which customers are pleased with the products, and this is going to be the first time that it is analysed to represent the level of pleasure that customers have with the dataset.

Through a review of the label distribution, as seen in figure 3.5, the coding was carried out, with a particular emphasis placed on the overall rating that was supplied in the dataset. As a result, we were able to observe how reviews are split among the various star ratings, which is beneficial for gaining an understanding of the class balance when it comes to sentiment analysis.

```
# -------------------------
#  Rating Distribution
# -------------------------
plt.figure(figsize=(8,5))
sns.countplot(x='overall', data=df_agree, palette='viridis')
plt.title("Distribution of Ratings (1-5 Stars)")
plt.xlabel("Rating")
plt.ylabel("Count")
plt.show()
```

Figure 3.5 Label Distribution

Then founding the review length as shown the code in figure 3.6 which by quantifying the word number in each review. This helped to test whether users were being verbose while providing their opinion or not, as well as supply information on potential outliers, i.e., reviews that are too short or too long. Then proceeded to examine the most common words used in pre-processed review content. By stopping the common words and lemmatizing as shown the code in figure 3.7, can be determined which most commonly used significant words were used throughout all reviews, giving an idea of common themes or repeated phrases.

```
# Create bins for word counts
bins = [0, 20, 40, 60, 80, 100, 200, 500, 1000]
labels = [f"{bins[i]}-{bins[i+1]-1}" for i in range(len(bins)-1)]
df_agree['word_count_bin'] = pd.cut(df_agree['review_length_words'], bins=bins, labels=labels, include_lowest=True)

# Plot as barplot
plt.figure(figsize=(10,5))
ax = sns.countplot(x='word_count_bin', data=df_agree, color='skyblue')

# Add count labels above each bar
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x() + p.get_width()/2, height + 5,
            str(height), ha='center', va='bottom', fontsize=9, color='black')

plt.title("Binned Distribution of Review Length (Words)")
plt.xlabel("Word Count Range")
plt.ylabel("Frequency")
plt.show()
```

Figure 3.6 Review Length

```
# --------------------------
# Overall Most Common Words
# --------------------------
from collections import Counter

def plot_top_words_overall(texts, n=20):
    all_words = " ".join(texts).split()
    word_freq = Counter(all_words)
    common_words = pd.DataFrame(word_freq.most_common(n), columns=['word','count'])

    plt.figure(figsize=(10,5))
    sns.barplot(x='count', y='word', data=common_words, palette='viridis')
    plt.title(f"Top {n} Most Common Words - Overall Dataset")
    plt.xlabel("Count")
    plt.ylabel("Word")
    plt.show()

# Call function on full dataset
plot_top_words_overall(df_agree['preprocessed_text'], n=20)
```

Figure 3.7 Most Common Words

In order to observe word usage variation by sentiment, constructing word clouds by sentiment as shown the code in figure 3.8. by dividing reviews by their star rating into positive, neutral, and negative and constructed various word clouds for each of them. These visualizations revealed which words appeared most frequently in each sentiment category.

```
# --------------------------
# . WordClouds by Sentiment
# --------------------------
stopwords = set(STOPWORDS)

def show_wordcloud(data, title):
    wc = WordCloud(width=800, height=400, background_color='white',
                   stopwords=stopwords, colormap='viridis').generate(" ".join(data))
    plt.figure(figsize=(10,6))
    plt.imshow(wc, interpolation="bilinear")
    plt.axis('off')
    plt.title(title, fontsize=16)
    plt.show()

show_wordcloud(df_agree[df_agree['label']=='positive']['preprocessed_text'], "WordCloud - Positive Reviews")
show_wordcloud(df_agree[df_agree['label']=='negative']['preprocessed_text'], "WordCloud - Negative Reviews")
show_wordcloud(df_agree[df_agree['label']=='neutral']['preprocessed_text'], "WordCloud - Neutral Reviews")
```

Figure 3.8 Word Clouds by Sentiment

The chapter that follows this one will provide a more in-depth discussion of all of the discoveries that were discovered by the EDA analysis.

3.8 Phase 5: Feature Engineering

In this task of sentiment analysis, feature engineering was the key to turning unprocessed product review data into informative inputs which could be leveraged in training the machine. The process was meant to identify helpful patterns in both review text and available information such as user ratings and review length to support improving the accuracy and performance of the sentiment classification model.

In the first place, to deal with the problem of the data being imbalanced. The first release of the data set had a greater number of favourable reviews than either neutral or negative ratings. Utilising an under-sampling strategy that lowered the sample size of each class to that of the lowest group, as shown by the code in figure 3.9, in order to eliminate bias in the prediction of the model. During the training

process, the model will be biassed towards the no sentiment classes more than other different sentiment classes.

```
#-- Balancing --#

df_majority = df_agree[df_agree['label'] == 'neutral']
df_pos = df_agree[df_agree['label'] == 'positive']
df_neg = df_agree[df_agree['label'] == 'negative']

df_pos_up = resample(df_pos, replace=True, n_samples=len(df_majority), random_state=42)
df_neg_up = resample(df_neg, replace=True, n_samples=len(df_majority), random_state=42)
df_balanced = pd.concat([df_majority, df_pos_up, df_neg_up])

df_balanced = df_balanced.dropna(subset=['preprocessed_text'])
df_balanced = df_balanced[df_balanced['preprocessed_text'].str.strip() != '']
```

Figure 3.9 Under-Sampling Technique

In addition, the labelling of the data for the sentiment labelling in this project is carried out in two phases, the first of which is the VADER stage, and the second step is the TextBlob stage. When the two phases finished using both strategies, go to the Keep just agreement instances section. When compared to using just one tool or raw ratings, the labels that are produced using this method are more accurate. as shown the code in figure 3.10.

```
# VADER + TextBlob setup
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from textblob import TextBlob
import nltk

nltk.download('vader_lexicon')
nltk.download('punkt')

vader = SentimentIntensityAnalyzer()

# === 2. Define label functions ===
vader = SentimentIntensityAnalyzer()

def vader_label(text, pos_thresh=0.05, neg_thresh=-0.05):
    score = vader.polarity_scores(text)['compound']
    if score >= pos_thresh:
        return "positive"
    elif score <= neg_thresh:
        return "negative"
    else:
        return "neutral"

def textblob_label(text, pos_thresh=0.05, neg_thresh=-0.05):
    score = TextBlob(text).sentiment.polarity
    if score >= pos_thresh:
        return "positive"
    elif score <= neg_thresh:
        return "negative"
    else:
        return "neutral"

# === 3. Apply both labelers ===
df['vader_label'] = df['cleaned_text'].apply(vader_label)
df['textblob_label'] = df['cleaned_text'].apply(textblob_label)

# === 4. Keep only agreement cases ===
df_agree = df[df['vader_label'] == df['textblob_label']].copy()
df_agree['label'] = df_agree['vader_label']  # final label

print(f"Original dataset size: {len(df)}")
print(f"After agreement filter: {len(df_agree)}")
```

Figure 3.10 Sentiment Labelling

Also, extracted some review-based features as shown the code in figure 3-12. They included the review length (number of words in a review), number of characters, and average word length. These tell the model how long or short a review is, which in certain instances reflects how strong the reviewer's opinion is. Also, it applied to this feature normalization of the output because the output can vary from one review to another. The normalization is between 0 and 1 to prevent the outliers. The code of this feature as shown below in figure 3.11.

```
class ReviewFeatures(BaseEstimator, TransformerMixin):
    """Simple review meta features (non-negative): char_len, word_len, !, ?, CAPS ratio."""
    def __init__(self):
        self.scaler = MinMaxScaler()

    def fit(self, X, y=None):
        feats = self._extract(X)
        self.scaler.fit(feats)
        return self

    def transform(self, X):
        feats = self._extract(X)
        feats = self.scaler.transform(feats)
        return csr_matrix(feats)

    def _extract(self, X):
        X = pd.Series(X).fillna("").astype(str)
        rows = []
        for text in X:
            char_len = len(text)
            words = re.findall(r"\b\w+\b", text)
            word_len = len(words)
            exclam = text.count("!")
            qmark = text.count("?")
            caps = sum(1 for c in text if c.isupper())
            caps_ratio = (caps / char_len) if char_len else 0.0
            rows.append([char_len, word_len, exclam, qmark, caps_ratio])
        return np.asarray(rows, dtype=float)
```

Figure 3.11 Reviews Feature

Another essential process was pulling out Part-of-Speech (POS) tag features. With the help of NLP, it can able to count the occurrences of nouns, verbs, adjectives, and adverbs in each review as shown the code in figure 3.12. This is important because adverbs and adjectives have rich emotional content, while nouns and verbs provide information about the product and the actions. These grammatical patterns can help the model understand better how people express their opinions.

```
class POSFeatures(BaseEstimator, TransformerMixin):
    """Counts of POS buckets: nouns, verbs, adjectives, adverbs per doc (min-max scaled)."""
    def __init__(self):
        self.scaler = MinMaxScaler()

    def fit(self, X, y=None):
        feats = self._extract(X)
        self.scaler.fit(feats)
        return self

    def transform(self, X):
        feats = self._extract(X)
        feats = self.scaler.transform(feats)
        return csr_matrix(feats)

    def _extract(self, X):
        X = pd.Series(X).fillna("").astype(str)
        rows = []
        for text in X:
            tokens = word_tokenize(text)
            tags = pos_tag(tokens, tagset=None)
            n_noun = sum(t[1].startswith("NN") for t in tags)
            n_verb = sum(t[1].startswith("VB") for t in tags)
            n_adj  = sum(t[1].startswith("JJ") for t in tags)
            n_adv  = sum(t[1].startswith("RB") for t in tags)
            rows.append([n_noun, n_verb, n_adj, n_adv])
        return np.asarray(rows, dtype=float)
```

Figure 3.12 POS Tag Features

Furthermore, With the use of an emotion lexicon, the polarity count features are able to determine the total amount of positive and negative words that are included in each review. In this context, a larger positive count often signifies positive feedback, while a higher negative count suggests displeasure. These qualities give a clear and interpretable indicator of mood. When used in conjunction with sparse features such as TF-IDF, they provide a compact and dense representation that assists models in successfully distinguishing between different sentiments. as shown the code in figure 3.13.

```
class PolarityCountFeatures(BaseEstimator, TransformerMixin):
    """Counts of positive/negative lexicon hits (non-negative, scaled)."""
    def __init__(self):
        self.pos_set = set(opinion_lexicon.positive())
        self.neg_set = set(opinion_lexicon.negative())
        self.scaler = MinMaxScaler()

    def fit(self, X, y=None):
        feats = self._extract(X)
        self.scaler.fit(feats)
        return self

    def transform(self, X):
        feats = self._extract(X)
        feats = self.scaler.transform(feats)
        return csr_matrix(feats)

    def _extract(self, X):
        X = pd.Series(X).fillna("").astype(str)
        rows = []
        for text in X:
            tokens = [w.lower() for w in re.findall(r"\b[a-zA-Z']+\b", text)]
            p = sum(w in self.pos_set for w in tokens)
            n = sum(w in self.neg_set for w in tokens)
            rows.append([p, n])
        return np.asarray(rows, dtype=float)
```

Figure 3-13 Polarity Count Feature

Finally, transforming the cleaned review text into numerical form using TF-IDF vectorization with n-grams as shown the code in figure 3.14. TF-IDF is used to identify the most important words or phrases in a review, especially those that are rare or less common throughout the dataset. Using both unigrams (single words) and bigrams (two-word phrases) allows the model to more accurately learn context, such as catching negative two-word phrases like "not good" or positive two-word phrases like "very useful.". Chi-Square parameter (K) will discuss in details in “chapter 5”

```
# TF-IDF + chi2 selection
tfidf_chi2 = Pipeline([
    ("tfidf", TfidfVectorizer(stop_words="english",
                              sublinear_tf=True,
                              max_features=15000,
                              ngram_range=(1, 2))),
    ("chi2", SelectKBest(chi2, k=5000))
])

combined_features = FeatureUnion([
    ("tfidf_sel", tfidf_chi2),  # sparse
    ("pos", POSFeatures()),       # csr_matrix
    ("review_feats", ReviewFeatures()),  # csr_matrix
    ("polarity", PolarityCountFeatures())  # csr_matrix
])
```

Figure 3.13 TF-IDF with N-grams and Combining All Features

In general, this feature engineering process combined text analysis, social validation cues, writing style, and language use to create a dense and informative dataset. All these features engineering will help the model to better grasp and categorize the sentiment of each review, and this will lead to better prediction outcomes.

3.9 Phase 6: Machine Learning Models

In this project, the construction of sentiment classification models from Naive Bayes and Support Vector Machine is a step-by-step and rational process for which each crucial step that has to be adhered to in order to transform raw textual data into a format that can be utilized by machine learning models to learn from and predict.

The first and second step that have been done earlier which is loading the data and the feature engineering have been discussed in the previous section. Next, the data has been split as 80% training and 20% testing. This is a crucial step to check how well our models will generalize to new, unseen data. It prevents overfitting and gives us a real-world estimate of the performance.

In the first, training a Naive Bayes model. Naive Bayes is a fast and efficient algorithm that does very well on text data, especially when the features are sparse, such as in TF-IDF matrices. In the second model, training a Support Vector Machine model, which is a powerful classifier and well known for its robustness in high-dimensional spaces. While it requires more computational time and hyperparameter tuning, SVM performs better, especially when using a linear or RBF kernel.

after training both models on the trained set and predict on the test set. By doing this, it can understand how well the models work in actual life data. Finally, evaluating the performance of both models by verifying their classification using measures of accuracy, precision, recall, and F1 score. The measures show how well the two models recognize and predict the sentiment from the review content.

3.10 Chapter Summary

The research process is covered in a precise section breakdown within this chapter, beginning with the data collection and followed by model category testing and assessment. The process ensures that the step-by-step approach of carrying out the sentiment analysis of Amazon reviews is done in a systematic and data-intensive process.

CHAPTER 4

EXPLORATORY DATA ANALYSIS

4.1 Overview

This chapter presents the EDA and some analytics conducted on Amazon Reviews Repository collection [Amazon Reviews'23]. The purpose of this chapter is to have initial insight about the behavior of the customers opinion and driving meaningful insights from the EDA and the features engineering to prepare the data for the machine learning. Furthermore, machine Learning classifiers, including Support Vector Machine (SVM) and Naïve Bayes, were used in the study to evaluate predictive performance which can support actional business decisions.

4.2 Dataset Overview

After the preprocessing and cleaning methods that were discussed in chapter 3, the shape of the data is 708527 rows. This occurred after the steps were completed. Every single review that was published was categorised into three distinct emotion categories: positive, negative, and neutral. An explanation of how each feeling is categorised may be found in the table that follows.

Table 4.1 Dataset classification by sentiment

Dataset	Total Comments	Positive	Negative	Neutral
	708527	625058	56139	34265

4.3 Exploratory Data Analysis

Figure 4.1 illustrates the binned distribution of review lengths in words. The distribution is highly right-skewed, with the majority of reviews being extremely short. The largest group of reviews falls within the 0–19-word range, accounting for over 410,000 reviews. The next most common range is 20–39 words with about 165,000 reviews, after which the frequencies decline sharply. Reviews exceeding 100 words are relatively uncommon, and extremely long reviews (500–999 words) are rare, with fewer than 600 instances in the dataset. This indicates that most customers tend to leave brief feedback, often just a few words or short sentences, which is typical in e-commerce platforms where quick impressions or direct opinions are provided. Although long reviews are rare, they often contain richer contextual information and more nuanced sentiment, making them valuable for deeper analysis. The overall pattern suggests that the dataset is dominated by short reviews, which can pose challenges for sentiment classification due to limited context, while the small proportion of longer reviews introduces variability that may require special consideration during preprocessing.

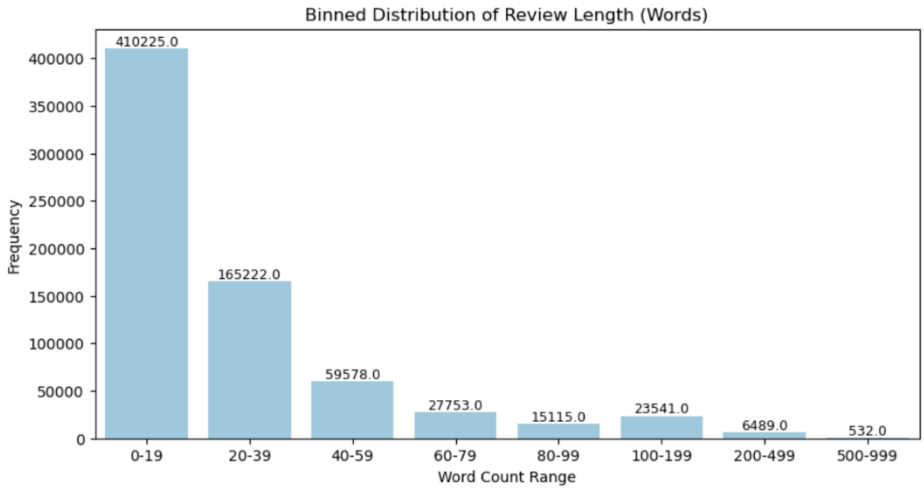

Figure 4.1 Review Length Distribution

The bar chart in figure 4.2 shows the frequency distribution of product review scores between 1 and 5. It can be observed that the data set is highly biased toward positive ratings as 5-star ratings are dominant in the distribution. Specifically, there are the most common 5-star reviews, followed by a significant number of 4-star reviews. By comparison, the lower rated ones 1-star and 2-star appear much less often,

suggesting that unhappy customers are less common or less likely to comment. Class imbalance of this sort is one of the principal concerns for any sentiment analysis or classification model as it might lead to biased predictions in the direction of the majority class unless properly handled.

This rating class imbalance can affect the model's ability to learn from negative or neutral feedback since it has fewer negative or neutral instances. Unless corrected, the model could perform well on positive feedback but may not have the ability to label less frequent but equally important negative ones accurately. Therefore, this visualization highlights the need for techniques such as resampling, class weighting, or data augmentation to ensure that the model is not biased towards any of the sentiment categories during training. Regarding the rating class imbalance issue is going to be discussed in details in the feature engineering section.

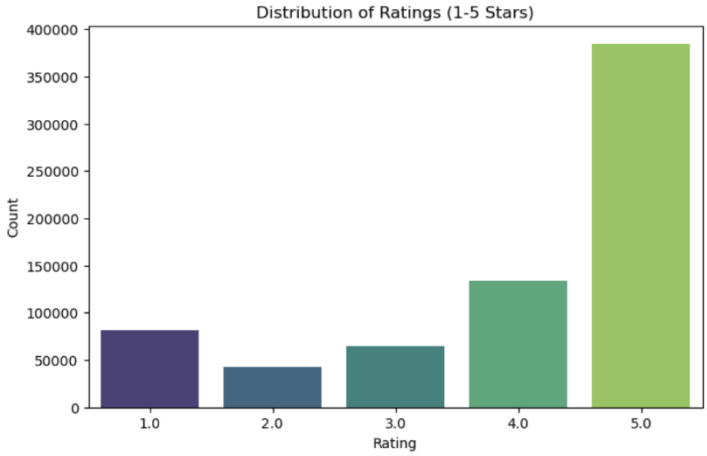

Figure 4.2 Review Rating Distribution

The bar chart in figure 4.3 shows the 20 most frequent words in the customer reviews dataset, providing a sense of what language is characteristic of product reviews. The most common term is "phone", followed by "case", likely because the dataset focuses on mobile phone accessories or electronics. These two words appear far more than any others, with more than 500,000 mentions each, suggesting that the majority of reviews are for phone products.

Other commonly used words are "one," "like," "great," "use," and "well," which are a mix of product words and sentiment words. Words such as "great," "good,"

"really," and "well" express positive sentiment, showing that a large percentage of the reviews may be positive.

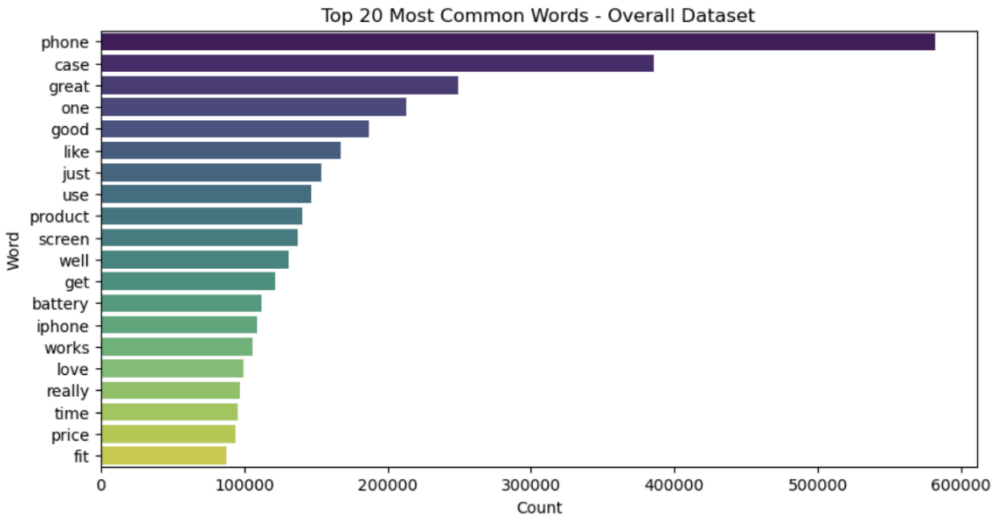

Figure 4.3 Top 20 Common Words

The description of the dataset, which includes fundamental statistical analysis such as the mean, standard deviation, minimum and maximum values, is displayed in figure 4.4.

	overall	review_length
count	708527.000000	708527.000000
mean	3.984332	29.955965
std	1.382597	43.145571
min	1.000000	0.000000
25%	3.000000	11.000000
50%	5.000000	17.000000
75%	5.000000	33.000000
max	5.000000	2561.000000

Figure 4.4 Dataset Description

Figure 4.5 is the illustrates the distribution of sentiment of each class by using the VADER and TextBlob methods and taking the common which ensure that the labelling is accurate and not misclassifying the reviews.

VADER is a method for analyzing feelings that works with short, casual writings like product evaluations. It uses a lexicon and rules to do this. It gives terms polarity scores and looks at how they are employed in language. VADER provides you the percentages of happy, neutral, and bad feelings, as well as a compound score that ranges from -1 (most negative) to +1 (most positive) to represent how you feel overall.

TextBlob is another way to use a lexicon that is built on the NLTK and Pattern libraries. It tells two main key things: polarity, which ranges from -1 to +1 to express how the feeling about something, and subjectivity, which ranges from 0 (objective) to 1 (subjective). is strong at figuring out how others feel in general and discerning the difference between fact and opinion.

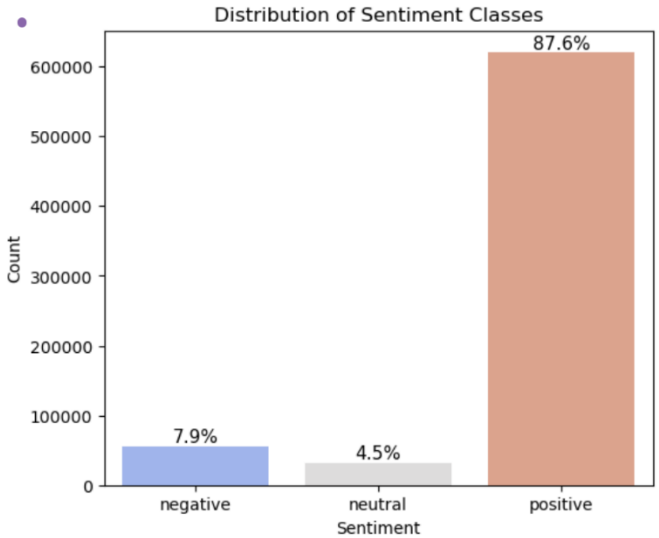

Figure 4.5 Distribution of Sentiment Classes

Figure 4.6 uses a logarithmic scale to show how the lengths of reviews are distributed out across ratings and sentiment categories. The results show that neutral reviews are always shorter than positive or negative ones, no matter how many stars they obtain. Customers tend to write longer and more detailed reviews when they are happy or upset. This is because they tend to give more information when they are pleased or sad. On the other hand, neutral feedback is usually short and to the point. The main pattern is the same for all ratings, which means that the length of a review is more determined by how positive or negative the feeling is than by the rating itself. The outliers in the plot are exceptionally long reviews that happen in every category. This shows that just a few customers submit feedback that is substantially more detailed than typical. Using a log scale makes these distributions simpler to discern since it makes extreme values less important, which would otherwise make the picture look cluttered.

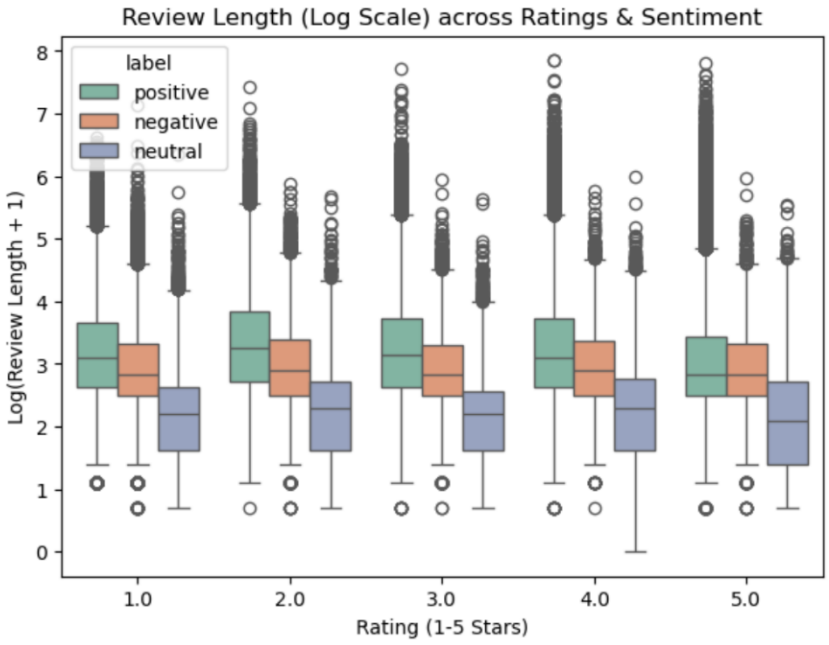

Figure 4.6 Boxplot of Review Lengths

A word cloud including reviews with positive sentiments was shown in the figure 4.7. Through the use of word cloud analysis, it was determined that the words "fit," "good," "love," "great," "recommended," "perfect," "highly", "well", "better" were the ones that were used the most often in the evaluation of the products. It was clear that the products were helping them to meet with their needs since the term "great", "good quality" was used. The term "work great" was a representation of the products that satisfy them. During this time, the words "recommended" and "love" were used to symbolize their satisfaction of the price of the products which is affordable.

Figure 4.7 World Cloud of Positive Sentiment

A word cloud including reviews with negative sentiments was shown in the figure 4.8. Through the use of word cloud analysis, it was determined that the words "horrible," "waste," "broken," "disappointed," and "bad" were the ones that were used the most often in the reviews of the products. The words "annoying," "broke," and "issue " demonstrate that a significant number of individuals will have a negative attitude towards a number of the supplies that have been purchased.

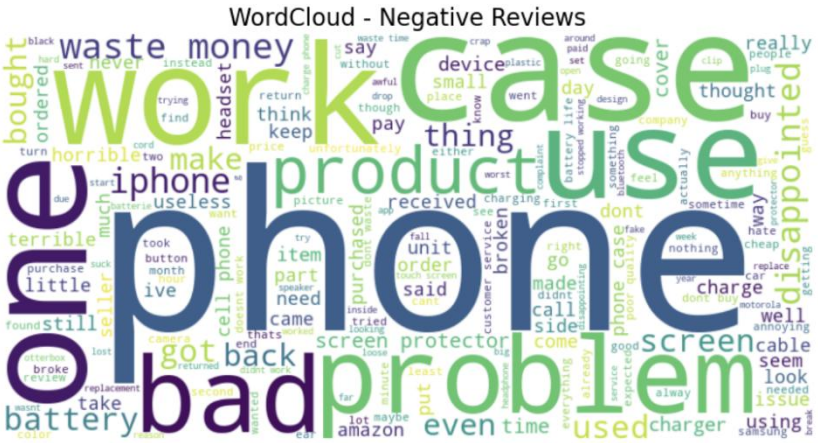

Figure 4.8 World Cloud of Negative Sentiment

Figure 4.9 illustrate the word cloud for neutral sentiment reviews. Word cloud analysis illustrated that "use," "phone," "still," "work," were the most frequent used words in the review. These words can represent that the products don't meet their meet as their expectation.

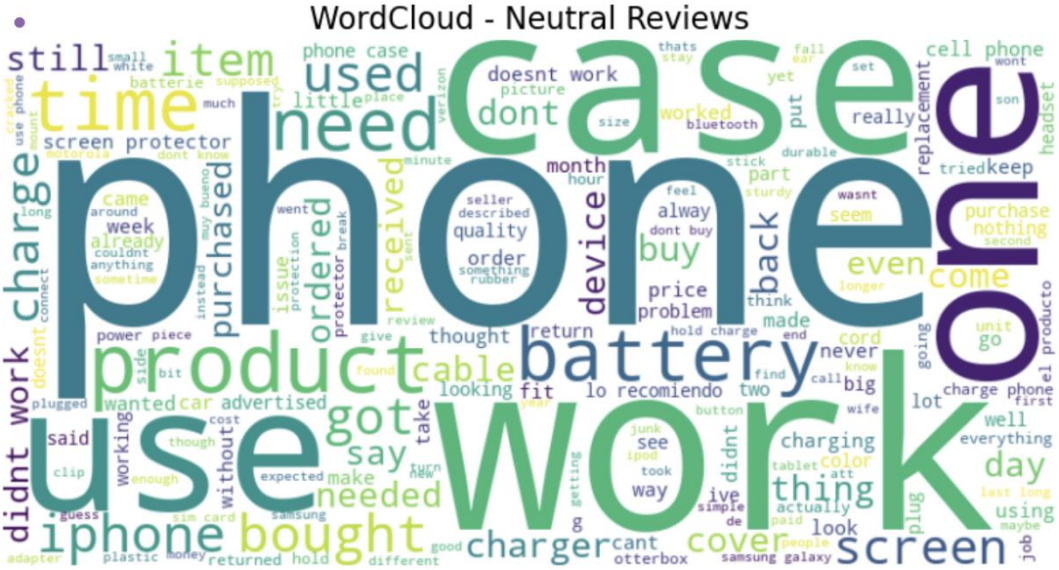

Figure 4.9 World Cloud of Neutral Sentiment

4.4 Feature Engineering

Feature engineering is the process of creating new input features or modifying existing ones to improve machine learning model performance. It's extracting useful information from raw data so that algorithms are better able to identify patterns and

relationships. Adding domain knowledge and selecting the correct features, this step enables the model to produce accurate predictions and generalize well to unseen cases. Feature engineering is typically considered as one of the most critical tasks in the development of efficient machine learning machines.

4.4.1 Class Sampling

Class balancing using an under-sampling technique was the first task performed on the dataset for this project. Real-world datasets, specifically product reviews, are prone to class imbalance—where the number of positive reviews far outweighs neutral and negative reviews. Class imbalance can negatively impact class distribution-sensitive machine learning algorithms such as Support Vector Machines (SVM) and Naive Bayes (NB). To counter this, a custom function was implemented to balance the dataset by random sampling of an equal number of instances from each sentiment class based on the population of the smallest class. This balances the model's learning across all sentiment classes so that the model does not become biased towards the majority class, and fairness in prediction is improved overall. Class balancing is necessary to enhance the credibility of evaluation metrics such as precision, recall, and F1-score for every sentiment label, thereby leading to more robust and generalizable models.

4.4.2 Review-Based Features

other review-based features were created to collect the structural and stylistic attributes of the text. They are review_length, which tallies up the words in every review; char_count, which calculates the characters; and avg_word_length, which is calculated by dividing the number of characters into the number of words (with a slight adjustment so it is never divided by zero). These features help the model to understand how a review is structured, as opposed to what's in the review. For example, longer reviews might be stronger opinions, while average word length might convey tone or complexity of the text. By incorporating these features into text features like TF-IDF, the model is given more context that enhances its performance in being able to identify

positive, negative, and neutral sentiments. This increases the overall accuracy and stability of sentiment classification with models such as SVM and Naive Bayes.

4.4.3 Polarity Count Features

The polarity count features take matches from the opinion lexicon corpus, which provides lists of good and bad words. For each review, we count how many good and negative words there are. This shows clearly how the individual feels. For instance, words like "excellent," "fantastic," "poor," or "terrible" might strongly express how someone thinks about something. These traits are quite useful for short evaluations that don't contain a lot of background information but do have clear emotion words. The values are scaled like the other feature sets so that longer texts don't get more weight.

4.4.4 POS Tag Features

Part-of-Speech (POS) tag-based feature features were extracted from the pre-cleaned reviews that might complement sentiment classification. The extract POS features function tokenizes each review, performs part-of-speech tagging on each one using the Natural Language Toolkit (NLTK), and determines how many nouns, verbs, adjectives, and adverbs the text contains. These features are crucial because they locate words playing different roles of describing meanings. For example, adverbs and adjectives are subject to high emotional load ("amazing," "terribly"), so they are strongly applicable to sentiment detection. Verbs and nouns, which are less emotional, provide syntactic and contextual information to enable models to understand sentence purpose. By adding these features to the dataset, the model is syntactically better informed and thus able to distinguish more effectively among positive, negative, and neutral reviews. This feature boosts other features like TF-IDF and review structure to increase the performance and stability of classifiers like SVM and Naive Bayes.

4.4.5 TF-IDF with N-grams

To convert word-level feedback to machine learning suitable numerical representations, Term Frequency-Inverse Document Frequency (TF-IDF) technique was utilized employing bigrams and unigrams. In this method, each word or two-word combination (e.g., "not good", "great product")'s worth or importance for each individual review within the entire corpus is being calculated. The TfidfVectorizer was configured with a max features parameter of 15000 to retain only the most informative words, and the ngram_range parameter was set at (1, 2) to capture single words and two-word phrases. This allows the model to not only detect the occurrence of specific words but also within what context the words are used. For instance, the tone of "not satisfied" is significantly different from standalone words "not" and "satisfied" alone. The inclusion of n-grams assists in adding weight to the model's perception to capture the nuances of senses that result in accurate prediction, more so when accompanied by other engineered features included in models such as SVM and Naive Bayes.

4.5 After the Feature Engineering

Before feature engineering, the data was in the form of raw review text and little metadata such as ratings. While sufficient for basic sentiment analysis, it was unstructured and did not contain informative features. After feature engineering, the data had been enriched with cleaned text and features such as word count, character count, average word length, helpfulness ratio, and part-of-speech counts (nouns, verbs, adjectives, adverbs). These additional features provided more context and improved the model's ability to recognize patterns in sentiment, resulting in improved classification performance compared to using raw text alone. As shown below the data structure that entered the machine learning models. The other features are integrated in the pipeline of the models.

textblob_label	label	preprocessed_text	review_length_words	review_length_chars	word_count_bin	review_length	vader_score	textblob_score	final_label
neutral	neutral	met expectations	2	16	0-19	2	0.0	0.0	neutral
neutral	neutral	excelente	1	9	0-19	1	0.0	0.0	neutral
neutral	neutral	everything came supposed durable case rhinesto...	10	74	0-19	10	0.0	0.0	neutral
neutral	neutral	said take days month another mother never got ...	10	56	0-19	10	0.0	0.0	neutral
neutral	neutral	bow fell multiple times keep gluing back would...	16	93	0-19	16	0.0	0.0	neutral

Figure 4.10 Data structure after the FE

4.6 Chapter Summary

Exploratory Data Analysis (EDA) was performed in order to understand the structure, distribution, and character of the review data before training the model. Followed by the feature engineering which enhance the dataset with enriched features. Now the dataset is ready for splitting process and developing the machine learning models.

CHAPTER 5

MODEL DEVELOPMENT

5.1 Introduction

This chapter indicates the full version of the model development process that has been taken in for sentiment analysis of Amazon cell phone and accessory reviews. It presents the continuous processes that have been applied to make the row dataset into a dataset that suitable for the classification of the machine learning. Steps begin with the data preparation which include the processing of the text. Followed by, labelling the sentiments through a hybrid method using VADER and TextBlob. Next phase involves the feature engineering, which multiple of features are extracted from the data, including TF-IDF, POS, reviews features, and polarity lexicon count.

Also, class imbalance issue present in the dataset, which make under sampling technique is applied to make the data well distributed across the sentiment categories. Then, the chapter introduce the development of two machine learning classifier which are Naïve Bayes and Support Vector Machine. Introducing of the parameters of each model. Then, the chapter going to introduce the performance metrics of the evaluation of the models. Lastly, a discussion of the development of the interactive dashboard using Microsoft Power BI software.

5.2 Dataset Overview

As being introduced of the dataset in chapter 3, the dataset of the project is obtained from Amazon Reviews Repository, specifically focusing on the Cell Phones and Accessories category. The dataset contains attribute such as review text, asin, overall rating, and brand. These attributes are the main of this project of the sentiment analysis classification.

The textual content is the main purpose of the input for the sentiment classification. Also, for the data labelling which will be discuss below of this chapter.

5.2.1 Data preprocessing

As it's discussed in chapter 3 and figure 3.3 which is Flowchart of Data Cleaning and Preparation the steps that involve of the data preprocessing. The transformation of reviews data from their raw form to the final form that is utilized in the classification model is shown in Table 4.1 below. it gets a better understanding of the actual influence that the preprocessing step has on the reviews data.

Table 5.1 Transformation of the Reviews data

Preprocessing Steps	**Example**
Before preprocessing	"This Ipad is AMAZING!!! Battery lasts 3 days :) Totally worth it… 👍"
Convert reviews into lower case	"this phone is amazing!!! battery lasts 3 days :) totally worth it… 👍"
Remove punctuation and special characters	"this phone is amazing battery lasts 3 days totally worth it"
Remove numbers and extra white space	"this phone is amazing battery lasts days totally worth it"
Tokenization	["this", "phone", "is", "amazing", "battery", "lasts", "days", "totally", "worth", "it"]
Remove stopwprds	["phone", "amazing", "battery", "lasts", "days", "totally", "worth"]
After preprocessing	"phone amazing battery lasts days totally worth"

5.2.2 Sentiment Labelling

It has been used a hybrid lexicon-based method that included VADER and TextBlob to get trustworthy class labels for supervised learning. Both tools always yield polarity values between −1 and +1. VADER also uses degree adverbs, and negation to figure out how strong something is. TextBlob offers an alternate polarity

measure and subjectivity, the latter of which is not utilized for labeling. Using two methods of sentiment analyzer makes it less probable that the labels are wrong or biased by just one tool.

To get a compound sentiment score for each review, VADER does the calculation. The overall sentiment polarity is reflected by the compound score, which runs from -1 (very negative) to +1 (extremely positive) and is based on the intensity and modifiers that are present in the text.

In addition to producing a subjectivity score that may range from 0 (objective) to 1 (subjective), TextBlob also generates a polarity score that can be anywhere between -1 and +1. Only on this particular model When converting ratings, the polarity score is taken into consideration.

A comparison between VADER and TextBlob is shown in the table that may be seen below.

Table 5.2 VADER vs TextBlob Sentiment Analysis Model

Criteria	**VADER**	**TextBlob**
Lexicon Based	Yes	Yes
Handles Negation	Yes	No
Emphasizes Sentiment Intensity	Yes	No
Customizability	Limited	Moderate
Accuracy in Mixed Sentiment	High	Moderate
Requires Training Dat	No	Yes
Provides Subjective Analysis	No	Yes
Support for Slang and Abbreviation	Yes	No
Dependency Parsing	No	Yes

Table 5.3 shows the output of VADER and TextBlob from Python in labelling reviews. Both methods offered each review a polarity score between -1 and +1 and a label for the sentiment. Both VADER's compound score and TextBlob's polarity score consistently rated the reviews as good. The last column of labels represents the agreed sentiment, which was preserved for supervised learning. This example shows how the method makes sure that the reviews are reliable by only keeping the ones that both analyzers agree on the sentiment categorization.

Table 5.3 Example of Sentiment Labeling using VADER and TextBlob

Review	VADER Score	VADER Label	TextBlob Score	TextBlob Label	Final Label
beautiful item received timely	0.5994	positive	0.850000	positive	positive
met expectations	0	neutral	0	neutral	neutral
hate looks cheap plastic	-0.5719	negative	-0.2	negative	negative

The graph of 5.1 demonstrates how VADER and TextBlob sort feelings. In both cases, most of the ratings are excellent, but the way neutral views are spaced out is different. TextBlob marks a larger percentage of reviews as neutral (14.2%) than VADER, which only marks only 7.2% of them as neutral. This means that VADER usually puts evaluations into good or bad groups, but TextBlob is more careful. Using both together helps battle these urges and generates emotion labels that are more even.

Figure 5.1 Sentiment Label Distribution: VADER vs TextBlob

The heatmap below shows how well the sentiment labels from TextBlob and VADER work together. The darker diagonal cells show where both tools put reviews in the same group. The lighter off-diagonal cells show where they didn't agree. The two methods agreed on about 78% of the whole dataset, with the most agreement in the positive class. Despite the fact that the approaches sometimes disagree with one another, this demonstrates that the areas of agreement between them provide a more credible foundation for labelling.

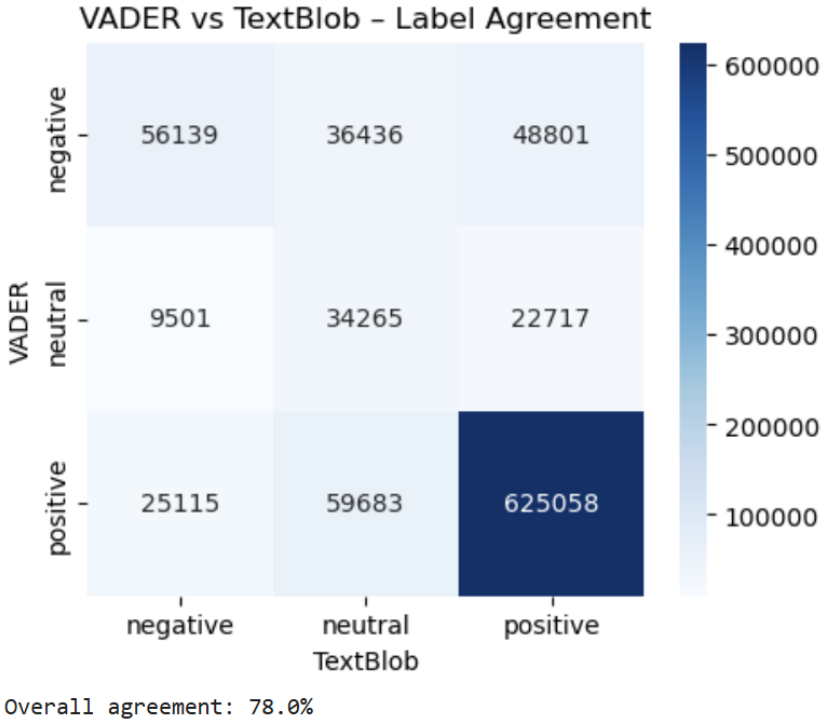

Figure 5.2 Heatmap of Label Agreement between VADER and TextBlob

the bar chart below illustrates the agreement between VADER and TextBlob for each sentiment class. The two tools align most strongly on positive reviews, with an agreement of 88.1%. Neutral reviews show a moderate agreement of 51.5%, while negative reviews have the lowest agreement at 39.7%. These results highlight that while positive sentiments are consistently recognized, neutral and negative sentiments are more prone to disagreement, reinforcing the need for a hybrid approach to improve labelling reliability.

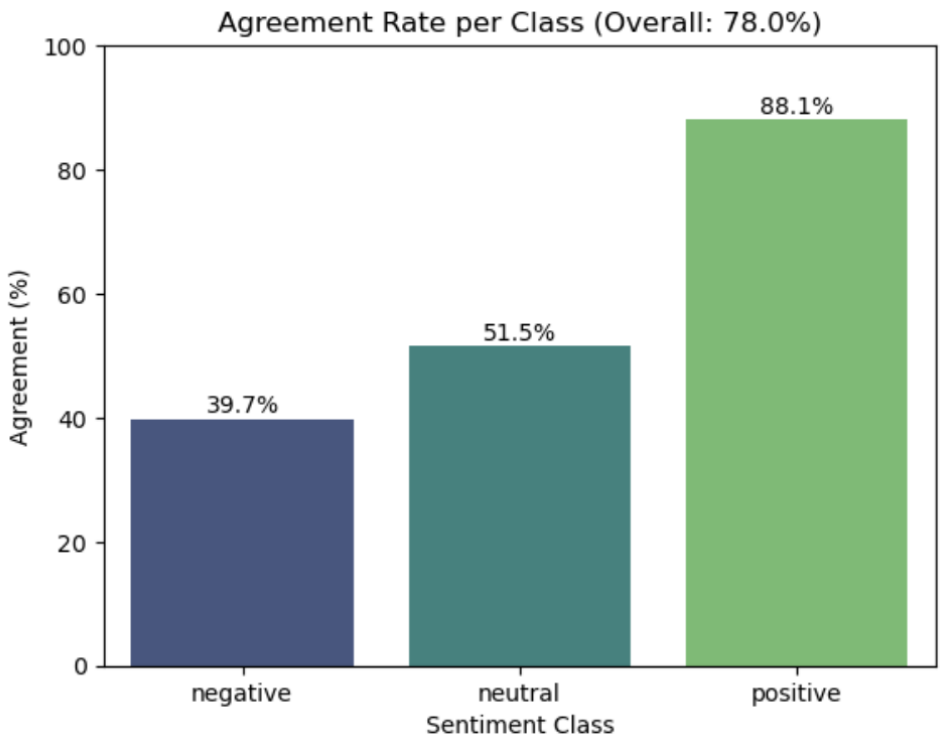

Figure 5.3 Agreement by Sentiment Class between VADER and TextBlob

5.3 Feature Engineering

There were many various sorts of attributes that were developed in order to improve the dataset. Due to the fact that raw text alone does not always represent the whole spectrum of emotions, it has been decided to include additional linguistic and stylistic indications in order to strengthen the models. Cures based on grammar, structure, and vocabulary were intended to be included into these factors in order to bring together the most beneficial aspects of word frequency techniques.

5.3.1 Term Frequency–Inverse Document Frequency (TF-IDF)

The primary feature extraction technique employed was Term Frequency–Inverse Document Frequency (TF-IDF), which transforms text into numerical vectors

according on the relative significance of words and phrases. It has been kept both single words and bigrams so that phrases like "very good" or "not bad" were still there. These phrases typically show stronger feelings than single words do. After trying out numerous thresholds, it has been found that the maximum vocabulary size should be 15,000. At this level, the model struck a nice compromise between performance and efficiency: it kept training time modest while still covering approximately 98% of the tokens. Discussed below the different maximum vocabulary size against some variable.

This figure below illustrates how the performance of the model in terms of classification is impacted by the number of maximum features that are included in the TF-IDF representation. The macro F1 score steadily increases as the number of features increases from 5,000 to 30,000. This raises the score. This indicates that the model improves its ability to cope with a wide range of negative emotions. When there are between 5,000 and 15,000 characteristics, the most significant shift occurs. After then, the adjustments will become more consistent, which indicates that the advantages will become less significant. In spite of the fact that it is capable of doing calculations in a short amount of time, the model already captures practically all of the necessary vocabulary at 15,000 attributes. As a result, it is an excellent point of balance.

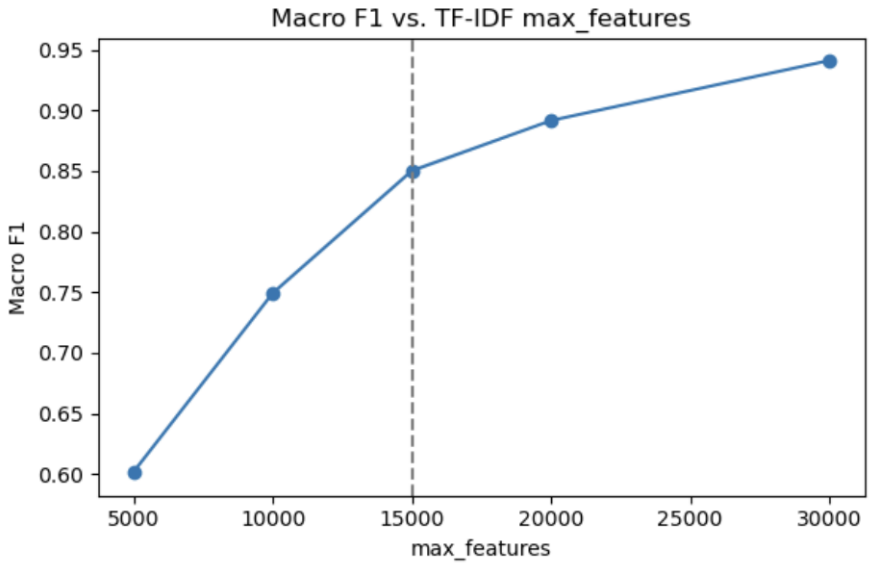

Figure 5.4 Macro F1 Vs Tf-IDF max features

This graph illustrates how the length of time spent training is impacted by the number of TF-IDF features that are currently being utilized. In proportion to the increase in the maximum number of features beyond 10,000, the amount of time required for training also increases. The reason for this is that the processing of larger vocabularies incurs more expenses. A short-term increase in efficiency is seen by the decline at 10,000 features; nevertheless, the average trend illustrates that an increase in the number of features requires an increase in the amount of processing power. The selection of 15,000 features finds a decent balance between achieving good performance and enabling training to be completed in a reasonable amount of time.

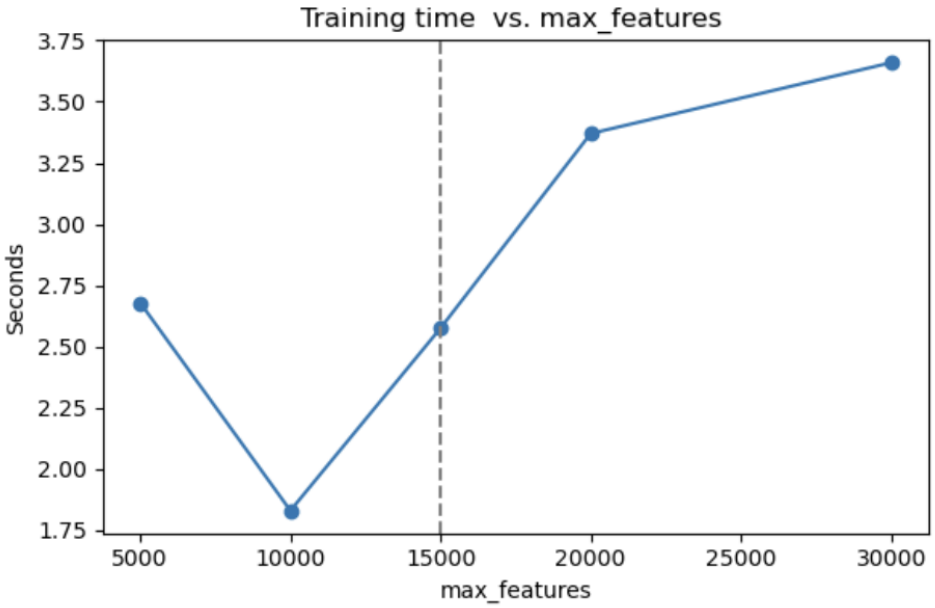

Figure 5.5 Training Time Vs Max_Features

This graph below presents an illustration of the number of times that each token has been utilized when only the top-N most often occurring phrases are retained. Due to the rapid ascent of the curve, it can be deduced that a vocabulary that is not excessively extensive already encompasses the majority of the terms contained in the dataset. At this point, there are approximately 15,000 tokens that contain more than 97% of all word occurrences. Increasing the number of tokens only results in marginal gains. By utilizing 15,000 as the TF-IDF feature limit, it is possible to provide satisfactory coverage without overly complicating the situation.

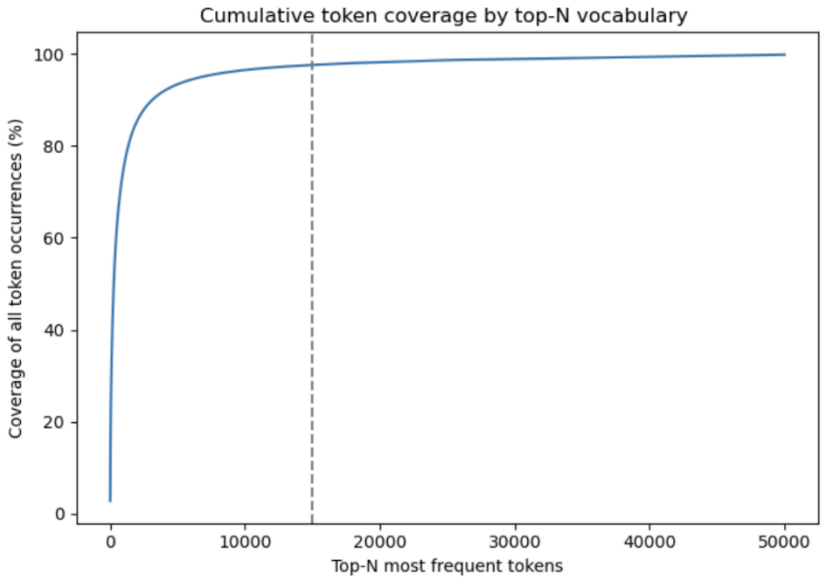

Figure 5.6 Token Coverage by Top N Vocabulary

To balance accuracy, efficiency, and vocabulary coverage, 15,000 attributes it has been chosen. With more features, the macro F1 score increased, but it peaked around 15,000, indicating diminishing returns. Training time increased faster with a bigger vocabulary, which could delay testing. The cumulative token coverage curve showed that 15,000 attributes correctly detected over 97% of word occurrences. Most of the dataset's key data was saved. This point balanced good predictive performance, a quick training duration, and a compact representation, making it an acceptable cutoff for the TF-IDF feature space.

5.3.2 Part of Speech Features (POS)

Through the extraction of Part-of-Speech (POS) features, the NLTK tagger was used to achieve the task of counting the nouns, verbs, adjectives, and adverbs that were present in each review. These are the categories that decided to employ since, in general, they correspond to significant linguistic functions in the manner in which individuals express their emotions. Adjectives and adverbs often carry a significant amount of emotional weight (for instance, "amazing" and "terribly"), while verbs and

nouns provide context and discuss actions or objects. Through the process of counting these items, the model is able to provide structural and grammatical insights that are compatible with TF-IDF. One example is that a review that has a large number of adjectives and adverbs is more likely to demonstrate strong views than a review that contains a large number of nouns. For the purpose of ensuring that the extracted numbers were consistent among of varying durations, it has been used the MinMax scaling method.

The graph below illustrates how the average number of part-of-speech (POS) categories is distributed throughout the various categories of sentiment. Nouns constitute the largest portion of all of the groupings. Nouns are more prevalent in reviews that are neutral, but adjectives are more prevalent in reviews that are either positive or negative, indicating a more descriptive or emotional tone from the reviewer. Although adverbs are still the least prevalent form of word, verbs seem to be used in a somewhat similar manner across the various classes. It may be deduced from these differences that point-of-sale (POS) patterns have the potential to provide useful indications for sentiment analysis, especially adjectives and adverbs, which often verbalize express emotion openly.

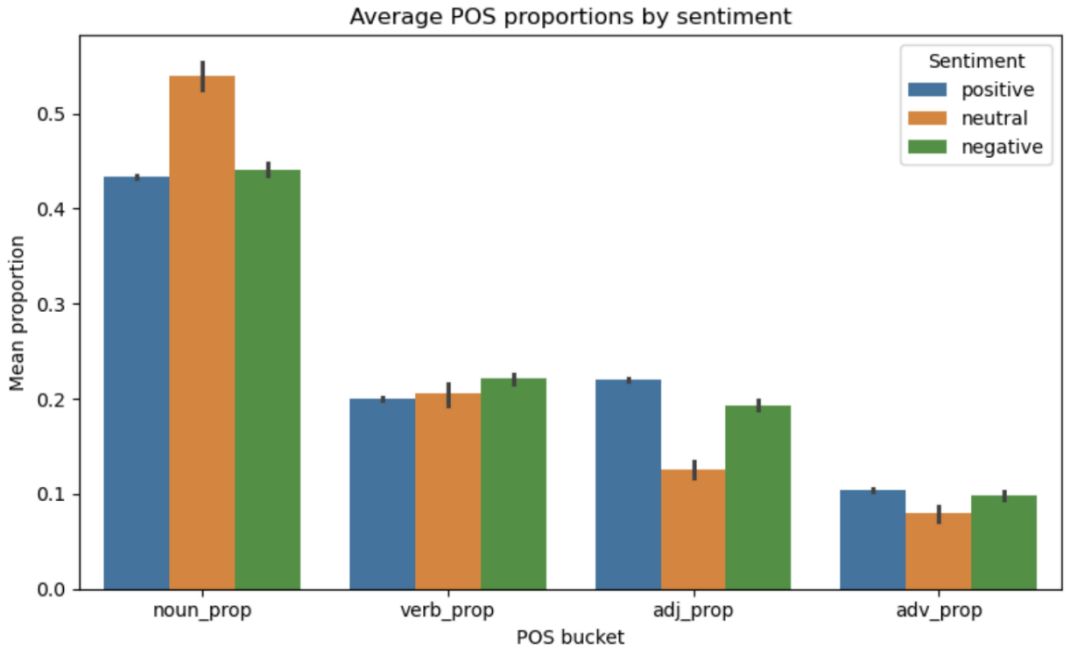

Figure 5.7 Average POS by Sentiment

The table below illustrates how different aspects of speech manifest themselves in a manner that corresponds to varied emotions. There are more nouns in neutral evaluations, which gives the impression that they are more thorough. Evaluations that are negative tend to have a greater number of verbs and adjectives, which usually demonstrate actions or complaints. There is a greater use of adjectives and adverbs in positive reviews, which indicates that the reviewers are expressing a greater degree of emotion. By taking into account these variations in the structure of language, the model is able to more accurately identify patterns of emotion.

Table 5.4 Average POS Proportions Across Sentiment Classes

Sentiment	Noun	Verb	Adjective	Adverb
Negative	0.441	0.221	0.193	0.098
Neutral	0.539	0.205	0.125	0.079
Positive	0.433	0.2	0.220	0.104

5.3.3 Review Feature

For the purpose of this investigation, two fundamental characteristics were utilized: the total number of characters and the total number of words included in each review. According to these metrics, a critic provides a certain amount of information. Most of the time, shorter evaluations are characterized by fast assessments, but lengthier evaluations often include more context and a higher number of emotional clues. The addition of these properties enables the system to differentiate between brief, general input and more ideas, which ultimately results in improved sentiment classification. Table below shows how is the output of this feature from Python.

Table 5.5 Example of Review Feature

Review	Review length words	Review length chars	Word count bin
met expectations	2	16	0-19
excellent	1	9	0-19
work give stars know cheap least expected week	8	46	0-19

5.3.4 Polarity features

Polarity features were added using the Opinion Lexicon from NLTK, which includes lists of words typically linked to positive or negative sentiment. For each review, counts were made of how many positive and negative terms appeared, creating two new features: `lex_pos` and `lex_neg`. These values give a direct indication of how much emotional vocabulary is present in a review. The analysis showed clear differences between classes: positive reviews contained more positive word hits on average about 4.2, while negative reviews included more negative terms around 2.6 as shown in the table below. Neutral reviews, as expected, had lower counts for both. This pattern highlights the usefulness of lexicon-based features, particularly in short reviews where individual sentiment words often dominate the expression.

Table 5.6 Average Positive and Negative Lexicon Counts by Sentiment Class

Sentiment	Positive	Negative
Negative	1.136	2.671
Neutral	0.551	0.373
Positive	4.245	1.337

5.3.5 Chi-Square (χ^2) feature selection

The Chi-Square (χ^2) test was used as a way to choose features to improve the TF-IDF representation by keeping just the phrases that were most closely related to

sentiment categories. This statistical method looks at how words and their class labels are related to each other. It makes sure that tokens that aren't important or aren't strongly related are thrown away. Using χ^2, the number of features in the TF-IDF matrix was cut down from 15,000 to 5,000. This let the model focus on the most important signals. This cut not only made the training process faster, but it also made it less likely that the classifier would overfit because it focused on features that were more clearly special. Performance study showed that accuracy had mostly leveled off with 15,000 features. Using χ^2 allowed us to get equivalent prediction strength with a smaller, more efficient collection. In this way, χ^2 became a key part of the modeling process that helped find the right balance between speed and accuracy.

The next chapter will be about making sure the parameters chosen in this stage are correct. This section talked about setting TF-IDF dimensions and used χ^2 feature reduction. Chapter 6 will look at how they affect performance by using metrics like accuracy, precision, recall, and F1-score. This phase is necessary to make sure that the chosen parameters are not only effective, but also help the models stay stable and flexible when they view new data.

5.4 Handling Class Imbalance

The dataset had a strong bias towards positive evaluations at first, with around 88% of entries marked as favorable. Neutral and negative reviews made up only a minor part of the data about 4.5% and 7.9. This kind of mismatch can make machine learning algorithms quite likely to guess the majority class. To fix this problem, an under-sampling method was used. The final dataset was balanced so that there were around 32,000 records for each sentiment as shown below. This change made the training field more level, which cut down on bias and made evaluation metrics like precision, recall, and F1-score more reliable for all classes.

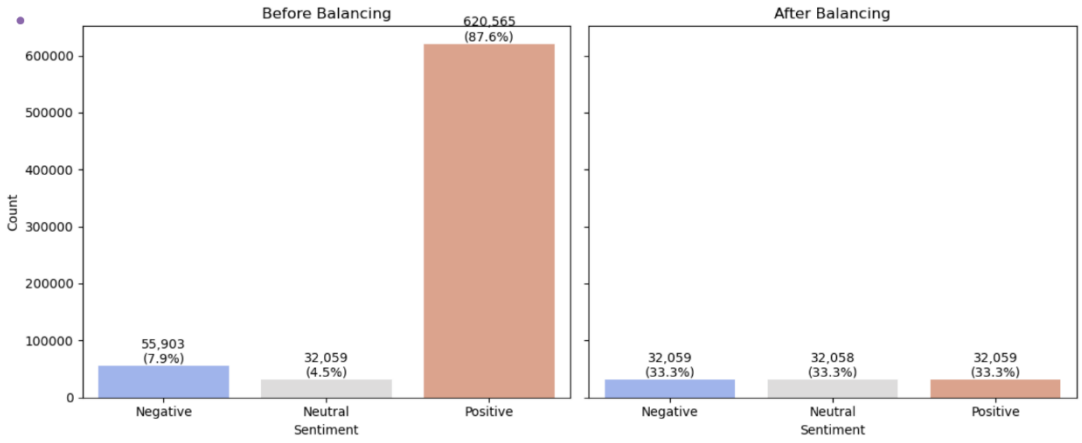

Figure 5.8 Before and After Balancing

5.5 Model Development

Simple baseline models, such as lexicon-based models or word frequency–based models (for example, majority-class prediction or TF-IDF), were the first approach to model development. These models were covered in the chapter literature review. The original strategy to model development was the introduction of these simple baseline models. The researcher was able to assess the incremental advantages that were produced through the deployment of more sophisticated machine learning because they had such baselines.

5.5.1 Machine Learning

using standard machine learning strategies in order to construct models for sentiment analysis. This was accomplished by transforming text documents into high-dimensional numerical vectors using a technique known as TF-IDF, which stands for term frequency–inverse document frequency with other features that been discussed previously. The algorithms that were evaluated for this study were the Support Vector Machine (SVM) and Naïve Bayes (NB). Using standard measures like as accuracy, precision, recall, and F1-score on a given test set, we analyzed the performance of each and every model.

5.5.2 Support Vector Machine Classifier

Support Vector Machines (SVM) are a well-known way to sort text into classifications. This is mostly because it works well with data that is sparse and has a lot of dimensions. LinearSVC has been chosen for implementation in scikit-learn for this project. It is made to work well with big datasets that are shown in chapter 6 as TF-IDF vectors. Kernel-based SVMs, such the RBF kernel, weren't useful here since they require a lot of computing power when there are tens of thousands of documents and they don't always give greater accuracy for text analysis.

Table 5-7 Classifier Choice

Aspect	Details
Algorithm	SVM
Implementation	LinearSVC
Reason of Choice	Works well with Spare and high dimensional data
Karnel Selection	Linear Karnel

The model was trained on a composite feature space that has four parts: TF-IDF vectors that were improved using Chi-Square feature selection (k = 5,000), part-of-speech counts, review meta information (such word and character length), and polarity lexicon counts. This combination of lexical, structural, and stylistic information let the classifier tell the difference between positive, negative, and neutral evaluations.

The regularization constant (C) was one of the most important parameters that was looked into. Tests with values like 0.5, 1.0, 2.0, and 5.0 showed that C = 1.0 offered the most dependable results since it struck a good balance between accuracy and generalization. The model preserved the default hinge loss function, and the dual optimization setting stayed True. This is the best configuration to use when the feature set is more than the number of samples, which is common for text data.

Table 5-8 Features Used

Features	Description
TF-IDF with chi-Square	5000
POS Counts	Noun, Verb, Adjective, Adverb
Review Feature	Word length, letters length, word size
Polarity	Counts of Positive, Negative

Table 5.7 Hyperparameter Settings

Parameter	Value
Regularization	1
Loss Function	Hinge
Dual Optimization	True

Through the use of a stratified 80/20 train-test split, able to guarantee that the class proportions remained consistent throughout both sets. In addition, k-fold cross-validation was used in order to guarantee that the outcomes were consistent. The quality of the model might be evaluated using a variety of methods, such as accuracy, precision, recall, F1-scores, and confusion matrices. Because of this, it became feasible to evaluate performance across all of the other categories of sentiment, rather of only focusing on the one that was most prevalent.

Table 5.8 Training and Validation Setup

Aspect	Details
Data Split	Stratified 80/20 train-test split
Cross Validation	k-fold CV used to ensure consistent performance
Evaluation Metrics	Accuracy, Precision, Recall, F1-score, Confusion Matrix

5.5.3 Naïve Bayes Classifier

One of the most often used algorithms is the Naïve Bayes technique, which is fast and simple to apply. This work used the Multinomial Naïve Bayes (MNB) variant, since it is proficient in handling frequency-based attributes such as the TF-IDF distribution. This assumption enables the model work quickly and well for large document collections, even though this is not usually the case in real life. The method assumes that characteristics are conditionally independent, which is not often the case in real life.

The model was trained on a combined feature set that included part-of-speech counts, review duration indications, and polarity lexicon features. Additionally, the model was trained on TF-IDF vectors that were reduced using Chi-Square feature selection (k = 5,000). Due to the fact that there was a combination of lexical and structural signals, the classifier took into consideration both the existence of words and the structure of the reviews.

Table 5.9 Classifier Choice

Aspect	Details
Algorithm	NM
Implementation	MNB
Reason of Choice	Quick, easy, and useful with features depending on frequency, like TF-IDF
Karnel Selection	Linear Karnel

When it comes to MNB, the smoothing factor (α) is the most crucial element to consider. The probability of a word appearing in test data but not in training data is prevented from being zero as a result of this functionality. A variety of alternative values were examined, but α = 1.0 proved to be the most suitable option since it provided with consistent outcomes without causing either overfitting or underfitting.

An 80/20 stratified split was used to maintain class balance, and cross-validation was used to ensure that the findings were consistent. The evaluation technique was set up in the same manner as the support vector machine (SVM). Quantifying performance was accomplished by the use of confusion matrices, accuracy, precision, recall, and F1-score.

5.6 Dashboard Development with Power BI

In order to make the findings of the sentiment analysis more accessible to end users, an interactive dashboard has been developed using Power BI. This dashboard displays the results of the analysis. The home page, the overview, the brand analysis, and the word cloud were the four most important components of the design. There was a distinct analytical goal for each individual component.

The first page of the report that you will read is called the Home page. It provides with buttons that allow to rapidly navigate to the different components of the dashboard and explains its purpose. Users are able to transition between high-level summaries and more in-depth examinations with ease because to this arrangement.

Once the dashboard opened will get a clear glimpse of three key components, which work together to give this information. There is a high-level viewpoint shown in the Overview, which includes the total number of reviews and negative reviews, the average rating, and the verification data for reviews. In addition, it is equipped with key performance indicator indications, which make it simple to comprehend. When use the Brand Analysis, able to compare different brands and observe how their ratings and sentiments vary from those of their competitors. This is accomplished via the use of bubble charts and tables. Additionally, it enables to go into more detail. Last but not least, the Word Cloud displays the terms that appear the most often in assessments, making it easy to perceive recurring topics such as "case," "love," and "phone." Because of the combined design, it is simple to recognize both the major trends and the minute details.

Due to the fact that it combines the results of machine learning with a straightforward visual presentation, the dashboard is an effective tool for decision-making procedures. Managers of businesses are able to keep an eye on trends of sentiment, identify businesses that are not performing well, and determine what customers care about the most with the aid of this capability. Also, it makes it simpler for stakeholders to understand how other people feel about a product that they are considering purchasing.

5.7 Chapter Summary

It was detailed in this chapter how the sentiment analysis models were constructed, beginning with the preparation of the dataset and the creation of features and continuing on to the management of class imbalance and the training of the classifiers. For the purpose of constructing both Support Vector Machine and Naïve Bayes, it has been used TF-IDF coupled with chi-square selection and several additional feature sets. In addition, it demonstrated how they were constructed as well as how to assess them. The chapter also discussed the dashboards that were created using Power BI. These dashboards were designed to make the results of the analysis simple to comprehend. next this, the outcomes of these models and visualizations will be presented and analyzed in the next chapter.

CHAPTER 6

RESULT AND DISCUSSION

6.1 Introduction

The purpose of this project was to investigate the feasibility of applying machine learning to the task of analyzing the sentiments that are expressed in Amazon product reviews. Because of these three primary goals, the work was influenced. The first thing that was done was exploratory data analysis (EDA), which looked for patterns in the dataset. These patterns included how reviews are distributed across various sentiment categories, which keywords appear the most frequently, and how review length changes over time. The subsequent phases of the study were significantly influenced by these observations, which led to their significance. The second objective was to learn and evaluate machine learning models that could categories reviews as either positive, neutral, or negative, and then to determine which model produced the most accurate results. Creating an interactive dashboard that could display the analysis in a way that was easy to understand and assist individuals in drawing insightful conclusions about the behavior of customers was the final objective.

As a result of these objectives, the remainder of this chapter will demonstrate what transpired. To begin, the models are put through a series of tests that demonstrate how well they function. These tests include learning curves, confusion matrices, and standard metrics. In the following step, the influence of feature engineering is investigated by contrasting the outcomes of the models both before and after the addition of new specifications. Dashboards that display the overall sentiment trends and brand-level insights are discussed in this chapter, in addition to the technical results that are presented elsewhere in the chapter. In conclusion, the findings are analyzed in relation to the research problem, and they are linked to studies that have been conducted in the past. After the conclusion of the chapter, there is a summary that leads into the following chapter.

6.2 Model Evaluation Results

For the purpose of evaluating the sentiment classifiers, it has been utilized a variety of different approaches. Learning curves were utilized in order to demonstrate how the performance of each model shifted due to the addition of additional training data. Confusion matrices were then utilized in order to demonstrate the areas in which the models were able to make accurate predictions and the areas in which they confused one sentiment with another. In the end, standard metrics such as accuracy, precision, recall, and F1-score were utilized in order to provide a more comprehensive understanding of the overall performance of the system being evaluated. When combined, these approaches provide a comprehensive and comprehensive picture of how well the models were able to deal with new reviews that they had not previously encountered.

6.2.1 Learning Curves

In order to observe how the models evolved as they were exposed to additional training data, learning curves were utilized. It was possible to evaluate the generalization capability of each classifier on data that had not been seen before and to identify indications of overfitting or underfitting by displaying both the training and validation F1-scores.

Figure 6-1 shows the learning curve for the Linear SVC model. The model got an F1 score that was almost perfect on the training data, close to 1.0, when the training subsets were smaller. The validation score, on the other hand, stayed much lower. This gap shows that the model tended to fit too well when it was given only a small amount of data. The training score went down a little as more samples were added, but the validation score went up steadily. The validation F1-score had gone over 0.80 and was very close to the training score, which leveled off at about 0.83, by the time the training size reached about 70,000 reviews. The two curves getting closer together shows how the model became more stable and worked better with more data.

Overall, the curve shows that Linear SVC is a good choice for this classification task as long as it is trained on enough data. It generalizes well once there is enough data, with only a small and acceptable difference between training and validation performance.

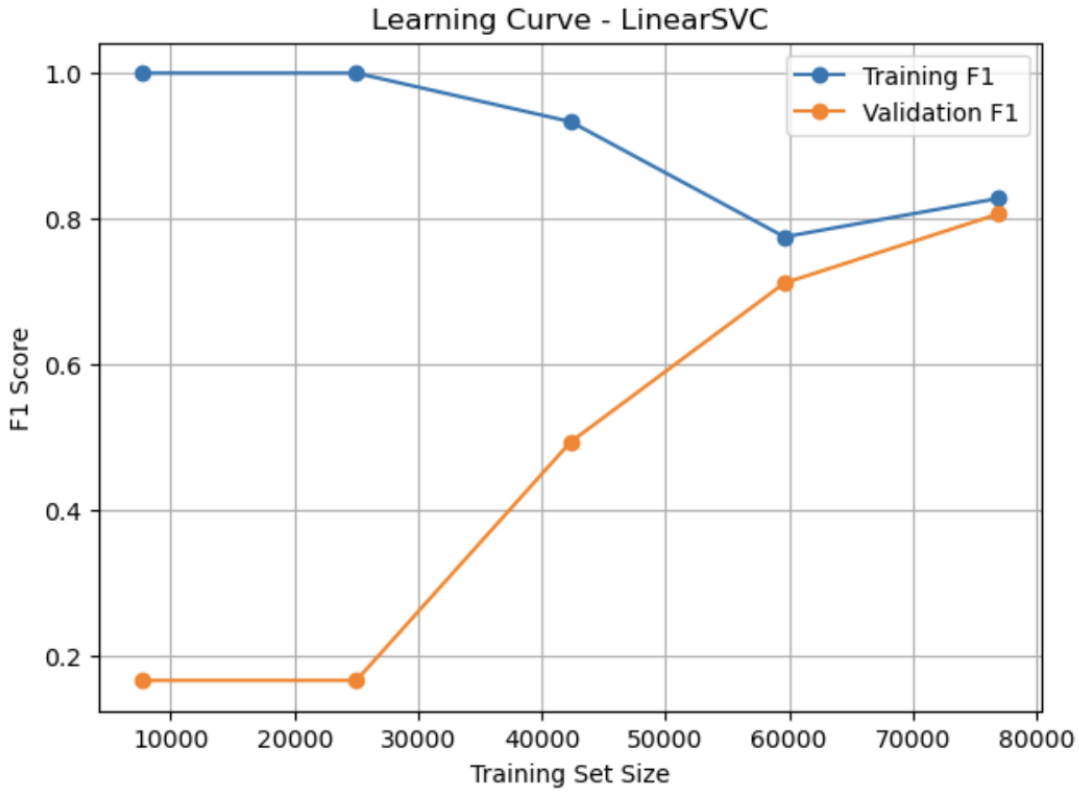

Figure 6.1 Learning Curve for SVM Classifier

Figure 6-2 illustrates the learning curve for the Multinomial Naïve Bayes model, which is similar to SVC in its early stages. The model had an almost perfect F1 score on the training set but a very low validation score, indicating overfitting due to a lack of training data. While the training score decreased as the training size increased, the validation scores steadily increased, reaching over 0.81 at 70,000 reviews. With more data, the model generalized better, almost eliminating the difference between training and validation performance. Compared to Linear SVC, Naïve Bayes reached its peak performance earlier and had a lower validation score. It was a good baseline model because it performed well and was easier and faster to train.

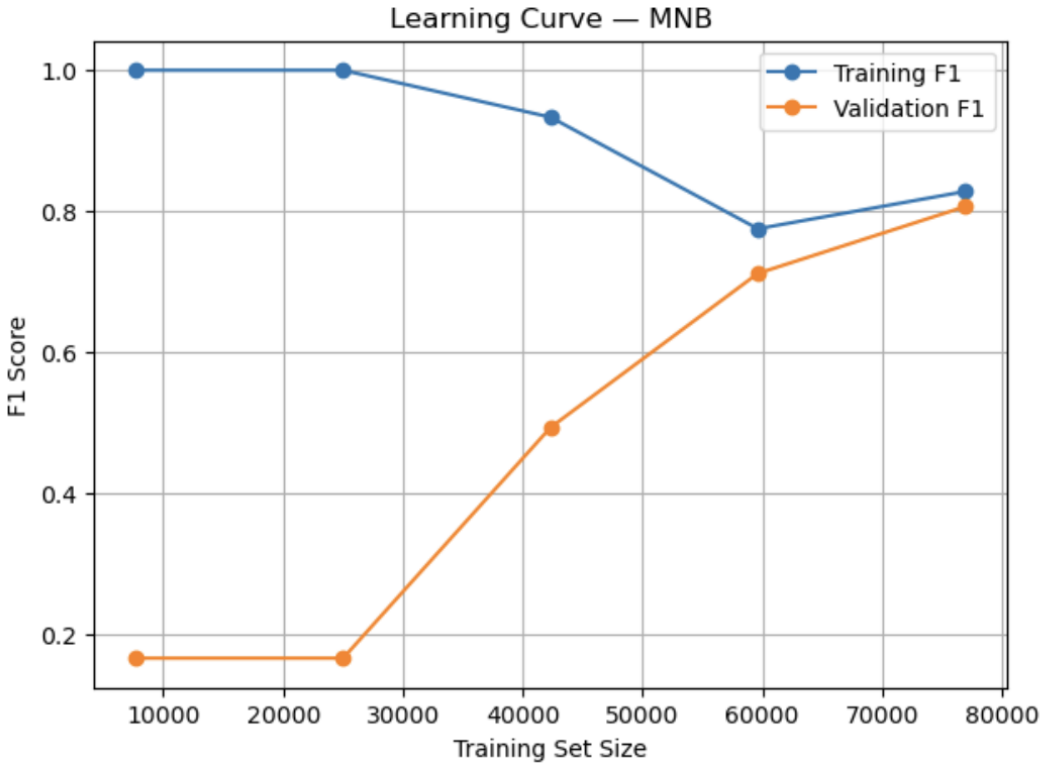

Figure 6.2 Learning Curve for MNB Classifier

6.2.2 Confusion Matrices

For the purpose of determining how well the classifier could differentiate between positive, negative, and neutral reviews, we created confusion matrices for both the training set and the test set. In comparison to the summary metrics, these graphs paint a more accurate picture of the performance of the class as a whole.

When applied to the training set, the Linear SVC achieved an extremely high level of accuracy across all three categories of sentiment Figure 6.3. It was determined that there were 24,740 reviews that were negative, 23,968 reviews that were neutral, and 25,101 reviews that were positive. Some reviews were incorrectly classified as either positive or negative, particularly in the group that was neutral. This was one of the few errors that occurred. In spite of this, the training data continued to be extremely accurate, and the model did not adhere to the unattainable pattern of classification perfection. In other words, it did not simply memorize the information rather, it learned it effectively.

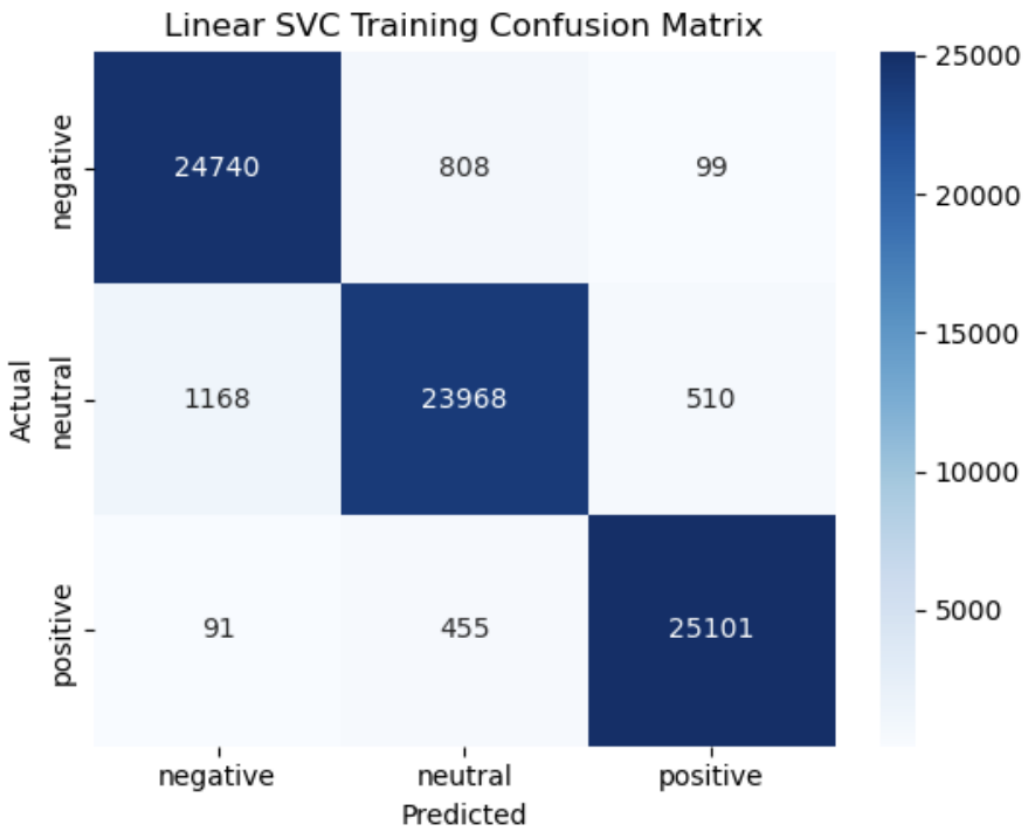

Figure 6.3 Training Confusion Matrix for SVM

A pattern that was very similar to the one that was found in the training set was observed in the test set Figure 6.4, which is a positive indication that generalization has occurred. There was a total of 6,397 reviews that were negative, and the model accurately predicted 6,116 of those reviews. On the other hand, only a few were confused with the other classes. There were 5,966 reviews that went into the neutral category, which was the correct classification. Among the 316 that were poor, there were 130 that were good. Due to the fact that 6,184 reviews were also correctly predicted in the positive class, there were only a few errors. Considering that these results are comparable to what observed during the training phase, it can be concluded that the model continued to be accurate when we tested it on new data.

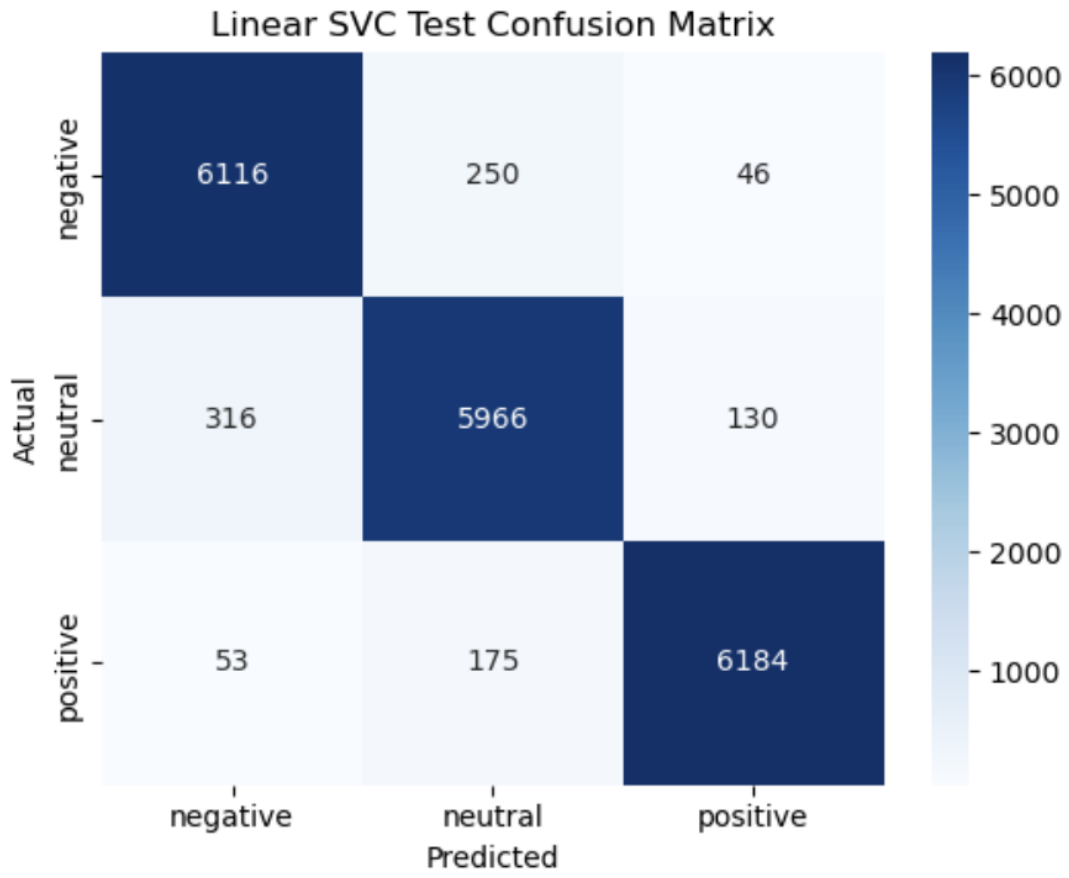

Figure 6.4 Testing Confusion Matrix for SVM

The confusion matrices show that the Linear SVC worked the same way in both the training and the testing. Most of the mistakes came from reviews that were neutral. This makes sense because neutral language and language that is slightly positive or negative often overlap. This strengthens the idea that the model's predictions are both correct and possible.

Moving to Figure 6-5 training and Figure 6-6 testing illustrate the confusion matrices that are generated by Multinomial Naïve Bayes. With regard to the training set, the model did an excellent job of dealing with both positive and negative feedback. It provided an accurate classification of 23,555 negative reviews and 23,680 positive reviews. However, the neutral category presented a greater challenge. The number of neutral reviews that we were able to correctly identify was 16,931. However, a significant number of them 4,140 were incorrectly labeled as negative, and a

significant number of them 4,575 were incorrectly labeled as positive. The fact that this is the case demonstrates that the model was able to determine how people generally felt, but it struggled with reviews that were inconsistently positive or negative.

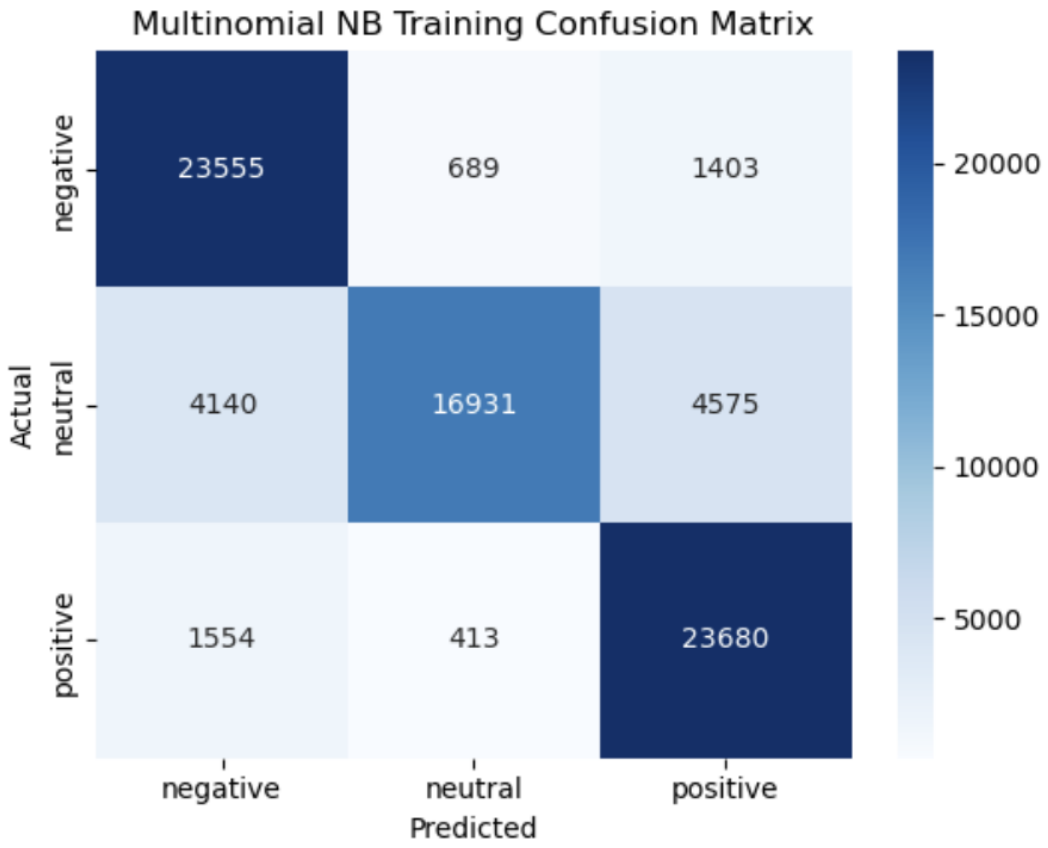

Figure 6.5 MNB Training Confusion Matrix

The test set exhibited a pattern that was comparable as well. Five thousand eight hundred and thirty-seven negative reviews were correctly predicted, while only a small number were incorrectly classified. With 5,874 correct guesses, the positive class performed exceptionally well once more. However, dealing with reviews that were neutral was a more difficult task. In total, only 4,083 were correctly labeled, while 1,079 were incorrectly labeled as negative and 1,250 were incorrectly labeled as positive. Taking into account these findings, it is clear that the most significant challenge that MNB faces is determining how to differentiate between reviews that are neutral and those that are extremely polarized.

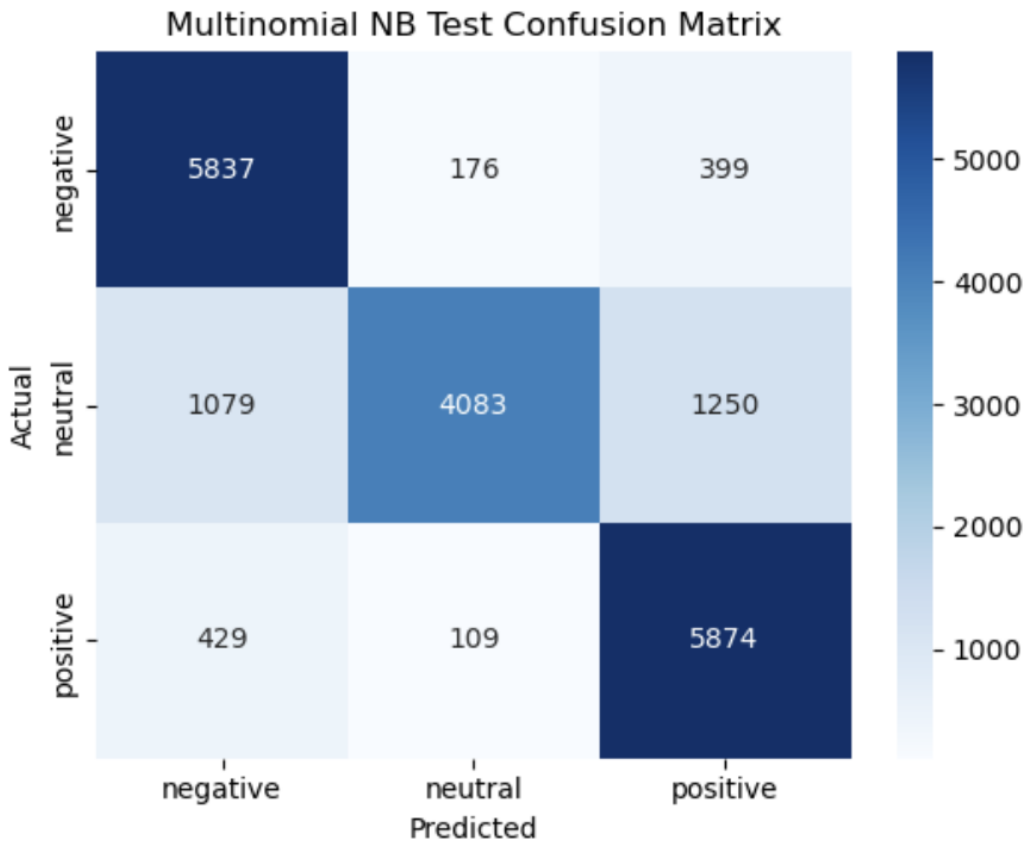

Figure 6.6 MNB Test Confusion Matrix

The confusion matrices show what Naïve Bayes is good at and what it isn't. This model is simple and works well for reviews that are clearly good or bad, but it doesn't always work for reviews that are neutral. This makes it less useful than SVC, but it still sets a high bar for competition and shows that simpler models can work well with less computing power.

6.2.3 Performance Metrics

In order to evaluate the effectiveness of the models, it has been utilized a collection of standard performance measures. Accuracy is the percentage of predictions that were accurate, precision is the degree to which the predictions were reliable, recall is the degree to which the model was able to locate the actual examples of each class, and the F1-score can be thought of as a number that combines precision

and recall into a single number. it can be determined how well the models learned and whether or not they were able to apply what they learned to new data by looking at these metrics on both the training set and the testing set.

When it came to both the training and the testing, the Linear SVC model performed exceptionally well. The primary findings are presented in Table 6.1. On the training set, the model achieved an accuracy of nearly 96%, and the values for precision, recall, and F1 were all extremely close to one another. It is clear from this equilibrium that the model did not favor any particular category over any other and was able to correctly categorize reviews into three distinct groups: negative, neutral, and positive. In spite of the fact that it has been used the test data, performance remained almost identical, with a slight decrease of less than one percent. The fact that this difference is barely noticeable demonstrates that the Linear SVC was able to discover patterns that were not present in the data that it was trained on. Therefore, it is of the utmost importance that these results remain unchanged. Overfitting is a problem that plagues many models, which means that they perform well on training data but not so well on new reviews. It is not the case here. The Linear SVC has not changed, which indicates that it is not only accurate but also dependable in the real world.

Table 6.1 Performance Metrics for Linear SVC

Dataset	Accuracy	Precision	Recall	F1-Score
Training	0.959	0.959	0.959	0.959
Testing	0.950	0.950	0.950	0.950

The results that the Multinomial Naïve Bayes model produced were not as good as those that the SVC model produced; however, it did exhibit clear learning and reliable generalization throughout the process. Table 6.2 contains the results, which you can view. With an accuracy rate of approximately 83% and an F1-score of 0.830, Naïve Bayes performed exceptionally well on the training set.

On the other hand, the model was somewhat more accurate than it was recalling, which indicates that it was more likely to exercise caution when making predictions and to avoid producing false positives, even if it failed to identify some actual cases. An accuracy of 82.1% and an F1-score of 0.816 were achieved when the results were tested on the test data. The results experienced only a slight decrease. Despite the fact that SVC performed better overall, the fact that there was a slight difference demonstrates that Naïve Bayes was able to generalize well.

Upon analyzing the model's performance across all classes, we discovered that the category of neutral sentiment was the one in which it struggled the most effectively. Numerous reviews that were neutral were incorrectly categorized as either positive or negative.

The occurrence of this phenomenon is quite common in simpler models such as Naïve Bayes, which are heavily dependent on word frequencies and may not be able to detect even minute variations in meaning. However, the model continued to function effectively for reviews that were either obviously positive or obviously negative. A further advantage of Naïve Bayes is that it does not consume a significant amount of computer power.

Table 6.2 Performance Metrics for Multinomial Naïve Bayes

Dataset	Accuracy	Precision	Recall	F1-Score
Training	0.834	0.848	0.834	0.830
Testing	0.821	0.837	0.821	0.816

Both models learned from the dataset and used what they learned to make new reviews, but with different degrees of success. The Linear SVC was best because it was almost 96% accurate and performed well on all metrics. This proved its suitability for strong sentiment classification. While the Naïve Bayes model was less accurate, it provided consistent results and was a good starting point. Because it is quick and easy to use, it can be used when speed is more important than accuracy.

6.2.4 Comparative Analysis of Classifiers

When the two models are compared to one another, it is simple to see how well they performed on the dataset. In this section demonstrate how accurate the models were in general, but they also demonstrate how well they did in certain classes and points out where they were incorrect.

With regard to their overall performance, the two classifiers are compared in Figure 6.7. In terms of performance, the Linear SVC model outperformed the Naïve Bayes model, which achieved a mere 82%. This distinction demonstrates that SVC is superior to other methods in terms of recognizing patterns in data and making more precise prediction.

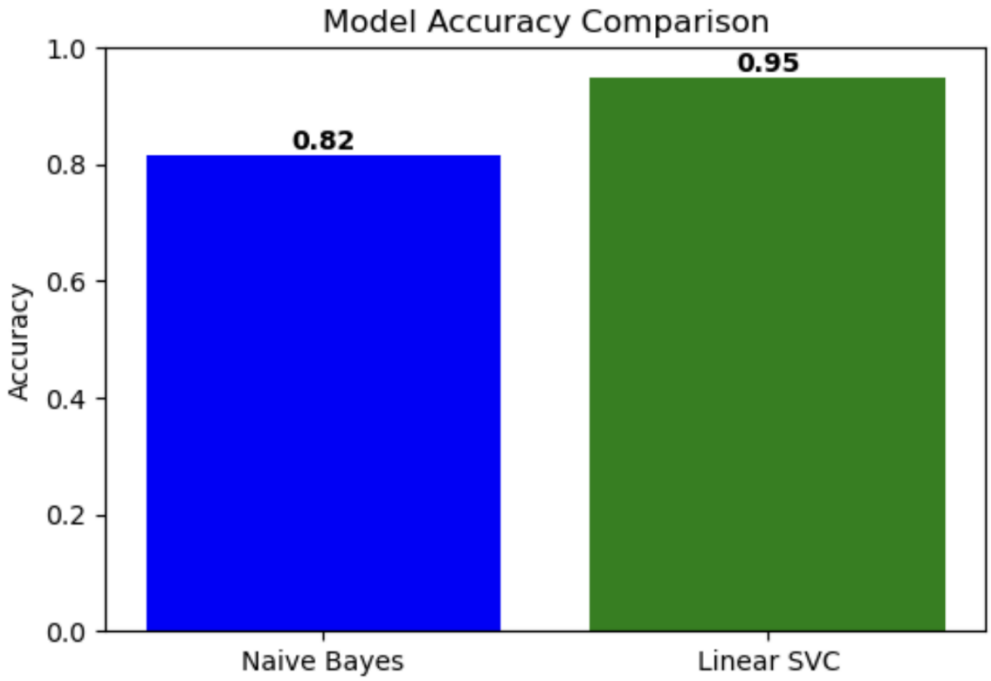

Figure 6.7 Models Accuracy Comparison

A comparison of the different types of sentiments is shown in Figure 6-8. The two models performed admirably when it came to both positive and negative reviews; however, there is a discernible distinction when it comes to reviews that were neutral. The Linear Support Vector Machine (SVC) exhibited a significantly higher F1-score for the neutral class. However, Naïve Bayes frequently erroneously incorrectly

identified neutral feedback as either positive or negative. The category of neutral sentiment is the most difficult, and SVC is clearly superior in this regard.

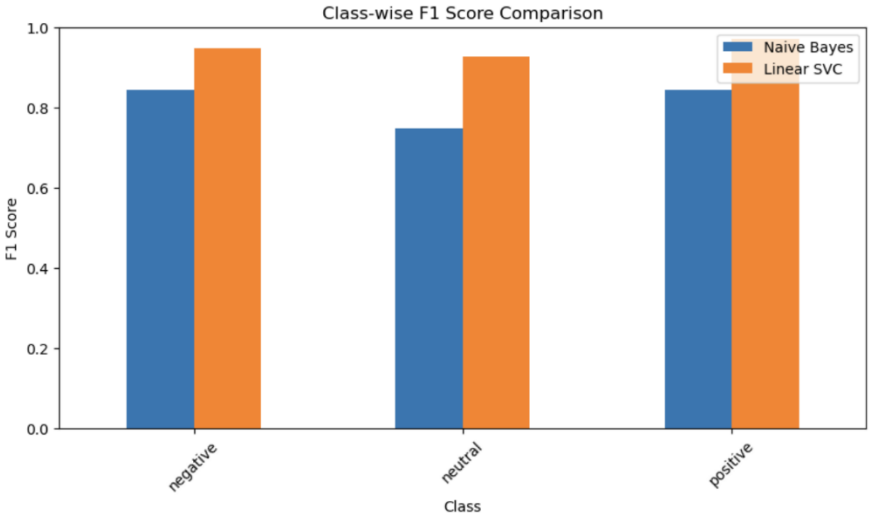

Figure 6.8 Class-wise F1 comparison

The analysis of disagreement that is presented in Figure 6.9 provides us with a fresh perspective on the situation. According to the figure, the SVC algorithm was more accurate than the Naïve Bayes algorithm. This indicates that SVC consistently exhibited the ability to identify patterns that Naïve Bayes struggled to identify, thereby enhancing its credibility and reliability.

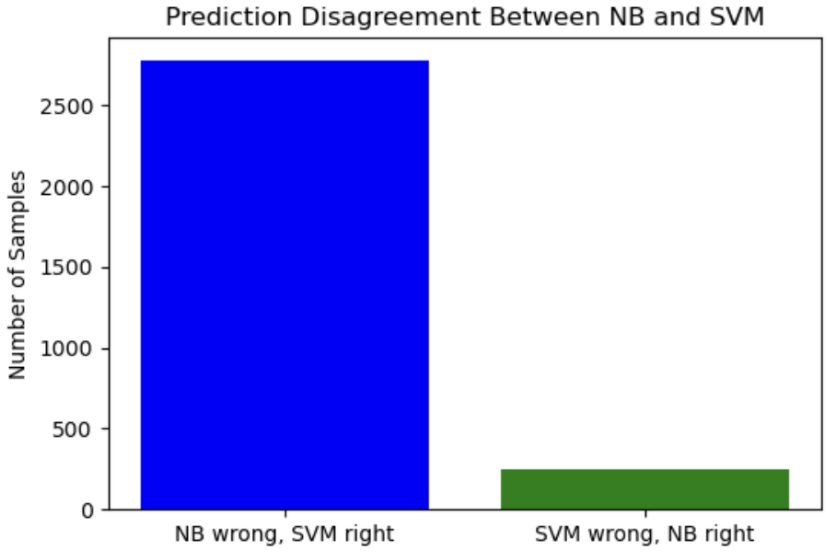

Figure 6.9 Prediction Disagreement between the classifiers

The Linear SVC model is the most suitable option for this investigation, according to the findings of the evaluation. Both the training set and the test set produced the same results, with the test set achieving the highest accuracy (95%) and the highest macro F1-score (0.95). This indicates that it was able to generalize well without necessarily fitting the data in an excessively close manner. The Linear SVC performed admirably over a wide range of reviews, regardless of whether they were positive, negative, or neutral. On the other hand, the Naïve Bayes algorithm struggled when working with reviews that were neutral. It was demonstrated by the learning curves that the model performed more effectively with a greater quantity of training data and reached a consistent high level of performance. Naïve Bayes trained more quickly and had a fairly high accuracy rate of 82%; however, it frequently makes mistakes with neutral reviews, which makes it less useful in this particular scenario. On the other hand, linear SVC is the superior option for programs that need to be extremely accurate and have a method of classifying feelings that is reasonably balanced.

6.3 Feature Engineering Contribution and Analysis

One aim of this study was to find out if adding more complex features to the text representation could make the model work better. To achieve this, the classifiers were initially trained solely with TF-IDF vectors. In the second stage, there were more signals added, like part-of-speech (POS) counts, review meta-features like text length, chi^2, and polarity lexicon counts that show how often positive and negative words are used. The outcomes from both configurations clearly demonstrate the impact of feature engineering on each model.

6.3.1 Multinomial Naïve Bayes (MNB)

The Naïve Bayes classifier had an accuracy of 0.80 and a macro F1-score of 0.80 before feature engineering. The model did well with both negative and positive reviews, getting F1-scores of 0.81 and 0.83, respectively. But it had trouble with the

neutral class, which dropped to 0.74. This showed that it mixed up neutral reviews with reviews that were more polar.

The performance improved slightly following feature engineering. The accuracy rose to 0.82, and the macro F1-score rose to 0.816. The neutral class had the most changes, with F1 going from 0.74 to 0.76. The change may not seem like much, but it shows that adding more signals, like polarity lexicons, helped the model understand reviews that were hard to read.

Table 6.3 Naïve Bayes Performance

Metric	Before	After
Accuracy	0.80	0.82
F1	0.80	0.816
Negative F1	0.81	0.85
Neutral F1	0.74	0.76
Positive F1	0.83	0.84

The new features definitely made Naïve Bayes more powerful. Adding structural and sentiment-based cues to word frequency patterns made the model more reliable overall. This reduced the number of neutral cases that were incorrectly classified.

6.3.2 Linear SVC

The Linear SVC already did very well before feature engineering, with an accuracy of 0.95 and a macro F1-score of 0.95. The results for each class were well-balanced: 0.96 for negative, 0.93 for neutral, and 0.97 for positive.

After feature engineering, the overall accuracy and macro F1-score stayed the same at 0.95. The headline numbers didn't get better, but the model's predictions became a little more stable, especially for the neutral class. The new features made it easier to tell the difference between neutral and polar categories, which made classification more consistent across all levels of sentiment.

Table 6.4 Linear SVC Performance

Metric	Before	After
Accuracy	0.95	0.95
F1	0.95	0.95
Negative F1	0.96	0.95
Neutral F1	0.93	0.93
Positive F1	0.97	0.97

The extra features didn't help SVC's overall score because it already had a good learning ability with TF-IDF. They were helpful because they kept things balanced between categories and stopped small drops in performance from unclear reviews.

Overall, the results show that feature engineering was worth it, even though the benefits were different for each model. Naïve Bayes made the most progress, especially in accuracy, F1, and sorting neutral reviews. Linear SVC, on the other hand, was already working almost perfectly with TF-IDF, so the extra features made the system more stable rather than more accurate. This shows that adding engineered features to sentiment analysis pipelines can make them stronger and easier to understand, even if the performance gains are small.

6.4 Visualization of the Dashboards

In order to provide support for the findings of the machine learning, a number of interactive dashboards were created with the help of Power BI. These dashboards do more than just display the results of sentiment classification. In addition to that, they make it simpler for business users and decision makers and also for the stakeholders to comprehend the results. There is a distinction between a static report and the dashboards because the dashboards are interactive. Additionally, users have the ability to filter the data based on brand or sentiment, and they can also use a question-and-answer assistant to directly ask questions about the data. Through the utilization of both clear visuals and analytical results, it becomes much simpler to

transform technical outputs into insights that are more beneficial and can be put into action. The following few sections provide an in-depth analysis of each dashboard, focusing on its primary characteristics as well as the categories of information that it provides.

6.4.1 Home Dashboard

The report begins and the analysis is compiled on the home dashboard. It simplifies access to the Overview and Brand Analysis. This lets users browse the results without feeling overwhelmed. This way, the dashboard is a clear starting point for finding the information users want.

A highlight of the home page is the interactive Q&A assistant. Questions can be typed in English and answered immediately from the dataset. If someone asks, "Which brands have the most positive reviews?" or "What is the average sentiment score for all products?" the tool will immediately display a chart or number. The dashboard has more options and stakeholders can view data in ways that matter to them.

Easy navigation and a Q&A feature make the home dashboard more than a static menu. It becomes an interactive tool for techies and non-techies. This page simplifies data analysis for non-data analysts who need quick insights.

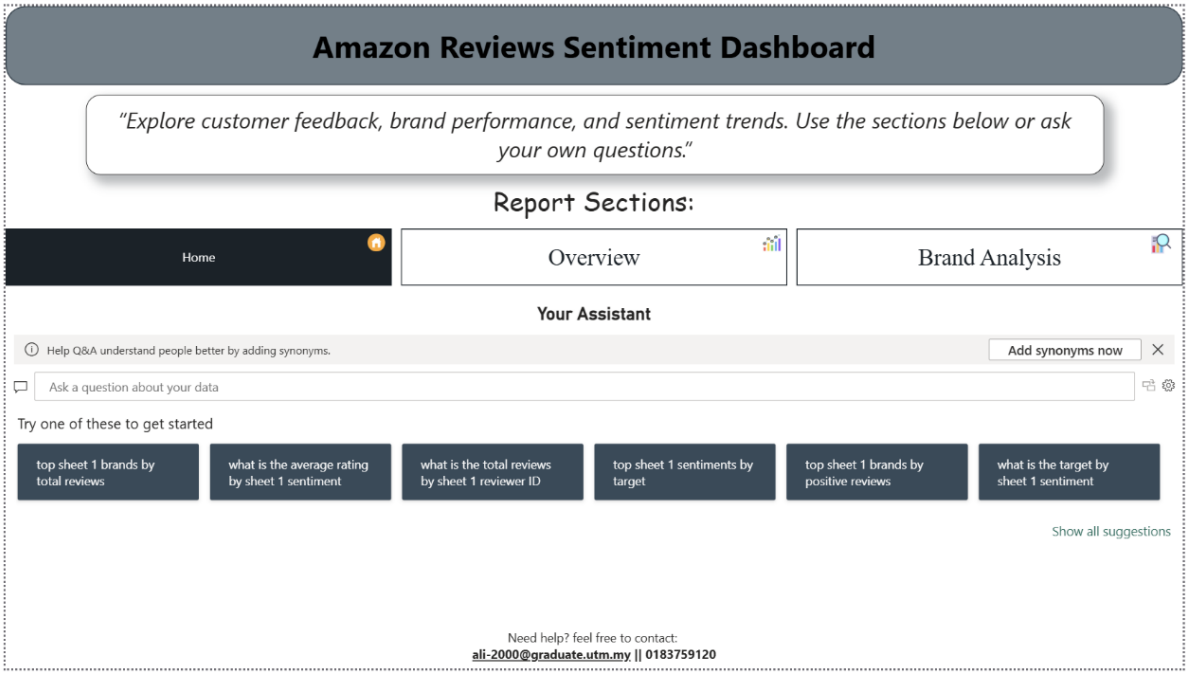

Figure 6.10 Home Dashboard

6.4.2 Overview Dashboard

The overview dashboard shows how people in the dataset feel about the company as a whole. The numbers show 618,000 positive reviews, 32,000 neutral reviews, and 56,000 negative reviews. It says the average rating is 3.99. The 87.61% positive review rate is notable. It quickly shows that most customers liked their purchases.

A bar chart shows if verified buyers rate products differently by comparing sentiment between verified and unverified reviews. A doughnut chart shows the percentage of each sentiment, and a brand rating ranking shows how well each company is doing.

Next to these numbers are sample customer comments on the dashboard. The scores' "I NEED TO RETURN THIS, PLEASE" and "2nd, PERFECT!" convey their tone. The results feel more like customer experiences with these phrases.

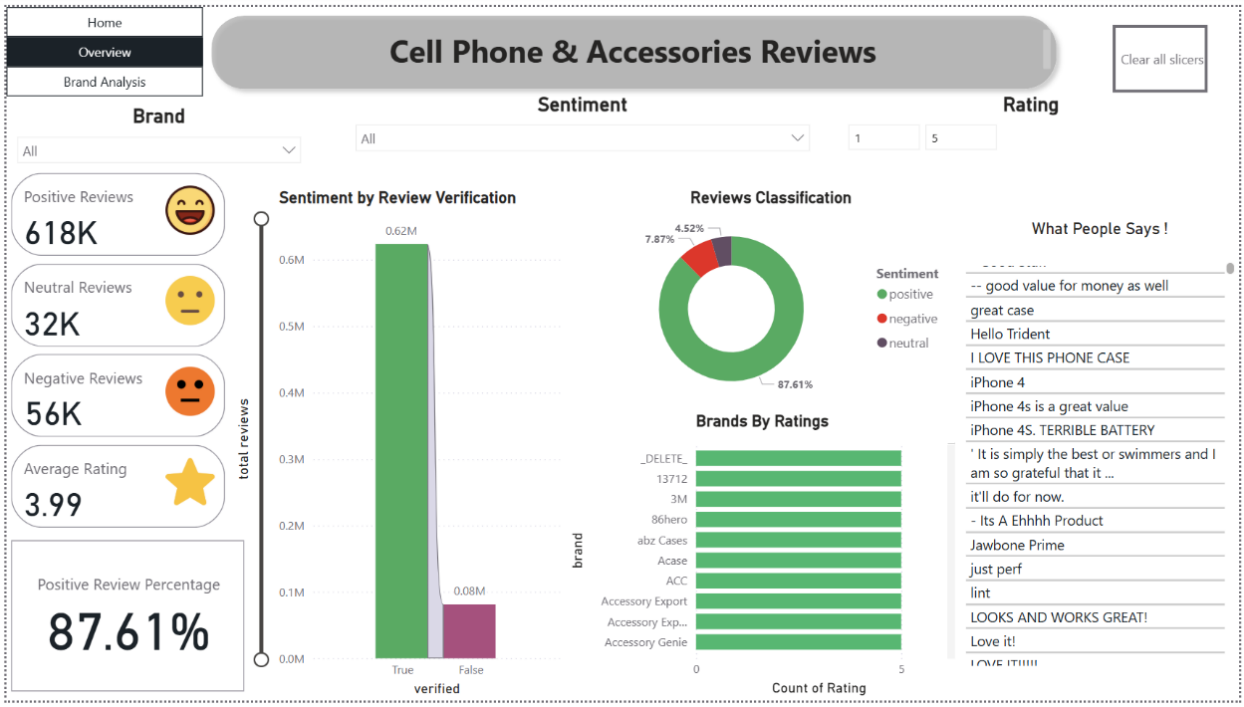

Figure 6.11 Overview Dashboard

6.4.3 Brand Analysis Dashboard

Detailed brand performance comparisons are available on the brand analysis dashboard. The number of reviews 705K and percentage of negative reviews 7.87% give a quick impression of brand sentiment. Furthermore, bubble charts show total reviews, sentiment, and average rating. Bubble size indicates review count. This helps identify brands that consistently receive positive reviews and those that receive negative ones.

The dashboard displays reviewer feedback and a word cloud of popular words like "love," "great," and "casephone." These words recur. These things reveal how customers feel about a brand and how they describe it.

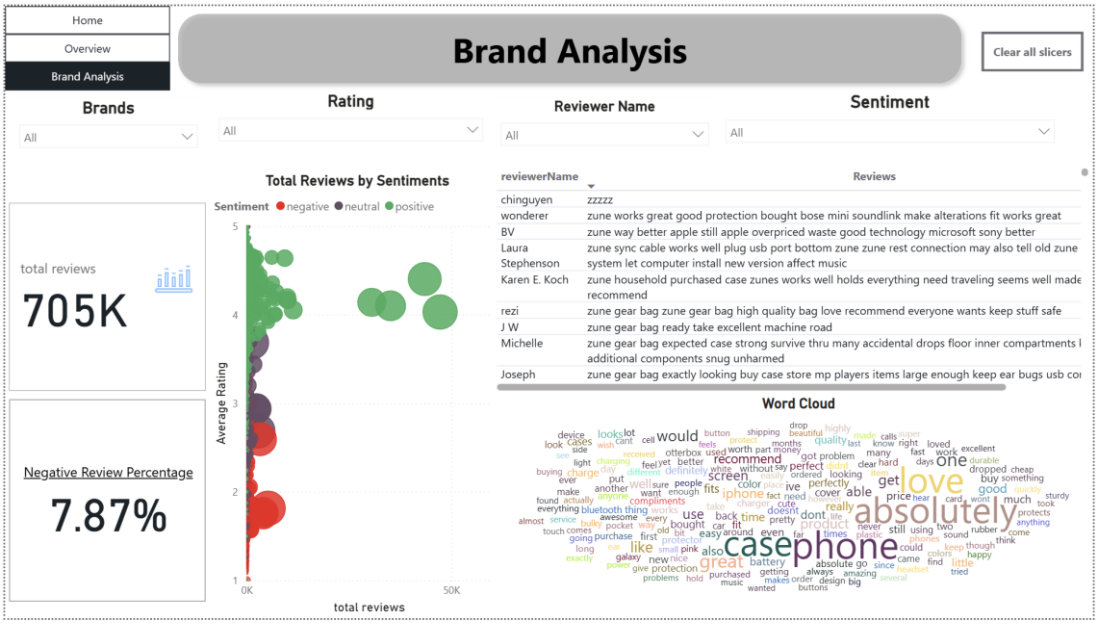

Figure 6.12 Brand Analysis Dashboard

6.5 Interpretation of Research Problem

The aim of this study was to determine whether machine learning could effectively categorize Amazon product reviews into three distinct sentiment classifications: positive, neutral, and negative. It also wanted to know if these categories could be shown in a way that helps businesses figure out how customers act. The models and dashboards show that we have reached this goal, and both the analysis and the visualization have helped us better understand the problem.

The results of the classification showed that automated methods are good for handling a lot of reviews. The Linear SVC model was the best because it was about 95% accurate and the scores stayed the same for all three sentiment classes. This shows that machine learning can give a good idea of what customers think without having to label each review by hand, as long as there is enough data and well-designed features. The less powerful Multinomial Naïve Bayes model was still able to find general patterns in sentiment. Its flaws, especially when it came to neutral reviews, showed that more advanced methods like SVC are better for complicated tasks. The two models demonstrated the feasibility of categorizing sentiment with varying degrees of precision and intricacy.

Power BI dashboards clarified the research problem. Dashboards made numbers and reports interactive and visual. Users could compare brand performance, see sentiment trends, and find customer feedback patterns. The brand-level dashboard showed which companies had the highest ratings and which issues, like poor product quality or delivery, were most often mentioned in negative reviews. This direct link between statistical models and real-world customer problems shows how analysis and visualization help.

The models were able to process and sort opinions because of the original research question, and the dashboards made the results clear and helpful. This mix fills the gap between raw review data and useful insights that can help businesses make decisions. This means that businesses can keep track of how people feel about their products, find areas that need improvement, and see how things are getting better over time. The study has shown that combining machine learning with dashboard visualization is a good and effective way to deal with the problem of analyzing a lot of customer feedback.

6.6 Comparison with Existing Work

During the past years, sentiment analysis has emerged as one of the most extensively researched applications of natural language processing. This is particularly true for large online organizations such as Amazon. Traditional machine learning approaches are frequently utilized by researchers. Support Vector Machines and Naïve Bayes are two of the most commonly used models for determining baseline performance. These techniques are widely used despite the fact that they are not as sophisticated as more recent deep learning models. This is due to the fact that they are both quick and effective regarding text classification.

This section's objective is to compare and contrast the findings of this study with those of previous research that has been conducted. In this approach, it will be able to observe where the results support what has been done already, such as the fact that SVM is superior to NB, and where new research expands on what has been done

in the past, such as by adding engineering features and tools for visualizing data. At the same time, these similarities enable to examine common issues, such as the persistent challenge of classifying unbiased sentiments, within the framework of a more comprehensive academic setting.

Table 6.5 presents a comparison that illustrates a pattern that has been observed in previous research including different studies: When it comes to the classification of emotions, Support Vector Machines (SVM) typically perform more effectively than Naïve Bayes (NB), particularly in situations when the datasets under consideration are larger and more complex. SVM was found to be consistently more accurate and stable than NB when applied to Amazon review data, according to the findings of Guia et al. (2019) and Dey et al. (2020). In addition, Nurul Jannah's (2024) analysis of Shopee discovered that SVM showed a higher level of accuracy than NB, with 86.1% accuracy compared to 81.4% accuracy. Consequently, this indicates that the model is more capable of managing product reviews.

Obiedat et al. (2022) exhibited a more pronounced gap in performance, with SVM obtaining 76% and NB achieving 50% on a dataset of customer reviews that were not evenly distributed. This research was conducted on a dataset that contained customer reviews. The hybrid approach that they developed raised accuracy to 80%, which implies that SVM-based algorithms are preferable when it comes to dealing with class distributions that are distinct from one another. Another significant issue was brought up by Tarimer et al. (2019), which is that the data from IMDB demonstrates that neural networks do better with short, sparse texts such as tweets, but support vector machines perform better with lengthier, more complete reviews.

An F1-score of 0.95 and an accuracy of 95% were achieved by Linear SVC in the current investigation, which produced results that were at the top end of the range of findings that were reported. This is an improvement over a significant number of the results that were obtained in the past, and it demonstrates how important it is to implement feature engineering approaches such as TF-IDF with Chi-square selection, POS tagging, review meta-features, and polarity lexicons. The result reveals the same problems that have been discovered in other studies, specifically how it is unable to

handle nuanced or neutral assessments well. This is despite the fact that NB achieved an accuracy rate of 82% in this particular instance.

Ali et al. (2024) explored sentiment analysis on a large collection of around 50,000 Amazon product reviews by comparing different approaches ranging from traditional machine learning to modern deep learning methods. In their work, classical algorithms such as Support Vector Machine and Naïve Bayes were tested alongside deep learning models like CNN and LSTM, as well as transformer models such as BERT and XLNet. The results showed a clear trend: transformer-based models provided the strongest performance, followed by CNN and LSTM, while the traditional machine learning methods achieved lower accuracy. Their study demonstrates how the field has shifted from relying mainly on classical techniques toward deep learning, with transformers setting the benchmark for sentiment analysis tasks on large review datasets.

overall, this project findings indicate that NB continues to be a straightforward and practical baseline. When dealing with vast amounts of review data that is abundant in language, however, support vector machines (SVM) are the most suitable option for conducting sentiment analysis. By illustrating the additional benefits that may be gained by combining SVM with feature engineering approaches, the findings of this study not only validate the findings of previous research but also add to the body of knowledge in this area.

Table 6.5 Comparison of This Study with Existing Research Using SVM and Naïve Bayes

Author – Year	Dataset	Dataset Size	Method and Features	Performance	Findings
Omar Albaagari (2025)	Amazon Cell Phones & Accessories	~96000 Balanced reviews	SVM vs NB with Feature Engineering	Linear SVC= 0.95 MNB= 0.82	SVM was always higher, whereas NB stayed a strong basis. The MNB was affected by the FE which make it better in performance.
Nurul.Jannah. (2024)	Shopee beauty product reviews	~5,000 reviews	SVM vs NB (10-fold CV)	SVM ≈ 86.1%, NB ≈ 81.4%	SVM was always higher, whereas NB stayed a strong basis.
Obiedat et al. (2022)	Customer reviews (imbalanced)	~15,000 reviews	SVM vs NB vs hybrid SVM-PSO	SVM ≈ 76%, NB ≈ 50%; Hybrid SVM ≈ 80%	SVM is far stronger than NB, and a hybrid SVM made things even better in scenarios where the data was imbalance.
Dey et al. (2020)	Amazon product reviews	~34,627 reviews	SVM vs NB (TF-IDF)	SVM > NB there is no specification of the accuracy	SVM was better than NB in classifying sentiment.
Guia et al. (2019)	Amazon mobile phone reviews	~10,000 reviews	SVM, NB, DT, RF	SVM ≈ 85%, NB ≈ 78%	SVM is the best in terms of accuracy, precision, and recall. NB is poorer, especially when it comes to neutrality.
Tarimer et al. (2019)	IMDB movie reviews & Twitter	IMDB: 2,000 reviews; Twitter: 3,200 tweets	SVM vs NB	**IMDB**: SVM ≈ 85.5%, NB ≈ 73.2%; **Twitter**: NB ≈ 75.4%, SVM ≈ 72.5%	SVM is better for long reviews (like those on IMDB), whereas NB is better for short texts (like those on Twitter).
Ali et al. (2024)	Amazon product reviews	~50,000 reviews	Machine Learning Vs Deep Learning	BERT/XLNet > CNN/LSTM > SVM/NB	Deep learning models achieve perfect sentiment accuracy compared to machine learning.

6.7 Chapter Summary

In this chapter, you were instructed on how to test the models for sentiment classification and how to report the results of those tests in a manner that is recognizable. As the training size increased, the learning curve study shown that both Linear SVC and Naïve Bayes improved their performance for the better. Nevertheless, SVC consistently shown superior generalization. The verification of this was done through the utilization of confusion matrices and performance indicators. SVC achieved a remarkable 95% accuracy and balanced scores across all sentiment classes. On the other hand, Naïve Bayes achieved an accuracy of 82% and encountered greater challenges while evaluating neutral cases.

In this research, it was observed that SVM-based techniques often outperform Naïve Bayes in review datasets. This pattern was also observed in prior studies, which validated the findings of this research. Things were made even better by the addition of designed features, particularly in situations where things were unclear. Last but not least, the dashboards that were created with Power BI were able to transform these technical facts into information that was actually helpful to people in making decisions. There was a connection made between statistical analysis and business interpretation.

In a nutshell, the Linear SVC model was the most effective one, but the Naïve Bayes method was still an outstanding instrument for comparing models. Using dashboards, the study was able to become more transparent, which ultimately led to the primary findings and suggestions that are presented in the next chapter.

CHAPTER 7

CONCLUSION

7.1 Achievement

The study aimed to evaluate the efficacy of machine learning in analyzing the sentiment of Amazon product evaluations and to determine if the findings might be transformed into beneficial instruments for enterprises. The research yielded significant results. Two models for classifying sentiment were successfully built and tested which are Linear SVC and Multinomial Naïve Bayes. The Linear SVC model was very accurate, with an F1 score and accuracy of 95%. The Naïve Bayes model, on the other hand, was only right 82% of the time. It was a good place to start when comparing. These results showed that machine learning approaches can accurately sort a lot of tests into positive, neutral, and negative sentiments.

The second success was learning about feature engineering and how it may impact how well a model works. The models were better at picking up on minor linguistic and structural signals when TF-IDF was combined with Chi-Square feature selection and other characteristics including part-of-speech counts, review information, and polarity lexicons. This was notably helpful for Naïve Bayes, which demonstrated a clear improvement in how it dealt with neutral evaluations when these intended features were added. The results showed that combining different sorts of features can improve classification results.

Finally, the study did more than simply look at models. It also made interactive dashboards in Power BI. These dashboards made it easy for anyone who needed to make decisions to see the classification results. Users could observe how overall sentiment was spread out, how well different brands did, and what terms were used most often in reviews. By connecting technological research with real-world knowledge, the dashboards helped businesses make sense of hard data.

7.2 Limitation

The study showed promising results, but significant limitations must be acknowledged. The first issue is that the dataset only included Amazon Cell Phones and Accessories reviews. Due to its exclusive focus, the vast dataset can't be used to other product categories or review sites. Another constraint is the number of models tested. This project examined only two classifiers which are Naïve Bayes and Linear SVC. These were useful for comparison; however, BERT or LSTM might have worked better.

Classifying neutral sentiments remains difficult. This class was difficult for both models, as sentiment analysis research has shown. Old models still struggle to distinguish between positive and negative sentiments. Power BI dashboards were great for discovering outcomes; however, they merely showed data. They didn't include real-time information, future prediction, or interactive simulations to help individuals make decision.

7.3 Future Work

Future studies might broaden this subject in many directions. It would be interesting to see how the models work in a larger range of situations if added reviews from other product categories or sites like eBay, Shopee, or Yelp to the dataset. Looking into more advanced methods, including deep learning models like BERT, RoBERTa, or hybrid ensembles, could improve performance, especially when it comes to handling neutral emotions. Companies will also learn more about certain product features, like battery life or delivery service, by switching from three-way classification to aspect-based sentiment analysis. It may also be helpful if the dashboards had real-time data, predictive features, and alarms that could be modified. Finally, introducing explainable AI approaches like SHAP or LIME would make things clearer and more reliable by illustrating how the model makes predictions.