Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Analyzing consumer impressions in E-commerce data through feature extraction and sentiment analysis using deep learning

Master Thesis

Name of the Study Programme

Master Project Management and Data Science

Faculty 3

from

Himanshu Dharm

Date:

Berlin, 06.08.2023

1st Supervisor: Prof. Dr. Tilo Wendler

2nd Supervisor: Prof. Dr. Bertil Haack

Table of Contents

List of figures

List of Tables

Index of abbreviations

ANN	Artificial neural network
SVM	Support vector machine
StSc	Sentiment score Long Short-term memory
LSTM	Long Short-term memory
CNN	Convolutional Neural Network
BERT	Bidirectional Encoder Representations from Transformers
SVC	Support Vector Classifier
TF-IDF	Term Frequency - Inverse Document Frequency
ML	Machine learning
AI	Artificial Network
POS	Part of speech
BOW	Bag of words
GRU	Gated Recurrent Unit
RNN	Recurrent Neural Network
ReLU	Rectified Linear Unit
NSP	Next Sentence Prediction
NLP	Natural Language Processing
KDD	Knowledge Discovery in Database
CRISP-DM	Cross-Industry Standard Process for Data Mining

1 Introduction

Technology transformation is inevitable, transforming the functioning of our day-to-day life. This can be as simple as selecting different products for consumption. The importance of e-commerce is on the rise, especially in a post covid economy. Global e-commerce sales are anticipated to reach $6.5 trillion by 2023, according to a forecast by eMarketer Ethan Cramer-Flood, (2020). With this ever-increasing scale of E-commerce, businesses are generating enormous amounts of data. This data is incoming from multiple channels, like social media, customer reviews, feedback, and ratings. As more consumers turn to online shopping, businesses are rapidly investing in digital transformation and harnessing the power of Artificial Intelligence in an effort to improve their offerings and better serve their customer expectations.

An average consumer in their lifetime purchases more than 1 product via online platforms. Out of these, the consumers which are impressed by the product tend to provide positive feedback that others can refer to. The unhappy consumer will do the exact opposite and provide negative feedback addressing the issue. From a general perspective it is a straightforward system, whatever the consumer feels is expressed directly and is provided a value from 1 to 5, 1 being the lowest and 5 being the highest. The complexity arises when the feedback given the highest rating to the product was completely based on their personal choice or a particular aspect of that product, whereas someone who bought the same product based on the same feedback might not have the same experience. For example: a consumer might care more about the price of the product whereas the other put emphasis on its location. Thus, the overall rating does not encompass the true measure of satisfaction of the consumer.

A single review usually consists of multiple components of human emotions that the consumer is expressing. Quite often positive sentiments get expressed with the use of negative words. Such reviews pose a challenge for traditional systems to classify them as positive. As observed by Madhusudhan Aithal, (2021), their experiment showed that, in their dataset, the negative word in the majority of the cases was followed by a positive word. Furthermore, sentences which have positive words are usually negative depending on the classifier. Sentiment analysis is highly subjected to multiple challenges since not every consumer will express their emotions the same way. There will be the presence of sarcasm, emoticons, and slang words to name a few. As a result, it becomes a complex task to take into consideration the human sentiment expressed via the text.

It is important to extract relevant features from the text data, such as words, phrases, sentiment indicators, or grammatical structures, that can provide insights into the expressed sentiment. Effective feature extraction requires a thorough understanding of the language, context, and domain used in the text data. The main goal of feature extraction is to create a set of features that can be used to train a sentiment analysis model. A high-quality feature set can significantly improve the accuracy and performance of the sentiment analysis model and enable it to identify sentiment in new and unseen text data. In summary, sentiment analysis and feature extraction are closely related and essential components in the development of precise and reliable sentiment analysis models.

In a situation where there is a surplus of data, businesses must analyze consumer impressions in order to enhance their customer experience and boost sales. Analyzing this data gives the e-commerce businesses the ability to customize or modify their products and services to better match and enhance the customer experience and expectations. Understanding customer sentiment, preferences, and pain areas can also help derive efficient marketing campaigns, which can be targeted toward a specific audience. E-commerce companies like Amazon, Zalando, eBay, and multiple more companies have thousands of products being purchased. The reviews left by their customers become an important criterion for other users, which causes deviations in their consumption choices. Such direct customer experiences can form a source for businesses to address problems and maintain their quality.

1.1 Business Application and background of sentiment analysis

A study by Haque et al. (2018) on Amazon revealed over 88% of online shoppers entrust reviews, they are equivalent to personal recommendations**.** The vast amount of human knowledge that is available now has given an impetus to the development of sentiment and opinion mining. Although there has been advancement in the field of AI and currently with the rise of ChatGPT there has been a complete shift with how businesses and consumers are perceiving AI. A key application is recommendation systems which are based on sentiment analysis. Elzeheiry et al., (2023) have signified the role played by sentiment analysis, and word embeddings. Further mentioning the use of consumer-centric (reviewer) features to increase the accuracy, indicating the effectiveness of feature extraction methods like Word2Vec or TF-IDF.

Brand reputation and management

E-commerce platforms like amazon server as a medium for businesses to make their products available to the consumer market. As a result, when these consumers start providing feedback as per their satisfaction levels these businesses/brands can begin understanding the overall sentiment towards their products overall and at an individual level. Businesses can learn important information about how clients and the general public view their brand by utilising sentiment analysis. They can spot sentimental tendencies like good or negative shifts, reoccurring problems, or newly arising worries. With the use of this data, companies can better understand client needs, spot opportunities for development, and take data-driven decisions to build their brand's reputation. This process can guide these brands to increase their reputation by promoting their best sellers as well as by addressing and taking actions on customers' pain points.

Consumers who tend to make a purchase usually end up purchasing when they refer to after browsing multiple reviews (Andrienko et al., 2021). As per Chen, (2022) there is a direct connection between the person who has reviewed this product in detail and other consumers who will purchase the product based on the opinion of the reviewer. The author further goes on to mention how brands can divert their marketing budgets by getting products reviewed by such opinion influencers.

Market research and competitor analysis

Sentiment analysis can help brands identify different market trends with the help of where the sentiment is pivoting. As a result, these brands can further adjust the course of their products in order to meet the market demands. Similarly, with the available data the competitor brand analysis can also be carried out. This information can be a key aspect in order to get an edge over the competition. For example: Brand A can conduct a sentiment analysis of Brand B’s newly launched product line. They can understand the consumer sentiment, identify the gaps, and come up with a product themselves that will have the gaps of the competitor.

The research conducted by Rambocas and Pacheco, (2018) indicates that marketers might often rely on a single destination for the reviews or comments. It is essential that they explore the plethora of venues available, this practice can help them tackle the challenge of biases. Researcher further addresses how small scales business can be affected when dealing with the costs imparted by brand specific comment tracking. Thus, signifying such businesses to stick to open-source tools like Python NLTK.

Consumer feedback and marketing strategies

In Ghose and Ipeirotis, (2011) research, the authors point out the significance of the structure of consumer reviews. It is realised that product reviews have 2 parts, subjective and objective. Product sales were associated if the review was only subjective or only objective. But was perceived negatively if it had a combination of both.

Furthermore, the researchers Ghose and Ipeirotis, (2011) also conclude that the readability of the reviews directly affected the product sales. In cases of products from electronic categories performed well which had a mixture of objective parts and less subjective parts which described well. Thus, indicating how the consumer feedback can help drive product sales. This information can help businesses drive marketing strategies for example: incentivising the consumers for providing detailed and well curated reviews. Customized applications can be deployed which can process these feedbacks automatically and group them into different categories as per their priority.

1.2 Research Problem

The growing industry attracts more customers on a daily basis. The customers in turn generate a huge amount of data. This data that is mined from customer reviews and comments can be unstructured, thus using a sentiment analysis the customer sentiment can be extracted from this data. A report by statista (Stephanie Chevalier, 2022) displays that the e-commerce industry will hit the 8.1 trillion mark by 2026. Thus, making it even a concrete need to understand the customer emotion.

There is an exponential growth in the direction of Machine learning development, especially for LLMs (large language models). With this ever-increasing development, there is an excess availability of different versions of a single model that can be implemented. SVM (Support Vector Machine), XG-Boost, and Random Forest are a subset of the machine learning models that are present at our disposal. Moreover, LSTM (Long term short memory), and BERT (Bidirectional Encoder Representations from Transformers) are a pair of Deep learning models to name a few whose applications are well suited for sentiment analysis.

RQ1: How do different machine learning models compare in terms of their performance and effectiveness for sentiment analysis on e-commerce review data?

RQ2: Which feature extraction method is best suited for sentiment analysis of Amazon reviews (TF-IDF or Word2Vec)?

The proposed research is developed in the direction of exploring different deep learning models and comparing their results. The findings of this study aid in increasing prediction accuracy by investigating which machine learning model will perform better in terms of accuracy on Amazon customer review dataset.

1.3 Objective

A substantial portion of the world's data was generated in a small period. Over decades and continuous digitization, there has been an exponential surge in this data. As a result, there is an increasing demand to mine this data and achieve a greater understanding of different subjects. These subjects refer to different domains from where this data is generated from. Present day organizations heavily depend on this data to run their algorithms and make an impact. Such is the power of this collected knowledge. Especially when there is an infinite amount of data getting generated as we live. Even though there have been many breakthroughs of current technology of AI (Artificial Intelligence), natural language is a field that is still one of the toughest to conquer. ML (Machine Learning) approaches do provide high accuracy, but they require certain expectations in terms of data availability, processing power and time for training the models. Most often, the data required must be labelled data, in real world applications this is difficult to obtain.

The target entity is the sentiment in online customer reviews. Ecommerce websites provide a platform for the users to share their opinions in the form of product reviews that the user has consumed. These reviews are provided with an overall rating, this is a numerical rating that expresses a general user sentiment of the product. For example, a rating of less than or equal to 3 can be considered as bad or moderate, whereas those above 3 are good. This rating associated with the review helps understand the user perspective and guide the model to predict and assign scores to new and unseen data.

This thesis aims to explore the history and development of sentiment analysis techniques. A comparative in-depth analysis of machine learning methods and further apply these methodologies and compare their results using different evaluations like accuracy, precision, F1 score and other relevant metrics. Furthermore, an effort will be made to understand feature extraction and its effectiveness. The dataset used for this research is from Amazon (Amazon product customer reviews). Furthermore, the difficulties that persist while performing the sentiment analysis using deep learning methods will be explored and how they can be addressed.

1.4 Outline of the thesis

During this thesis, it will begin with exploring the existing research work in the direction of sentiment analysis. These studies focus on algorithms, techniques, and models like SVM, LSTM, and BERT as well as the dataset employed and the evaluation metrics. Targeting the history and the development in this domain of Natural Language Processing (NLP). In order to establish the basis of this thesis, a concrete study and understanding of machine learning models and their evolution will be done. The literature review will explore and examine the existing work in this field and evaluate their result. Furthermore, it will help the research address and tackle the challenges that other researchers have come across.

Following the literature review, the theoretical section serves as the basis in understanding sentiment analysis, its evolution, and different methods to address human emotions behind the text data generated by the average consumer. Additionally, a deep dive is done in different types of machine learning methods available for sentiment analysis. As each method has a unique way of processing the textual data with different levels of granularity. Finally, evaluations metrics will be discussed that will help this research compare the final result, before that it is essential to understand how each of these metrics work in assisting the research. This theoretical understanding will ensure a concrete foothold in supporting the methodologies that will follow.

The methodology will cover the section wherein the dataset used for this thesis will be pre-processed to make it in the acceptable format for the machine inputs. The preprocessing will involve an initial exploratory analysis of the dataset, followed by cleaning the dataset to get rid of null values, duplicates and other abnormalities which can adversely affect the modelling phase

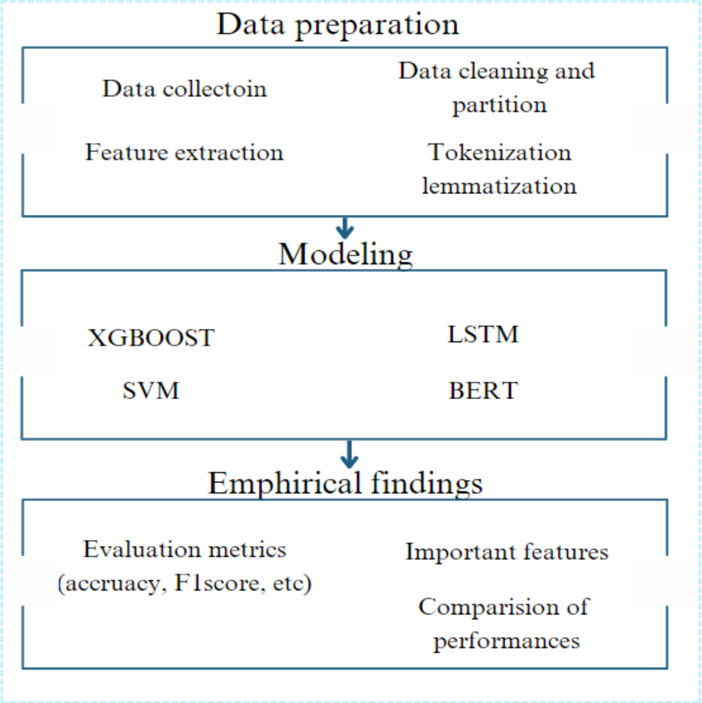

Data preparation
Data collectoin
Data cleaning and partition
Feature extraction
Tokenization lemmatization
Modeling
XGBOOST
LSTM
SVM
BERT
Emphirical findings
Evaluation metrics (accruacy, F1score, etc)
Important features
Comparision of performances

Figure 1. Representation of methodology

The experiment will be set by executing XGBOOST, SVM, LSTM, and BERT with the input data. Thus, resulting in a comparative analysis to discover which study proves to be more efficient and accurate. Ultimately, the resulting metrics will help the research in determining which of the selected machine learning models perform most efficiently in determining whether the sentiment is positive or negative.

It is not this thesis's objective to devise a completely new model, neither is it the aim of this thesis to develop a model that achieves 100% accuracy in recognising the polarity of the hidden elements in the customer reviews. Rather, is it possible to capture all the hidden information in the customer reviews without errors. This thesis will aim to provide a comparative analysis by providing acceptable results for real-world applications.

2 Literature review

The literature review for this thesis can be summarized in 6 parts, starting from an overview of sentiment analysis, techniques for sentiment analysis, sentiment analysis using deep learning models, review of relevant research work comparing machine learning models, and final y challenges faced in sentiment analysis. The fundamental analysis section will focus on providing a basis for the need of sentiment analysis, and using it as a segway to understand different types of sentiment analysis techniques. Traditional approaches section will go through techniques that were available before deep learning came into picture. As for sentiment analysis using machine learning will encompass the development of machine learning as a tool to understand human emotion from the textual data. Moreover, also dive deep into different types of machine learning models. Final section will cover the existing research work that is available in the direction of comparing the results of different machine learning models. The findings and the theory will be explored and explained.

2.1 Overview of Sentiment analysis

Sentiment analysis is a challenging task that includes identifying and extracting opinions, emotions, and attitudes from text data. In the first paper, the author discusses various techniques used for sentiment analysis, including lexicon-based methods, machine learning-based methods, and hybrid methods. It further goes on to discuss subjectivity analysis, which involves the question of whether a test is subjective or objective. It is imminent that sentiment analysis is a complex task given how people or consumers express their emotions in different ways. Lack of labelled data, handling sarcasm and irony, and domain-specific languages are some of the challenges associated with sentiment analysis. The sentiment analysis models need to be able to recognize different contexts, in many cases the same word or phrase tends to have different meanings depending on the context (Liu)**.**

The paper by Pang and Lee, (2008) gives an overview of opinion mining, the authors utilize machine learning approaches like supervised and unsupervised learning for sentiment classification. Furthermore, the research explores challenges put forth by irony, and subjectivity in sentiment analysis. This opinion classification can be of utmost importance for product-based brands. They can utilize it for their marketing research, addressing customer issues and further strengthening their brand’s position.

In Jemimah Ojima Abah, (2021) research, amazon electronic product reviews were used for sentiment analysis. The author utilised CNN and LSTM model to perform sentiment classification. Initially, an imbalanced dataset was used to perform sentiment analysis which resulted in biased results and overfitting. Thus, an up sampling was performed for the minority class in order to remove the bias. Furthermore, the author concluded that, when an imbalance dataset was used to train the model there was a high type 1 error rate (False Positive). Whereas. When models were trained using the unsampled data there was a high type 2 error rate (False Negative). Finally, hyperparameter tuning was performed where it was observed that LSTM showed improvements, but CNN was performing the same as before.

Dave et al., (2003) explores the topic of sentiment analysis and opinion mining that would help the data decision making processes as well as aid in understanding the customer preferences. The authors discuss the challenges faced during the research like rating inconsistency, and reviews being of sparse length. Ambivalence was one of the challenges where reviewers used negative words but ultimately tend to express their satisfaction. As a result, traditional approaches like SVM face difficulties to perform granular classifications.

In the paper by Jonathon Read, (2005), titled "*Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classification*," the author explores the use of emoticons as a valuable resource for sentiment analysis. The paper investigates how incorporating emoticons can help reduce the dependency on manually annotated data and improve the performance of machine learning techniques for sentiment classification. The paper begins by highlighting the challenges of sentiment classification, including the scarcity of labelled data and the need for effective feature selection. It then introduces the idea of leveraging emoticons, which are often used in text to express emotions and sentiments, as a source of information for sentiment analysis. Read presents an experimental study in which machine learning techniques are applied to sentiment classification tasks, with and without the inclusion of emoticons as features. The author compares the performance of different classifiers and evaluates the impact of including emoticons on sentiment classification accuracy. The results of the experiments demonstrate that incorporating emoticons as features can enhance the performance of sentiment classification models, particularly in cases where labelled training data is limited. The paper concludes by discussing the implications of these findings and highlighting the potential of emoticons as a valuable resource for sentiment analysis tasks (Read, 2005).

In their research Tripathy and Rath, (2017) used 3 datasets with different JSON format. The data was unlabelled and manual labelling was not a feasible option. They pre-processed their data and used active learners for labelling. The scale they used was, 5 stars being the most positive review, 3 star being the neutral and 1 or 0 stars as the most negative. The data pre-processing involved tokenization of data, removing stop words and Part of speech tagging. Feature extraction process involved Bag of Words, Term Frequency–Inverse Document Frequency (TF-IDF), and Chi-Square. The research included use of Naïve Bayesian, Support vector Machine Classifier (SVC), Stochastic Gradient Descent (SGD), Linear Regression (LR), Random Forest and Decision Tree.

The result of their study found out that Support vector Machine Classifier with an accuracy of 94.02%.

2.2 Techniques of sentiment analysis

Supervised learning is a machine learning technique where the algorithm is trained on a labeled dataset, which means that the dataset has pre-assigned output values. In the case of sentiment analysis of Twitter data, the algorithm is trained on a dataset of tweets that have been manually labeled as positive, negative, or neutral. The algorithm then learns to recognize patterns in the data associated with each sentiment label, and uses these patterns to classify new tweets as positive, negative, or neutral. Supervised learning algorithms commonly used in sentiment analysis include logistic regression, support vector machines, naive Bayes, and decision trees. Recently, deep learning techniques like recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have also shown promising results in sentiment analysis. It is worth noting that feature engineering techniques can also be used to extract relevant information from the text of tweets, which can improve the accuracy of sentiment analysis (Hochreiter and Schmidhuber, 1997).

In unsupervised learning, the algorithm is not given labeled data but rather must find patterns and structures in the data on its own. This can be useful in cases where labeled data is scarce or expensive to obtain. However, unsupervised learning algorithms may not perform as well as supervised learning algorithms in sentiment analysis because they are not trained on labeled data with known sentiment values. The author notes that some unsupervised learning techniques, such as clustering and topic modeling, can be used for sentiment analysis by grouping tweets based on similarities in content or topic. However, the accuracy of these techniques may depend on the quality of the data and the selection of appropriate features for analysis. Overall, the author suggests that supervised learning algorithms are generally more effective for sentiment analysis of Twitter data, but that unsupervised learning techniques may be worth exploring in certain situations.

There are different types of techniques to perform sentiment analysis. In their paper Kaur and Sidhu explore these techniques (Kaur and Sidhu, 2018). The paper delves into the application of sentiment analysis in different domains of the industry such as social media, marketing, and customer services are some of them. The authors then provide an overview and limitation of some of the traditional approaches, such as lexicon-based approaches and machine learning-based approaches. Furthermore, they finish off by discussing the deep learning approaches which have shown promising results in sentiment analysis due to their ability to automatically learn features from raw data.

The paper found that the proposed RNN model outperformed other machine learning models such as Naive Bayes and Support Vector Machines (SVM) with an accuracy of over 90% in sentiment classification of Amazon product reviews. The study also demonstrated the importance of word embeddings and hyperparameter tuning for optimising the performance of the RNN model (Iqbal et al., 2022).

There are 3 different types of analytical approaches to perform sentiment analysis

Lexicon-based approach

A lexicon translates to a list of words and their associated meaning. In this context, lexicon contains words that are associated with a single polarity. It can be either positive or negative (Hota et al., 2021). The lexicon-based approach in sentiment analysis involves using a pre-prepared sentiment lexicon in order to determine the polarity of a document. For example, in a lexicon-based approach, the word "happy" might be assigned a positive polarity score, while the word "sad" might be assigned a negative polarity score. When analyzing a sentence that contains both words, the lexicon-based approach would take into account the overall sentiment expressed by the words to determine the sentiment score of the sentence.

A popular formula to calculate the sentiment score (StSc). This formula can also be used for a dictionary-based approach (Aline Bessa, 2022).

$$\text{StSc} = \frac{\text{number of positive words - number of negative words}}{\text{total number of words}}$$

Equation 1. dictionary base approach to calculate sentiment score

The lexicon-based approach is divided into 2 main categories: Dictionary based, and Corpus based

a. Dictionary-based

In the dictionary-based approach, pre-defined dictionaries contain lists of words and their associated polarity scores. For example, this method of sentiment analysis might use a lexicon comprising a list of positive words, such as “good”, “great”, and “beautiful”, and also a list of negative words, like “horrible”, “bad”, and “unhappy”. Each word in the lexicon is assigned a value, which is also its polarity, it ranges from -1 to 1.

This process utilizes the information of the polarity that is associated within the dictionary. When analyzing any text using the dictionary-based method, the approach involves computing the overall sentiment of that text. For example, if the text contains multiple positive words and a single negative word then it is assigned a positive sentiment score. Similarly, when there are more negatives than positives, a negative sentiment score is assigned.

Overall, the dictionary-based approach is a simple and effective way to perform sentiment analysis, especially when dealing with smaller datasets. This approach has limitations in capturing the nuances of the language, and may be as sophisticated as some of the machine learning approaches. Due to their reliance on pre-defined lexicons, dictionary-based approach may struggle with content specific sentiment, having difficulty to handle humour or sarcasm.

b. Corpus-based

In the Corpus-based approach a large amount of text is analyzed in order to identify the common patterns and associations between words. For example, use of a corpus of product reviews to build a sentiment lexicon by analyzing the language used in the reviews. This has an advantage over the dictionary-based approach as it has the potential to obtain domain specific orientations.

This approach requires more computational power in order to analyze large corpora. Additional option is crowdsourcing the association of sentiments for a large corpus. Although, this method involves generating a lexicon dictionary, it is efficient has demonstrated that it can achieve higher accuracy in sentiment analysis (Mohammad and Turney, 2013).

Machine learning-based approach

There are 3 types of machine learning approaches for sentiment analysis. Supervised, Unsupervised which are discussed above, and semi supervised which is an amalgamation of both supervised and unsupervised learning.

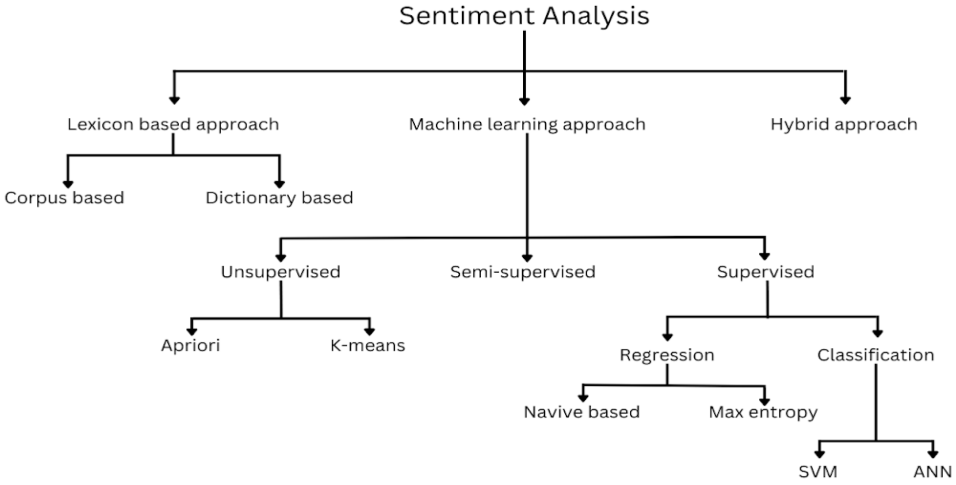
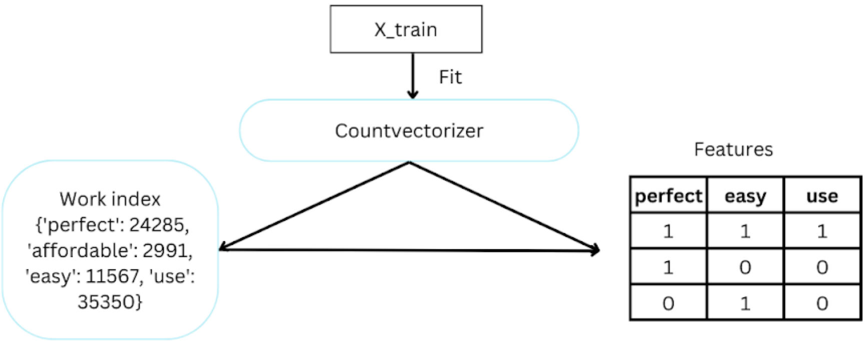


Figure 2. Overview of the Approaches to Sentiment Analysis

2.3 Overview of feature extraction

Feature extraction points to transforming the textual data into number i.e., text vectorization as defined in section 4.2.6. There are various methods for feature extraction using genism and sklearn like, countvectorizer, TF-IDF vectorizer, and Word embeddings. In this paper, TF IDF feature extraction method will be implemented in order to extract features from the textual data. This helps our modelling process to take into consideration the semantic information from the text.

Countvectorizer is a flexible and a general way to extract features. It is a 'Bag of Words' approach, since it only represents the occurrence of words in a document. No attention is paid to the grammatical conventions or the word order. Any information of the arrangement or the structure of the document's word is disregarded (turbolab, 2021).


X_train
Fit
Countvectorizer
Work index
{'perfect': 24285, 'affordable': 2991, 'easy': 11567, 'use': 35350}
Features

perfect	easy	use
1	1	1
1	0	0
0	1	0

Figure 3. Countvectorizer feature extraction(turbolab, 2021)

TF – IDF Vectorizer (Term Frequency – Inverse Document Frequency), considers 2 factors when it calculates the weight for each word in the document i.e., Term frequency and Inverse document frequency. Term Frequency (TF) measures the frequency of a term in a document. Demonstrating the importance of that term of word in the document. Which means that more the occurrence of that term, more the importance. Term Frequency is calculated using different formulas, such as raw term frequency, binary representation, or logarithmic scaling. This approach is similar to the Bag of Words approach. Inverse Document Frequency (IDF) measures the rarity (importance) of a term across the whole corpus. It gives higher weightage to the term or a word that appears less frequently in the document. Inverse Document Frequency is calculated as the logarithm of the ratio between the total number of documents and the number of documents that contain the term (Haque et al., 2018).

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF
Term x within document y

$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

Equation 2. TF-IDF (turbolab, 2021)

Further, the values are combined in order to calculate TF-IDF weight for each word in the document. Higher the weight, more the importance of the term within the corpus and the document.

In Liu et al., (2018) experiment, the author used TF IDF to extract semantic features from the text data. In order to extract features and cluster them, a dataset with 2 main topics was used with 2500 for each topic. TF-IDF and Word2vec were used to generate a virtual word vector. Further, classification was done using KNN (K-nearest neighbour) algorithm. With TF-IDF it was observed that it was able to classify 98% for topic 1 and 82% for topic 2. Whereas, the method proposed using Word2vec saw an increase for topic 101% and topic 2 at 82.2%. Ultimately it was concluded that there was no significant improvement in the accuracy.

In Natural language processing Word embeddings are used to represent the words in a high dimensional space as vectors. Design of these vectors is in such a way that related words are located such that they are close to one another. It helps capture the semantic relationship as well as the context information. This enables the machine learning models to understand and have a generalized understanding of the words in a effective manner.

Introduced by Mikolov et al., (2013), Word2vec is the most common algorithm used on a large corpus of data for learning the Word embeddings. The algorithm learns these embeddings by training a large amount of textual data on neural networks (Mikolov et al., 2013). There are 2 prominent architectures of Word2Vec. Continuous Bag-of-Words (CBOW) and Skip-gram. In Continuous Bag-of-Words the model predicts the current word on the basis of the context it is surrounded by. On the other hand, in Skip-gram, the model predicts the context words given a target word. Both architectures focus on learning word embeddings that carry meaningful information and capture the semantic connections between words.

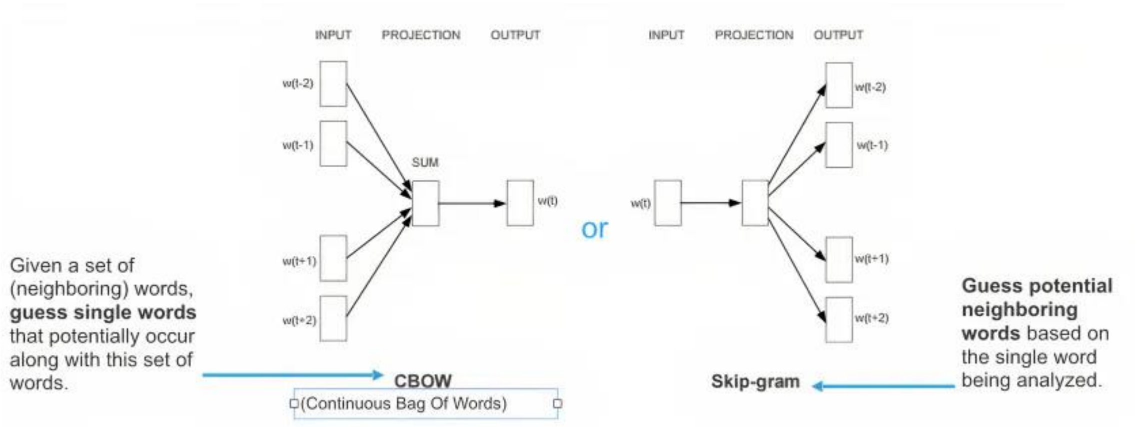

INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
SUM
w(t)
w(t+1)
w(t+2)
or
INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
Given a set of (neighboring) words, guess single words that potentially occur along with this set of words.
CBOW
(Continuous Bag Of Words)
Skip-gram
Guess potential neighboring words based on the single word being analyzed.

Figure 4. Word2Vec feature extraction(turbolab, 2021)

Read, (2005) showed that feature extraction is also dependent on the feature. For the bag of features approach, if a classifier is trained on movie review corpus, there is less surety that it will perform the same on other dataset, for example: automobile or food reviews. (Turney, 2002) showed that the unigram unpredictable could show a positive sentiment for the movie dataset, unfortunately give different results for the automobile dataset. Thus, it is noteworthy to mention that some of the feature extraction methods are feature dependent and can perform with a bias or deviation when tested on other vocabulary.

2.4 Sentiment analysis using Deep learning models

This section will dive into the existing research work of the models that will be used in the research of this thesis; these algorithms belong to the deep learning family of machine learning models. This will help establish the expectation from the deep learning models and the challenges faced by researchers, if any.

Convolutional neural network

"Convolutional Neural Networks for Sentence Classification" by Kim, (2014) is a widely cited paper that explores the application of Convolutional Neural Networks (CNNs) for sentence classification. This paper emphasizes on the effectiveness of CNN for sentiment analysis. The dataset used in this paper belongs to IMDb movie reviews and Stanford Sentiment TreeBank. The author uses the CNN architecture for sentence classification. It employs a convolutional layer followed by max-pooling to capture local features from different n-grams (word sequences of length n) within the sentences. Multiple filters of varying window sizes are utilized to capture features at different scales. The resulting feature maps are then fed into fully connected layers for classification. The experiment was done by using different hyperparameters, such as filter sizes, pooling strategies, and activation functions, to optimize the performance of the CNN model. They also compare the results of their CNN model with several traditional methods and demonstrate the superior performance of CNNs in sentence classification tasks. In conclusion the experiment from the paper demonstrates that CNN achieves state-of-the-art performance on sentence classification tasks in turn outperforming traditional methods. The proposed CNN model in this paper achieved an accuracy of 88.89% using the IMDb movie reviews dataset.

Recurrent neural network

Neural networks have been in use since the 90s. An artificial neural network is similar to the biological neural network. Recurrent neural networks are such that their design is to learn from pattern through time. In simple terms a RNN is a neural connection having a feedback mechanism (Fausett, 1994).

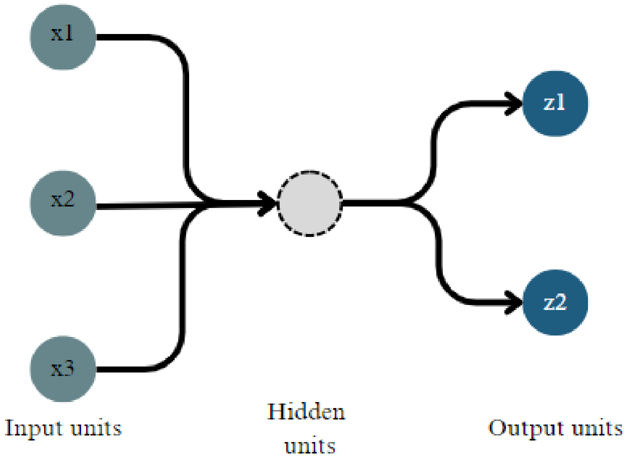


Figure 5. Simple neural network

Recurrent neural networks (RNN), are termed as recurrent because they carry out the same task for each unit of a sequence, and the output always relies on previous calculations (Javaid Nabi, 2019) .

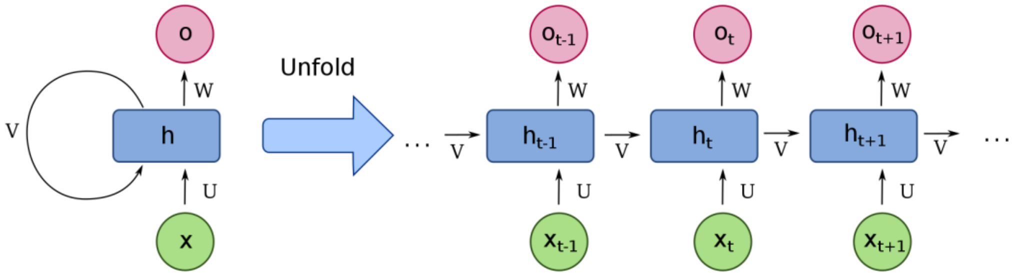


Figure 6. Recurrent Neural Network (Wikipedia, 2023)

For calculating the current state:

$\mathbf{h_t = f(h_{t-1}, x_t)}$

Where,

h_t= **current state**

h_{t-1}= **previous state**

x_t= **input state**

To apply the activation tanh function,

$\mathbf{ht = tanh\ (w_{hh}h_t\text{-}1 + w_{xh}x_t)}$

There are different types of Recurrent Neural Networks.

- Vanilla RNN: This is the base level model of RNN, also known as Elman network named after *Jeffrey Elman*, who introduced the concept of SRNN (Simple recurrent neural network)

- Long Short-Term Memory (LSTM): A gating mechanism and memory unit(cell) are introduced. A deep dive for LSTM will be done in section section number
- Gated Recurrent Unit (GRU): They are similar to LSTM but are computationally better.
- Bidirectional RNN: Convenient where comprehensive understanding is required. They include both backward and forward direction, thus retaining both past and future information.
- Hierarchical RNN: Utilise multiple layers of RNN to process hierarchical data, for example: document classification.

Moreover, as stated by Chung et al., (2014) the successful implementations have been with different versions of RNN and not the base level Vanilla RNN. LSTM, introduced by (Chung et al., 2014) performs significantly better. Likewise, the sophisticated recurrent network GRU has achieved similar results (Cho et al., 2014). (Werbos, 1990) emphasises on back propagation through time as RNNs face a challenge of vanishing gradient. Where the author discusses a sophisticated forward feeding neural network that will unfold the RNN with time.

Transformer based model

Vaswani et al., (2017) introduced the transformer based model. They showed how the transformer based model was able to outperform the convolutional models and the recurrent models. Attention mechanism is where the model tries to understand the importance of a term or a word in a sequence. It is the same as a human being would try to focus on the important part of the sentence. A transformer consists of an encoder and a decoder.

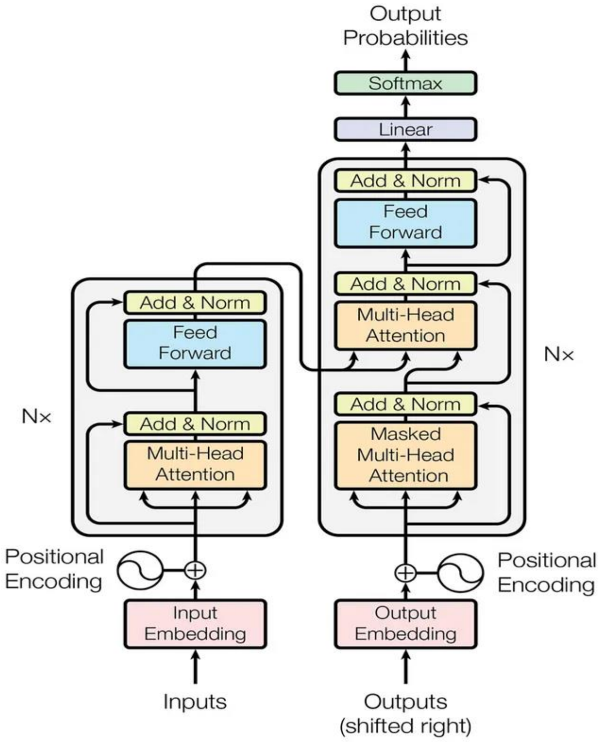


Figure 7. Detailed architecture of a transformer (Vaswani et al., 2017)

A part this thesis will also cover using a transformer-based model called as BERT (Bidirectional Encoder Representations from Transformers) which will help us further understand if it outperforms the other selected models.

2.5 Review of relevant research work comparing machine learning models

There have been multiple studies to determine which machine learning algorithm can yield the best result. Although this process depends on quite a few factors. To begin with is the dataset, the language, and availability of the resources are to name a few. In this paper by Tan et al., (2022) the authors perform a comparative analysis on game reviews using XGBoost, SVC, Multinomial Naïve Bayes, and Multi-layer Perceptron Classifier. These models come from a family of traditional machine learning models and are supervised learning models. It was observed that on an imbalanced and an oversampled dataset SVC performed out-performed other models by an accuracy of 72% and 82% respectively. Furthermore, after hyperparameter tuning, the model was able to achieve an accuracy of 89%. This goes on to show that hyperparameter tuning is a key to realise which parameters perform the best for a given model.

An experiment done by SEPIDEH PAKNEJAD, (2018) where the author has utilised the TF-IDF feature extraction methods on amazon review data on a corpus of approximately 200000 reviews. The author used SVM and Naive Bayes for classification. SVM out performed Naive Bayes(90%) with a score of 93% on reviews. However, the models performed better on summaries of reviews, with a potential limitation being the sparsity of data.

Srinivas et al., (2021) experiment shows the result for a LSTM, CNN and a single layer neural network. The authors utilised 1.6 million tweets for text data and classified them into positive and negative categories. It was observed that LSTM performed the best with an accuracy of 87%. This goes to show that RNN based models have the capacity to perform better in terms of classification for sentiment analysis.

A comparison of CNN, BERT along with a combination of both can be seen in (Li et al., 2021) experiment. Where the author used 1million of Weibo data, which was labelled. The author used multiple combinations of BERT with CNN and LSTM layers at times, but for this thesis focus will be on the outcome of BERT and CNN only. Jieba tokenizer was used for preprocessing the data. There was a noticeable difference in accuracy with BERT at 84.4% and CNN with 73.5%. Indicating a transformer-based model has the potential to yield better performance.

2.6 Challenges faced in Sentiment analysis

In this section, the challenges faced in sentiment analysis will be discussed. These challenges range from techniques used to the availability of data. Below points cover the challenges faced in the application of sentiment analysis.

Handling sarcasm and irony

Sentiment analysis often struggles with detecting the sentiment behind sarcastic or ironic statements. Developing methods to handle such instances would be a valuable contribution to the field. Sarcasm implies leaving a satirical remark and one which is used to emotional hut or mock someone. It is done by using information which is completely opposite of the context provided (polar opposite). It is one of the most challenging aspect in Natural language processing, but gaining an interest due to its usefulness (Eke et al., 2020).

Multilingual sentiment analysis:

Sentiment analysis is often focused on English-language data. However, there is a growing need for sentiment analysis in other languages, particularly in the context of social media. Developing methods to perform sentiment analysis in multiple languages would be a valuable contribution to the field. For instance, English language has its own variations depending on the region like in India, Australia, USA or UK. The word *"thong"* implies undergarments in Indian English (same as British), whereas it means flip flops for Australian english. This difference in itself can lead to a lot of implications on how the textual data is processed. This can lead to duplications and difficulties in classification. A prominent example is *"color"* and *"colour"* (Wankhade et al., 2022).

Sentiment analysis for specific domains

Sentiment analysis is often trained on general datasets, but performance can suffer when applied to specific domains (e.g., healthcare, politics, finance). Developing domain-specific sentiment analysis models would be a valuable contribution to the field.

Ambiguity

Ambiguity is a significant challenge in sentiment analysis. Some words can have multiple meanings depending on the context in which they are used, and identifying the correct sentiment can be difficult.

Emotion detection

Sentiment analysis typically focuses on identifying positive, negative, or neutral sentiment, but emotions are more complex. For example, a statement could be negative, but the emotions expressed could be anger, sadness, or frustration.

2.7 Summary

The literature review section gives a detailed understanding by giving an overview of sentiment analysis. Different methods involved in sentiment analysis. Sentiment analysis has been in existence since ages, after the boom of the internet and freedom for people to express their emotions there was a need for better methods to help business drive change. Furthermore, techniques of sentiment analysis were discussed to develop a better understanding of machine learning and deep learning approaches. The section further explains the relevant work in the field, wherein multiple machine learning models were used for a same corpus in order to derive which perform the best. More the data, the better the result. Though this comes with its fair share of challenges which were addressed above.

Following table helps summarize the overall findings in the literature review section.

Approach	Characteristics	pros	Cons
Lexicon based approach	A straightforward rule-based approach. Assigning negative or positive polarity to keyword in order to calculate the result.	Easy to use. Interpretable.	Does not consider sentence structure, thus a constraint on contextual understanding.
CNN	Able to learn hierarchical representations and patterns from text using convolution layer	Captures local features. Can handle textual data with variable lengths.	Contextual understanding is limited as it does not understand the overall context. Fixed sized input is required for CNN, thus unable to handle varying length text effectively.
LSTM	Versatile in data handling. Utilises Back propagation through time. Gating mechanism and memory cells.	Captures dependencies between the data	Problem of vanishing gradient. Sensitive to hyper parameters.
BERT	A transformer-based model. Utilizes mechanism of self-attention and generate word embeddings.	Transfer learning. Bidirectional attention. Contextual word embedding generation	Higher training and memory requirement. Pre-trained models might not be precise (fine-tuned) for domain specific tasks
SVM	Utilizes a hyperplane to separate classes. A simple linear binary classification technique.	Easy to interpret. Memory efficient.	Limited to binary classification. Sensitive to outliers and noisy data.

Table 1. Different machine learning models employed in this research

Following the thorough research and understanding of sentiment analysis, the following models are chosen for the objective of this research: CNN (convolutional Neural Network), LSTM (Long Short Term Memory), and BERT (Bidirectional Encoder Representations from Transformers). These models belong to the category of deep learning models. As a result, for comparison purposes, this research will also include a traditional SVM (Support Vector Machine). This helps further understand the differences with each method.

3 Methodology

This section comprises multiple aspects relating to this research. It begins with understanding the deep learning models that are going to be used along with traditional machine learning models. it will help establish a firm ground to understand their working on the selected dataset. Following this, the section will cover different evaluation metrics that will be required in order to have a final comparison during the empirical finding and help score the models.

This research process incorporated a comprehensive methodology of Knowledge Discovery in Database (KDD). This methodology comprises certain steps that guide the research and development process. Ultimately yielding the desirable result.

1. Problem Understanding: This stage involves gaining a deep understanding of the problem domain and clearly defining the specific problem to be solved. It includes identifying the project objectives, requirements, and any constraints that need to be considered.
2. Data Selection: In this stage, relevant data is carefully chosen from various sources, considering the specific requirements of the problem. The focus is on determining the appropriate scope and level of detail for the data.
3. Data Preprocessing: This stage deals with preparing the selected data for analysis by cleaning and transforming it. Steps like removing duplicates, handling missing values, and normalizing the data are performed to ensure its quality and suitability for further analysis.
4. Data Transformation: Here, the selected data is transformed and enhanced to enable the discovery of meaningful patterns. Techniques such as aggregating data, reducing its dimensionality, and creating new features are employed to represent the data in a format that can be effectively analysed.
5. Data Mining: This stage involves applying various algorithms and techniques to extract patterns and valuable knowledge from the transformed data. Statistical analysis, machine learning, clustering, classification, and association rule mining are among the methods used to uncover hidden insights. Evaluation: The discovered patterns and knowledge are assessed in this stage to determine their quality, significance, and usefulness. Evaluation metrics and validation techniques are employed to measure the effectiveness and reliability of the discovered knowledge.
6. Knowledge Presentation: The final stage focuses on effectively communicating the discovered knowledge to stakeholders. This may involve using visualization techniques, reports, or interactive interfaces to present the insights and findings in a clear and understandable manner.

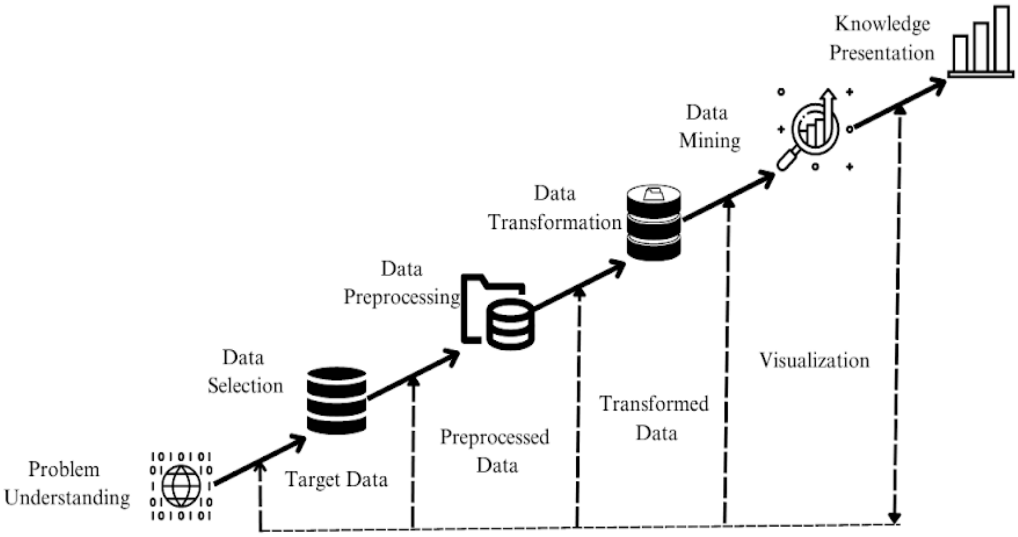


Figure 8. Knowledge Discovery in Database Process

3.1 Overview of algorithms used in this research

This section will cover the algorithms that are selected for the research. Initially, an understanding and architect of the models will be done. This process will help establish a solid foundation of the functioning and expectations from these models. Moreover, examples will be explored as to how the model deals with textual data and what features it understands to guide the sentiment analysis process. A traditional machine learning model SVM, and deep learning models namely CNN, LSTM, and BERT will be explored in detail.

3.1.1. SVM

Support Vector Machine (SVM) belongs to the family of supervised machine learning algorithms. This method uses a hyperplane to segregate the data for classification. This hyperplane that separates the 2 classes uses criterion such as variance between class and variance within class, they are maximized and minimized respectively (Thompson et al., 1974).

Cortes and Vapnik, (1995) explains hyperplane as a linear function between (vectors) 2 classes. The author goes on to mention that for a hyperplane, there is a need for support vectors, they help in determining the margin. According to (SEPIDEH PAKNEJAD, 2018) SVM performed at an accuracy of 93%, which indicates that it has a better potential in binary classification tasks for sentiment analysis. The dataset of amazon reviews was utilised in their research which is like this experiment. As a result, not exactly the similar score can be obtained but it is foreseeable that SVM is better fit than other traditional approaches.

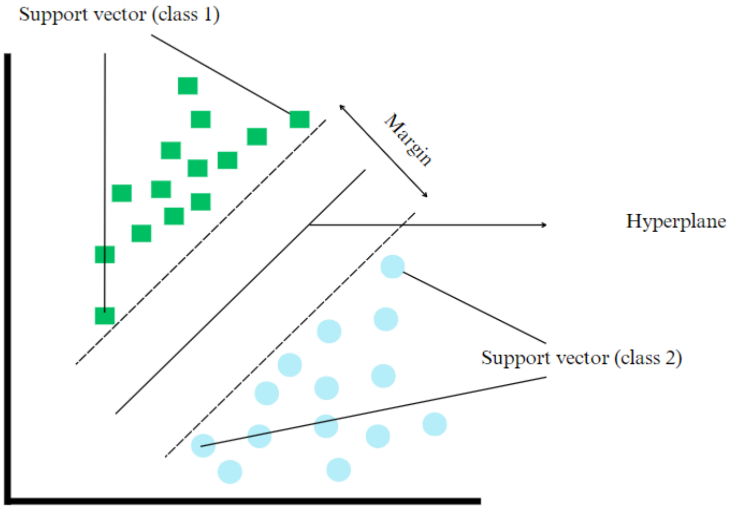


Figure 9. SVM Hyperplane representation

a. **Hyperplane**: A plane separating(distribution) 2 classes in a given space.

b. **Margin**: Line parallel to the hyperplane on each side. Separates positive class and negative class

c. **Support Vectors:** Points, positive or negative which are closest to the margin respectively.

3.1.2. LSTM

Long short-term memory (LSTM) is a type of RNN that aims to address the vanishing gradient problem encountered in traditional RNNs. Vanishing gradient arises when back propagation comes into the picture. Weight of the neural network is updated via gradient values. Vanishing gradient is when there is shrinkage of the gradient as it is propagating through time. There is less contribution from a gradient if its values have become very small. Short-term memory is a problem faced by RNN (Recurrent Neural Networks). RNN will struggle to transfer the information from earlier steps or later coming ones if the sequence is too protracted. As a result, RNN may exclude information from the start if a paragraph of text is processed for predictions (Michael Phi, 2018). LSTM introduces memory cells; these cells are used to store selective data. As a result, they are able to maintain long-term dependencies.

The LSTM architecture incorporates these memory cells by introducing 3 types of gates: the input gate, the output gate, and the forget gate. These gates govern the information flow within the network, selectively modifying the memory cell and influencing the output based on the current input and the content stored in the memory cell.

1. Forget Gate

   First gate as highlighted in the figure xx, if the forget gate. What information should be kept and what should be forgotten (thrown away) is decided by this gate. Information from 2 states, previous hidden and current input states are passed through a sigmoid function. Sigmoid function squishes these values between the range of 0 and 1. The value nearer to 0 is thrown away and the value nearer to 1 is kept.

2. Input Gate

   This gate is essential for updating the state of the cell. Current input and previous hidden state get passed through a sigmoid function. The values are converted between 0 and 1 in order to decide which values are to be kept and thrown away. Further, the current input and previous hidden state are passed through a tanh function. This transforms the values between -1 and 1 which helps in regulating the network. These 2 values are passed to a plot wise multiplication, wherein the output from sigmoid decides which information from the tanh output is to be kept

3. Cell state

   A key part of LSTM is the cell state. The forget vector is first multiplied pointwise by the cell state. If multiplied by values close to 0, this could result in the cell state losing values. Then, using a pointwise addition, we update the cell state with the latest values that the neural network deems pertinent by taking the output from the input gate (Michael Phi, 2018).

4. Output Gate

   Lastly, there is an output gate. It decides the values for the next hidden state. Current input and previous hidden state are passed through a sigmoid function. The newly generated cell state is passed through a tanh function. The output of these 2 is then passed through a pointwise multiplication. Here, the sigmoid output decides the information to kept form the tanh function, that will be the hidden state for the next step.

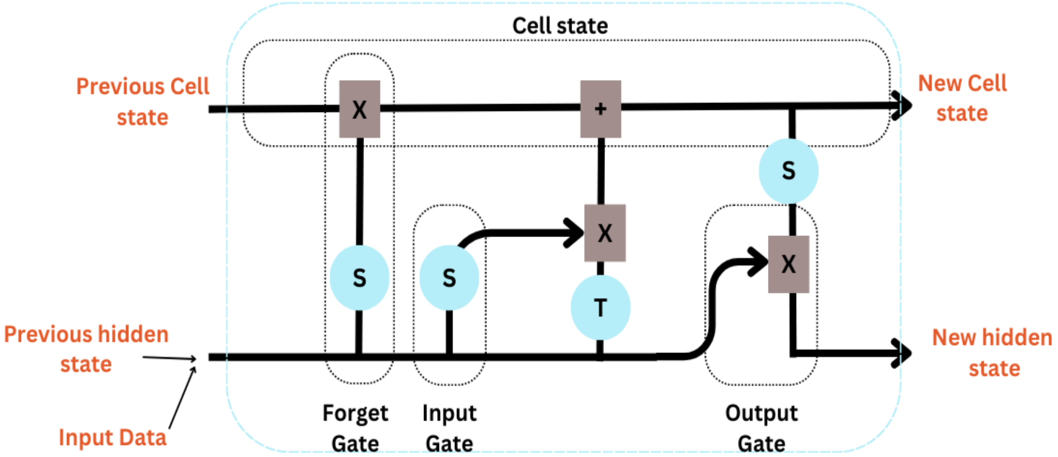


Figure 10. Architecture if LSTM (Michael Phi, 2018)

S = Sigmoid (Sigmoid activation)
T = Tanh (Tanh activation)
X = Plotwise multiplication
+ = Plotwise addition

The Sigmoid activation is the same as tanh activation (as it exists in RNN). Sigmoid squishes the values between the range of 0 and 1. It is necessary to forget or update the data. When a number is multiplied by 0 obviously the output is 0m and when multiplied by 1 then the same value. The output with 0 is ‘forgotten’, the output with 1 is ‘kept’. As a result, the least important data is forgotten and the important data is kept. The tanh activation squishes the values between the range of -1 and 1 (Michael Phi, 2018).

LSTM's ability to selectively retain, discard, and update information within the memory cell allows it to effectively capture long-range connections in sequential data, addressing the issue of vanishing gradients. This makes LSTM networks highly suitable for tasks involving sequences, such as speech recognition, time series prediction, and natural language processing. In summary, LSTM is a specialized type of recurrent neural network that employs memory cells and adaptive gating mechanisms to overcome the limitations of traditional RNNs. This unique architecture enables LSTM to capture long-term dependencies in sequential data, making it particularly well-suited for various applications (Hochreiter and Schmidhuber, 1997). In simpler terms, the working of LSTM takes place in such a way that for example, a customer wants to buy a guitar from amazon. This customer will initially go through the vast number of reviews from that present for that said Guitar. If the customer comes across a review such as, “**Amazing!** This guitar is **perfectly tuned**. I just got it, and will **recommend it to more people**.” The customer might not remember the whole product review, but only the highlighted part. The same information that customer will pass on to, if someone asks them for the review. That is basically what LSTM accomplishes.

3.1.3. CNN

A Convolutional Neural Network (CNN) is a type of deep learning model. It is specifically designed to process images or sequences i.e., grid-like data, similarly, this can be applied to work vectors stacked on each other to form an “image” (Rita Kurban, 2019**).** The input in case of NLP is a sentence or document that is represented as a matrix, where each row corresponds to a word represented as a vector. These vectors are typically word embeddings, which are compact representations of words in a lower-dimensional space. CNN is capable of learning and extracting hierarchical representations of features, capturing temporal and spatial patterns. This is possible by leveraging the convolutional layer. The convolutional layers form the core building blocks of a CNN. These layers consist of kernels a.k.a. set of learnable filters that perform operations on the input data. Each filter is responsible for detecting specific patterns or features such as edge, shape, or texture (Denny Britz, 2015).

After the convolutional layers, pooling layers come into the picture. The pooling layer reduces the spatial dimensions by subsampling their input. A common type of pooling applied is max pooling (max operation). When max pooling is applied to a region, it focuses on the most prominent feature of that region, whereas it discards the rest. It retains the important information of whether or not the feature appeared in a sentence. By doing this it ignores the less important details and also helps make computation faster. Max pooling reduces spatial information, meaning it does not save the exact location of the feature or the most prominent feature. But, it collectively knows that it exists in that locality. As a result, even if there is a slight change to that feature the model is able to recognize it irrespective of its position. This helps CNN deal with variations in the data. Ultimately, max pooling results in a better performance, as it helps the network deal with variations, reduces computation load, and retains important features effectively (Denny Britz, 2015).

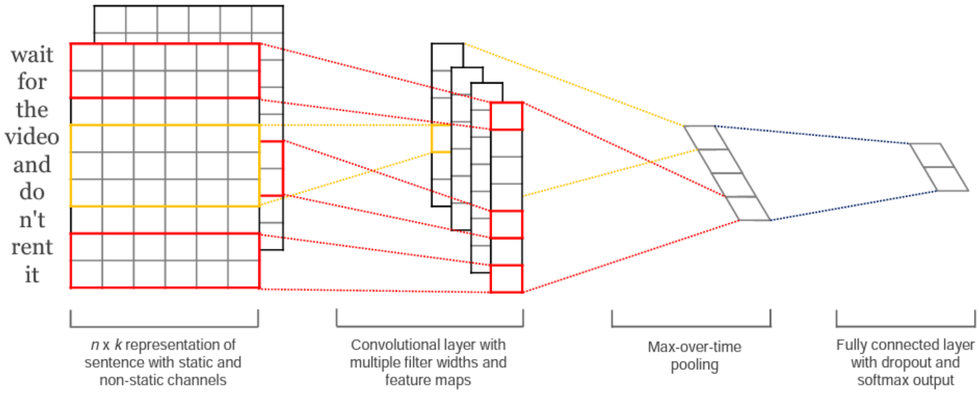

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 11. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification

Non-linear relationships can involve complex patterns and dependencies in the data that cannot be represented using linear transformations. Thus, to adequately capture these, CNN has activation functions. The functions introduce non-linearity which allows the network to represent complex relationships. Some of the common activation functions are ReLU (Rectified Linear Unit), Sigmoid, and SoftMax.

3.1.4. BERT

BERT stands for Bidirectional Encoder Representations from Transformers, a paper published by Google AI Language researchers. An approach followed by the traditional models is either from right to left or left to right. Whereas BERT is a combination of both, hence the name bidirectional. It simultaneously trains the model in both directions. BERT utilizes a specific model called a Transformer, which has the capability to focus on different sections of the input text. By training

BERT bidirectionally, it obtains a more profound comprehension of the language's context and progression in comparison to models that analyze only a single direction (Devlin et al., 2019).

In order to tackle the limitations of the single-directional approach, the researchers at Google AI Language introduced 2 strategies for training: Masked LM (MLM) and Next Sentence Prediction (NSP). Masked LM, involves randomly hiding or masking a word in the sentence and training the model to figure out the missing piece (mask) on the basis of its surrounding sentence. For example, "**The dog is brown and the dog is hungry**", the model will mask "brown", thus making the sentence "**The dog is** ***[MASK]*** **and the dog is hungry**", and train the model to figure out the missing word based on the context. Through this process, BERT is able to grasp the relationship between different words and is able to predict missing words effectively. This approach enables BERT to understand the overall message of a sentence, even if some words are hidden or removed.

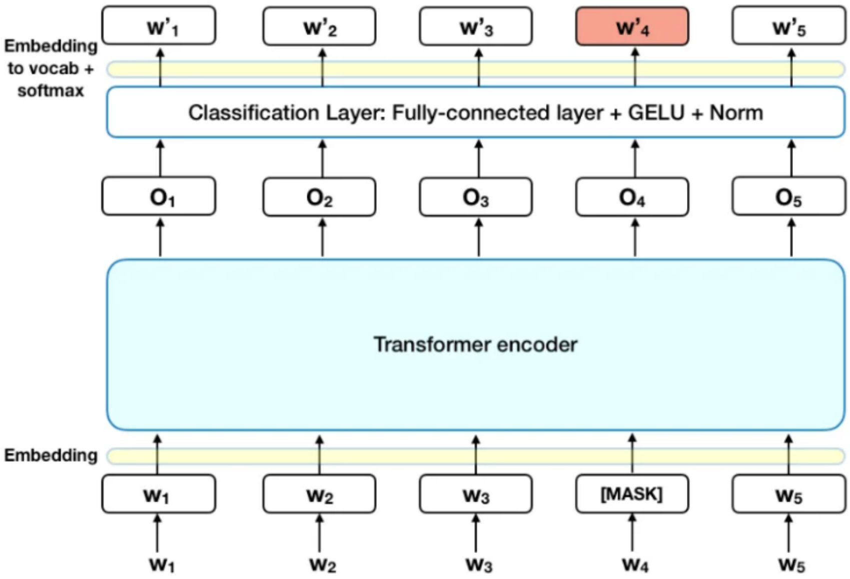

W'1
W'2
W'3
W'4
W'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

Figure 12. Architecture of BERT (Rani Horev, 2018)

Next Sentence Prediction (NSP) enhances the ability of BERT to understand the dependencies between sentences. It involves teaching the model to determine if two sentences in a pair are in a logical sequence or not. During training, BERT is given pairs of sentences and asked to determine if the second sentence logically follows the first one. For instance, consider a sentence pair: "The dog is hungry. It is raining outside." The BERT model needs to recognize that the first and the second sentence are not related to each other. Thus, training BERT on such tasks helps it grasp the relationship between sentences, the flow of information, and how the different parts of a sentence connect with each other. NSP is valuable for tasks like document summarization, questioning, and answering as understanding the relationship of the words and sentences matters.

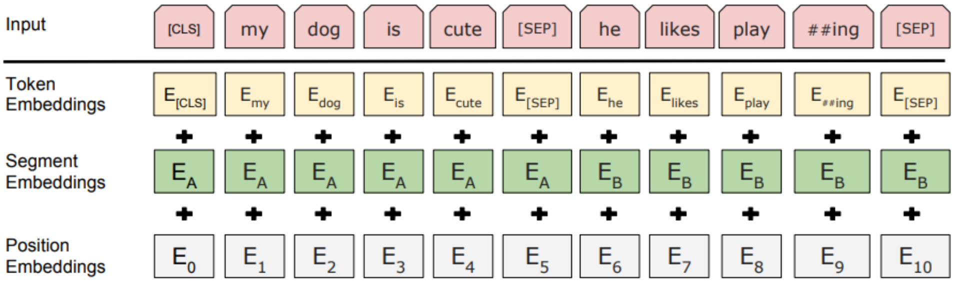


Figure 13. BERT input representation. (Devlin et al., 2019)

3.2. Overview of Evaluation Metrics

In this section, there is a presentation of evaluation metrics that will be used to assess the performances and other aspects of the models that will be used for sentiment analysis. These metrics play a crucial in quantifying the effectiveness and accuracy in predicting the sentiment labels for the reviews data.

Accuracy

Accuracy is an evaluation metric that determines the overall correctness of predictions by calculating the proportion of correctly predicted instances out of the total number of instances. It is commonly used for balanced datasets, but its reliability may diminish when dealing with imbalanced classes. Accuracy is calculated using equation 3 as below.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Equation 3. Accuracy

Precision

Precision is an evaluation metric that quantifies the ratio of accurately predicted positive instances to the total instances predicted as positive. Its emphasis lies in minimizing false positives, making it valuable when the cost of false positives is significant. Precision is calculated using equation 4 as below.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Equation 4. Precision

Recall

Recall, also known as sensitivity or true positive rate, is an evaluation metric that assesses the ratio of accurately predicted positive instances to the total actual positive instances. Its significance lies in situations where the cost of false negatives is substantial, as it focuses on reducing the number of missed positive instances. Recall is calculated using equation 5 as below.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Equation 5. Recall

F1 Score

The F1 Score is an evaluation metric that represents the balanced measure between precision and recall. It is calculated as the harmonic mean of precision and recall. The F1 Score is particularly valuable when there is an imbalance between the classes, ensuring that both precision and recall are considered in the evaluation. Recall is calculated using equation 6 as below.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Equation 6. F1 Score

Confusion matrix

A confusion matrix helps assess how a machine learning model is performing in predicting different classes. It helps to understand and evaluate the performance of a machine learning model in a classification task. This matrix usually comprises of 4 categories:

- True Positives (TP): These are instances where the model is capable of correctly predicting the positive class.
- True Negatives (TN): These are instances where the model is capable of correctly predicting the negative class.
- False Positives (FP): These are instances where the model incorrectly predicts the positive class whereas the real result is negative. It's also known as a Type I error or false alarm.
- False Negatives (FN): These are instances where the model incorrectly predicts the negative class whereas the real result is positive. It's also known as a Type II error or miss.

	(Predicted) Negative	*(Predicted) Positive*
(Actual) Negative	*TN*	*FP*
(Actual) Positive	*FN*	*TP*

Figure 14. Confusion matrix

By examining these categories, we can evaluate the model's performance and calculate various metrics such as accuracy, precision, recall, and F1 Score. The confusion matrix helps us understand how well the model is distinguishing between different classes and making correct predictions.

3.3. Data selection

In the data selection phase, careful consideration was given in order to select an appropriate dataset. The data used for the research purpose is sourced from Amazon. The dataset selection process was driven by its alignment with the research objectives, considering the significance of Amazon product reviews as a valuable resource for sentiment analysis. The dataset consisted of a diverse range of reviews covering different products belonging to the Music Category of products. Furthermore, the availability of sentiment labels was taken into consideration, as labelled data plays a vital role in training and evaluating sentiment analysis models. Stringent measures were implemented to ensure the dataset's integrity, involving thorough data cleaning, and preprocessing to eliminate any noise or irrelevant information.

Attribute name	Attribute description
reviewerID	ID of the reviewer, e.g. A2SUAM1J3GNN3B
asin	ID of the product, e.g. 0000013714
reviewerName	Name of the reviewer
vote	Helpful votes of the review
style	A dictionary of the product metadata, e.g., "Format" is "Hardcover"
reviewText	Text of the review
overall	Rating of the product
summary	Summary of the review
unixReviewTime	Time of the review (Unix time)
reviewTime	Time of the review (raw)
image	Images that users post after they have received the product

Table 2. Dataset attributes (amazon e-commerce consumer reviews) (Ni et al., 2019).

The reviews are from the time period of May 1996 - Oct 2018. The above table represents the dataset attributes. The dataset is available publicly and provided by Ni et al., (2019).

3.4. Summary

From the methodology section, it is clearly understood how the selected algorithm will function with example data. Followed by an overview of evaluation metrics that will be utilised in order to rank the models. Finally, the selected dataset will be pre-processed, which will be discussed more in the next sections. This process and result will be explored in further section of Empirical findings.

4 Experimental setup and Empirical Findings

This section particularly shows the setup used in this research. An overview of the technical design and project architecture employed in the implementation of this research project. Initially the section covers the resources that were utilized for this research. Following this, the results will be displayed for the above selected models.

Furthermore, the section will cover the results obtained by different models. These results with the help of evaluation metrics will help determine the best model. Ultimately, there will be an understanding as to which feature extraction method performs better in combination with the selected machine learning models.

The figure below provides a basic guideline for the flow of the research experiment.

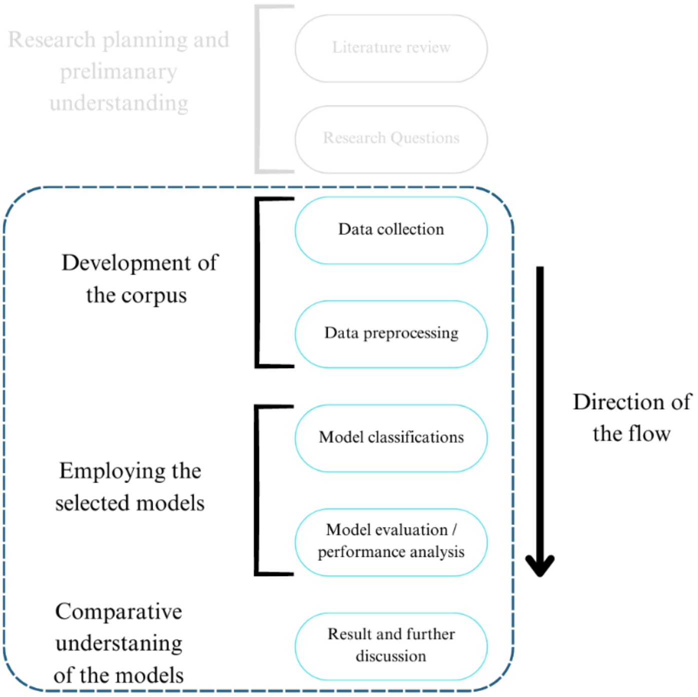

Research planning and prelimanary understanding
Literature review
Research Questions
Development of the corpus
Data collection
Data preprocessing
Employing the selected models
Model classifications
Model evaluation / performance analysis
Comparative understaning of the models
Result and further discussion
Direction of the flow

Figure 15. Basic guideline for the flow of the research experiment

Data Collection: Raw data is obtained from an e-commerce platform, such as Amazon reviews.

Data Preprocessing: The text data undergoes cleaning, tokenization, and removal of stop words and special characters.

Model Selection: Several machine learning and deep learning models are evaluated, including SVM, LSTM, CNN, and BERT in the model setup section.

Model Evaluation: Performance metrics, such as accuracy, precision, recall, F1 score, and AUC ROC score, are used to assess the effectiveness of each model.

4.1 Design Specification

4.1.1 Overview of Architecture

The research process utilizes data from Amazon Customer Reviews. This is publicly available data and is enormous in size. Keeping in mind the limitation of the research process in terms of resource availability and the resource-intensive nature of processing such an amount of data, a randomly selected subset of that data is taken into consideration.

Below is a high-level representation of the architecture which will support the implementation of this research. It is divided into 3 levels.

Presentation level: This level consists of the visualization and representation of the results and insights that this project will deliver.

Logic/Machine Learning level: The logic tier comprises the Deep learning model that will be used to carry out sentiment analysis.

Data level: The last layer comprises the raw data that will be used for the research experiments and different tools and technologies to retrieve, manipulate and store this data.

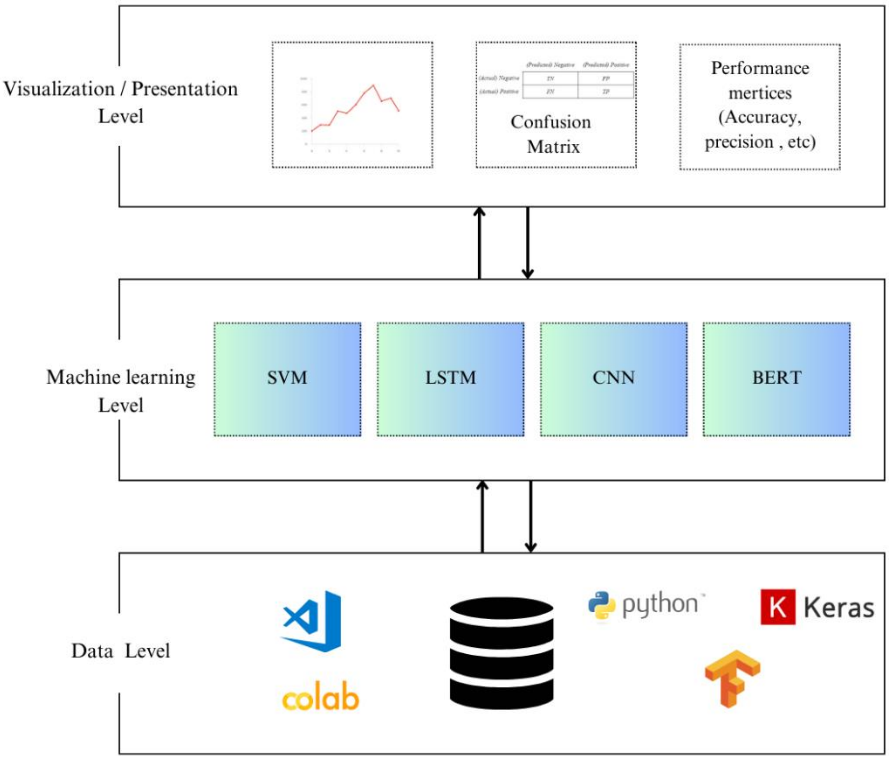

Visualization / Presentation Level
Confusion Matrix
Performance mertices (Accuracy, precision , etc)
Machine learning Level
SVM
LSTM
CNN
BERT
Data Level
colab
python
Keras

Figure 16. Overview of the project architecture

4.1.2 System configuration

In this research, the system configuration involved using Google Colab, a cloud-based Jupyter Notebook environment, to conduct the experiments. The programming language used for this purpose was Python, along with different Machine Learning (ML) libraries and deep learning frameworks suitable for Natural Language Processing (NLP) tasks were utilized.

Table xx describes the system configuration

Operating System	Windows 11
Memory	8 GB
CPU	Intel® Core™ i5-10210U processor
Software	Jupyter Notebook, Google Colaboratory
Python Version	3.10.12
Keras Version	2.13.1
TensorFlow Version	2.13.0

Table 3. System configuration

The NLP tasks in this study involved applying deep learning methods such as long-term short memory (LSTM), convolutional neural networks (CNNs), and transformers, with BERT being a prominent model used. These deep learning models were implemented using the TensorFlow and Keras libraries. The experiments were conducted on a Google Colab virtual machine, which provided access to high-performance GPUs for efficient computation. By utilizing Google Colab, Python, machine learning libraries, and deep learning methods, this research aimed to take advantage of the convenience and computational power offered by these tools to perform comprehensive NLP analysis, including accurate sentiment analysis and other related tasks.

A potential methodology that can be used in comparison to KDD (Knowledge Discovery in Database) is the CRISP-DM (Cross-Industry Standard Process for Data Mining) method. CRISP-DM is more detail-oriented, iterative, and systematic as compared to KDD which provides a more high-level framework. KDD takes a more comprehensive approach by considering the entire process of knowledge discovery, which includes not only data mining but also aspects like data selection, preprocessing, transformation, and the presentation of knowledge.

In this phase of research, the pre-processed data will go through different machine learning models. Wherein, the train dataset will be used to train the said models and the test dataset will be utilized to further evaluate the model performance.

4.2 Data preprocessing

It is the next in the KDD methodology following the data selection step. In this stage we perform a preliminary investigation in the selected data. The following subsections help dive deep into this stage.

4.2.1 Data exploratory analysis

Null values

In this step, an initial analysis is carried out to check the existence of null or missing values in the dataset. It is essential to perform this check in order to verify if the target attribute has any null values which need to be addressed, so they can be handled in the cleaning process. As shown in the table 4, null values are present in 5 attributes.

Attribute name	Null values
reviewerID	0
asin	0
reviewerName	150
vote	1268582
style	839992
reviewText	855
overall	0
summary	380
unixReviewTime	0
reviewTime	0
image	1482365

Table 4. Null values in the dataset per column

Rating count plot

This step further helps us understand the distribution of the user ratings from the dataset. This gives an understanding of how the reviews are distributed overall. As observed, the 5 rating has the highest distribution close to 10,00,000. Thus, it is imminent that the dataset consists of majority positive reviews. Further, the information will be helpful in order to create a balanced dataset for training and test.

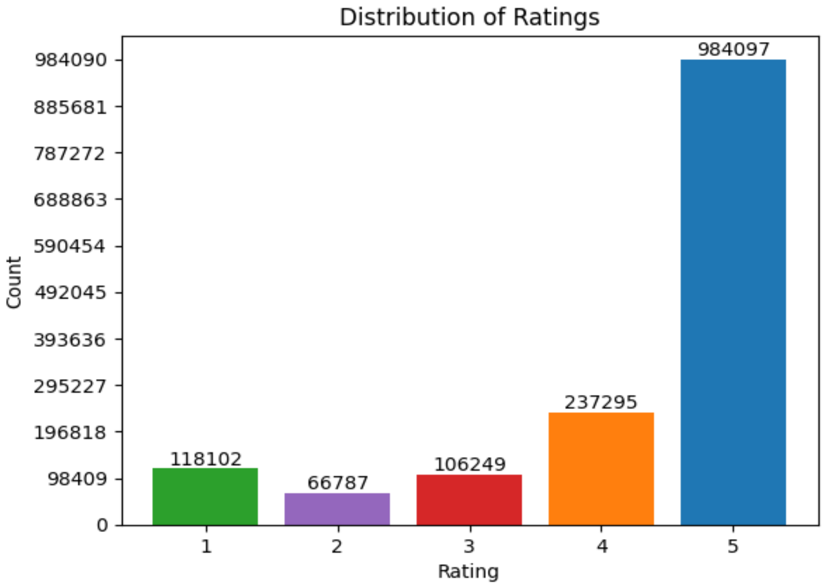


Figure 17. Distribution of Ratings

Word cloud

Word cloud helps understand the frequency of the word in the corpus. This visualization provides a summary of the most frequent words, represented by the size of the word which is directly proportional to its occurrence in the corpus. As established in the previous section, since the major contribution is of 5 rating reviews. There is an imminent possibility to have more positive words with larger size as compared to negative words.

1. Positive reviews only word cloud.

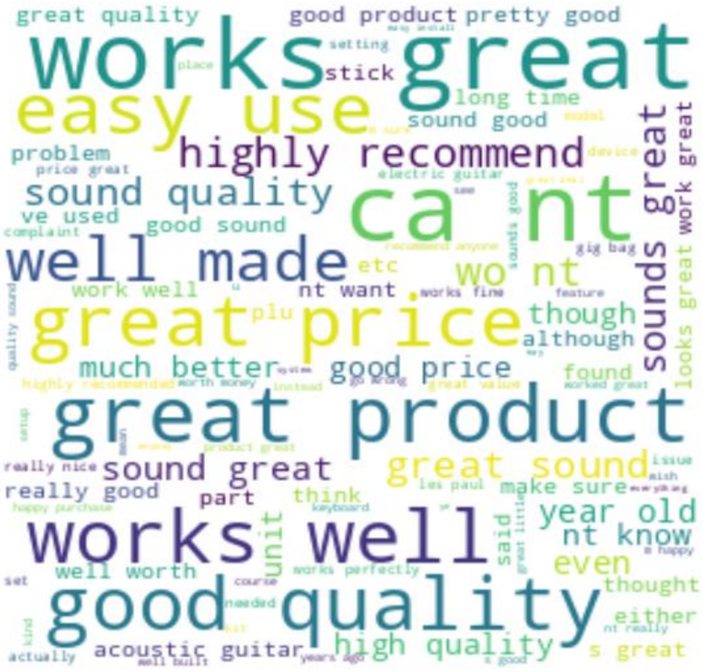


Figure 18. Word cloud of positive reviews

2. Negative reviews word cloud

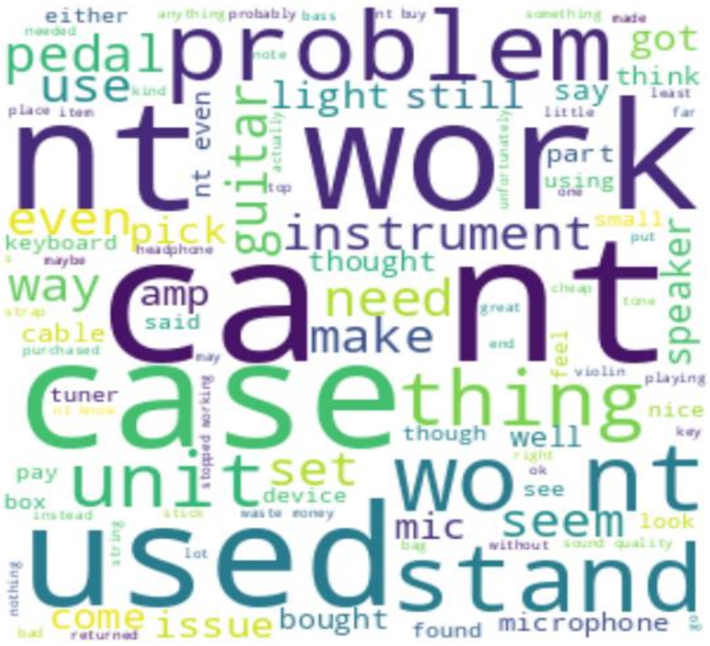


Figure 19. Word cloud of negative reviews

4.2.2 Data cleansing

Data cleaning, also known as data cleaning or data scrubbing, is the process of identifying and correcting or removing errors, inconsistencies, and anomalies in a data set. It involves a series of steps to ensure that the data to be used for analysis is accurate, complete, and reliable. Here are some common steps in the data cleansing process:

1. Handling missing values

   It is essential to handle missing values in a dataset in order to remove the discrepancies that the analysis process might face because of incomplete data. There are multiple ways to handle these missing values. One method is to replace these missing values with mean, median, or mode imputation. Some additional examples of imputation methods are SMOTE and ROSE. Moreover, removing the column with missing values is an option, if imputing these values is not a viable option. A good practice is to remove columns that have more than 55% of missing values. Imputing a large number of data can introduce bias and also distort the original distribution of the data.

   With reference to fig xx, which gives us the details of the number of missing values in the dataset. As a result, Vote, Style, and Image columns can be removed from the dataset. Furthermore, when it comes to imputing the values of the remaining columns which have missing values. In this case, imputation is not a feasible option since their data is either a unique id or string data. Thus, they can be kept as it is.

However, when considering the missing values for reviewText attribute, these missing values have to be removed from the dataset. As imputation of string data can be challenging, in this research approach the rows with missing data will be dropped in order to have a uniform column for sentiment analysis. Some missing values persist in the reviewer Name and summary columns, although for this research, these are acceptable since they do not have a direct impact on the analysis. As a result, these columns can be dropped from the dataset.

Final acceptable attributes in the dataset

Attribute name	**Null values**
reviewerID	0
asin	0
reviewText	0
overall	0
unixReviewTime	0
reviewTime	0

Table 5. Null values after cleaning the dataset

2. Dealing with duplicates

   Duplicate values can introduce a lot of disparity in the sentiment analysis process. It is thus necessary to deduplicate the data beforehand. Duplication can lead to bias in the representation of the sentiment. Because there is a lot of redundant data present it can influence the training process of the sentiment model disproportionately. Moreover, there are chances of overfitting to occur. Since the model is well-versed in a specific instance of the data.

   Performance matrices can also get inflated due to the presence of duplicate data. It will be easier for the model to predict the scores of the duplicated data, increasing the accuracy artificially.

3. Handling outliers

   Outliers are the values that significantly deviate from the data. The nature of the analysis objectives and the data, it can be decided to remove the outliers or not. This research process required two primary columns. First are the ratings which is the ‘overall’ attributes in the dataset, and secondly, it’s the ‘reviewText’ attribute that is the data the research requires for the sentiment analysis.

   During the data cleansing process, it is observed during the initial analysis that there are no outliers that need to be handled

Ratings	Value count
5	960812
4	230948
3	65005
2	103417
1	114729

Table 6. Number reviews per rating (1-5)

4. Cleaning review text

   An essential effort is made to clean the text of the Reviews. Users tend to express themselves on the online platform haphazardly. Some users can different languages and different dialects. Moreover, textual data obtained from multiple sources, such as e-commerce website pages or any form of social media can contain multiple HTML tags. These tags are irrelevant for the purpose of sentiment analysis as they carry some sort of formatting information. Thus, it is important to clean the text of these HTML tags so that the focus of the analysis can be solely on the text of the review.

   Punctuation marks such as commas, periods, and exclamation marks can be viewed as noise since they do not typically carry any sentiment information, Likewise, extra spaces, line breaks, or tabs can also be cleaned or removed since they do not contribute to sentiment analysis either.

   In order to standardize the data even more, the accented characters are also removed. Lastly, all the strings are converted to lowercase in order to achieve a greater degree of normalization of the data. Thus, it would ensure words such as *'amazing'* and *'Amazing'* are given the same treatment.

4.2.3 Data balancing

Imbalanced datasets pose a definite hurdle in sentiment analysis. Thus, data balancing becomes a crucial aspect to address this challenge. Generally, this imbalance is introduced in the dataset when one of the classes is significantly more dominant than the other, this leads to bias in the prediction of the model. This section will go through the importance, advantages, and disadvantages of data balancing, and the steps involved in balancing the dataset used for this thesis.

As seen in the figure 20, it is clear that there is a massive imbalance in the 2 classes. As a result, the step for data balancing is introduced.

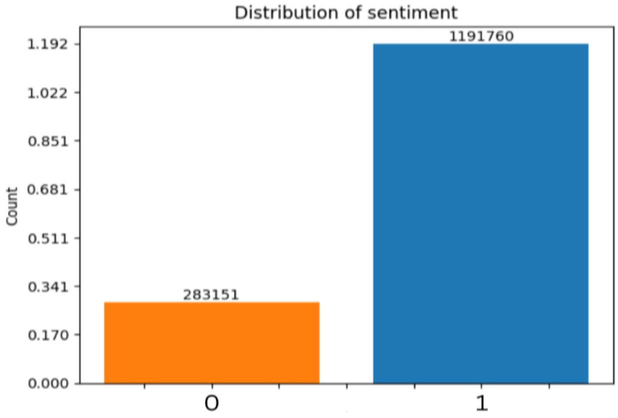


Figure 20. Distribution of sentiment Negative (0) and Positive (1) in the dataset

Importance of data balancing, points to biases introduced towards the majority classes. Such a model tends to under-represent and perform poorly with regards to the minority classes. To make unbiased predictions, the model must learn equally from all the classes present. There are various techniques for data balancing like,

1. Oversampling
   a. Random oversampling: The representation of the minority class is randomly increased by replicating the minority instances.
   b. SMOTE (Synthetic Minority Over-Sampling Technique): This technique works by synthetically increasing the instance. It does it by interpolating between the existing instances.
2. Under sampling
   In this approach, the instances belonging to the majority class are randomly removed, thus finding the balancing within the dataset. This method can also be done by identifying the centroid of the majority classes and then removing those instances.
3. Combustion sampling.
   This approach is a combination of oversampling and undersampling methods.

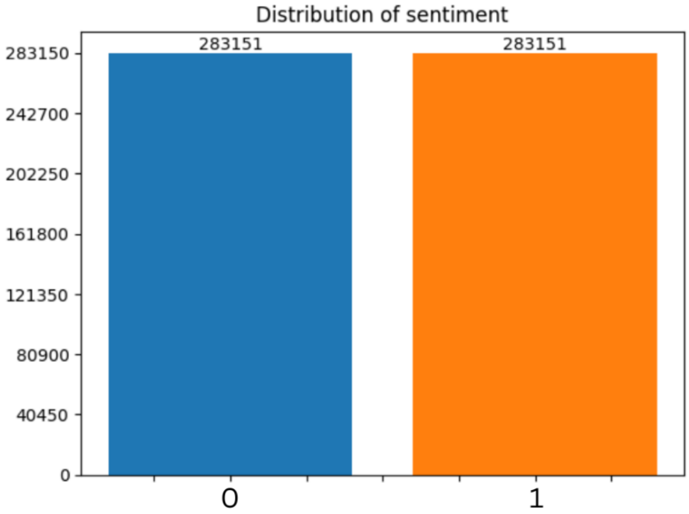


Figure 21. Distribution of Negative (0) and Positive (1) reviews after undersampling of the dataset

For this research purpose, the undersampling approach will be used. Wherein the majority class belonging to “target: 1” will be randomly oversampled in order to achieve a balanced dataset. Thus, curbing the possibilities of biases.

4.2.4 Train Test Split

One of the essential steps in any machine learning model development is the train test split. The dataset is divided into 2 subsets: training set and a test set. This split is helpful train the machine learning model and further evaluate the model’s performance.

The main objective of this partition is to have a subset of data available as unseen data for the model to evaluate. Since, this subset of data is already labelled it can be validated with the predicted label against the existing labels in order to obtain the accuracy of the trained model. A general approach is to slice the dataset by 80-20 rule, also known as the Pareto principle. 80% of the dataset is sliced as training dataset and the rest 20% as test dataset. A crucial aspect to avoid training on the test dataset as it can lead to overfitting.

Randomness and Reproducibility: It is important to ensure that the train-test split is performed randomly to avoid any bias in the data partitioning. This randomness helps in obtaining an unbiased estimate of the model's performance. Additionally, to ensure reproducibility, it is recommended to set a random seed value before performing the split.

Cross-Validation: In addition to the train-test split, you can mention the concept of cross-validation, which involves splitting the data into multiple subsets (folds) and performing multiple train-test splits. Cross-validation provides a more robust evaluation by averaging the performance across different splits and can be particularly useful when the dataset is limited.

4.2.5 Tokenization

Tokenization involves splitting or breaking the raw text into separate chunks. These chunks are called tokens. A token could be a word or a letter depending on the granularity and the requirement. For example, ‘It is a good product’ will get broken into chunks like ‘It’, ‘is’, ‘a’, ‘good’, ‘product’. These individual texts are tokens. This step is followed by Stop word removal, in order to remove the tokens which do not add any meaning to the sentence or their removal does not affect any processing (Srinivas et al., 2021). There are different libraries and methods which can be utilized to perform tokenization. Some of these are NLTK, Keras and Gensim.

Lemmatization is the process to reduce the words to their base form. Lemmatization and tokenization go hand in hand, since these tokens are converted to their dictionary form, called as ‘lemma’. For example, lemmatization of words like ‘running’, ‘drinking’, will get converted to ‘run’, and ‘drink’. This process helps accurately capture the semantic meaning of the words.

4.2.6 Feature extraction

As discussed briefly in section 2.3, feature extraction plays a vital role in sentiment analysis process. Following is an example of the working of feature extraction on the selected amazon review dataset.

For example, after extracting the features on the amazon review dataset using Word2Vec it is able to understand which group of words can belong together.

Example 1: Similar Words

Script:

```
model.wv.most_similar('guitar')
```

Output:

```
[('guitars', 0.7190992832183838), ('ukelele', 0.6840307116508484), ('taylor',
0.6734611988067627), ('mandolin', 0.6629037261009216), ('banjo',
0.6451244950294495), ('acoustic', 0.6443262696266174), ('electric',
0.6412191390991211), ('ukulele', 0.6366580128669739), ('electricacoustic',
0.6203750967979431)]
```

In above example as it is clearly visible when we pass the word ‘guitar’, Word2Vec is able to recognize which terms in the document correspond to it at what degree.

Example 2: Similarity Score

Script:

```
model.wv.similarity('guitar', 'banjo')
```

Output:

```
0.6451245
```

In above example as it is clearly visible when terms ‘guitar’ and ‘banjo’ are .64% similar which can also be cross checked from Example 1.

In Natural Language Processing, vectorization or feature extraction (Bengfort et al., between 1891 and 1894) in simple terms means that the textual data is converted into vectors (numerical representation). These vectors are easily understood for processing by the machine learning models.

4.3 Model setup and results

This section will focus on the machine learning approaches employed for the sentiment analysis experiment. 4 distinct models are explored: Support Vector Machine (SVM), Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). Each models have its specific design which helps in capturing the different essence of the sentiment and its performance can be evaluated for classification of sentiment labels

All models used for this experiment are fine-tuned on the selected dataset of Amazon customer reviews. The performances of these models will be evaluated in the later section using the different metrics such as accuracy, recall, precision, F1 score, confusion matrix and ROC curve. The model setup section aims to provide a comprehensive insight in the architecture and different parameters involved of each model. This systematic exploration will enable the research to draw meaningful conclusions about the most effective approach for sentiment classification on the selected specific dataset.

4.3.1 SVM

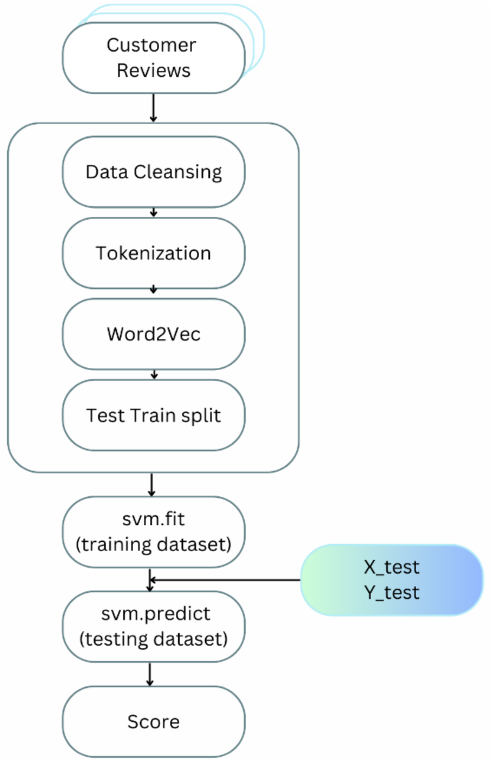


Figure 22. SVM Setup

The SVM architecture using Word2Vec embeddings involves training the Word2Vec model to learn the word representation and then create feature vectors using these embeddings. Further, the SVM classifier is trained on these vectors for sentiment classification. The effectiveness of this architecture majorly depends on the quality of embeddings done by Word2Vec. Likewise, the model with TF-IDF, the effectiveness depends on the representation done by the TF-IDF and the discriminative power that SVM can introduce. Following is the representation of the SVM architecture used for this experiment.

Parameter	Value
Kernel	Linear
Feature extractor	TD-IDF/Word2Vec
Vector Size (embedding)	100
C	Default (1.0)

Table 7. SVM model parameters

Model	Feature extraction	Accuracy	Precision	Recall	F1 Score	AUC ROC Score
SVM	TF IDF	0.853	0.857	0.848	0.852	0.925
SVM	Word2Vec	0.804	0.837	0.763	0.799	0.805

Table 8. SVM results using TF IDF and Word2Vec feature selection methods

From the above table 8 it is observed that SVM with TF-IDF achieved a 85.3% accuracy which is more that SVM with Word2Vec, which yielded 80.4% accuracy. Thus, indicating that both the version of the models were able to classify approximately >80% of the consumer reviews as positive or negative. Based on the provided results, it appears that using the SVM classifier with TF-IDF as the feature extraction method yielded the highest overall performance for the sentiment analysis task. The model achieved an accuracy of 0.853, indicating that it correctly predicted the sentiment of approximately 85.3% of the test samples.

Moreover, the SVM with TF-IDF achieved a precision of 0.857, indicating a high proportion of correctly predicted positive sentiment instances out of all predicted positive instances. The recall score of 0.848 suggests that the model effectively identified a considerable portion of the actual positive sentiment instances within the dataset. Lastly, the F1-score of 0.852, which is a harmonic mean of precision and recall, indicates a good balance between precision and recall. On the other hand, using SVM with Word2Vec as the feature extraction method resulted in slightly lower performance, with an accuracy of 0.837, precision of 0.763, and recall of 0.799. The F1-score was not provided, but it can be inferred that it may also be lower compared to the SVM with TF-IDF. In conclusion, the SVM with TF-IDF outperformed the SVM with Word2Vec in terms of accuracy, precision, recall, and overall F1-score, making it the preferable choice for sentiment analysis in this scenario.

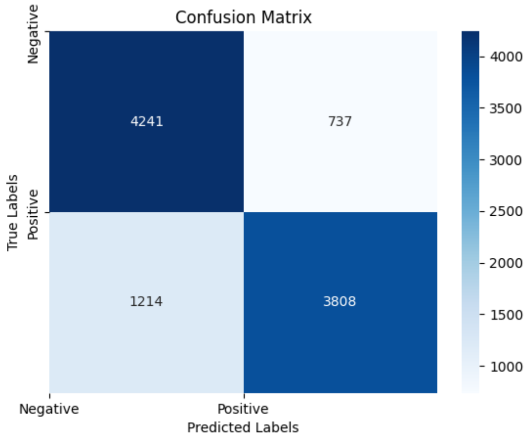


Figure 23. SVM Wrod2Vec Confusion Matrix

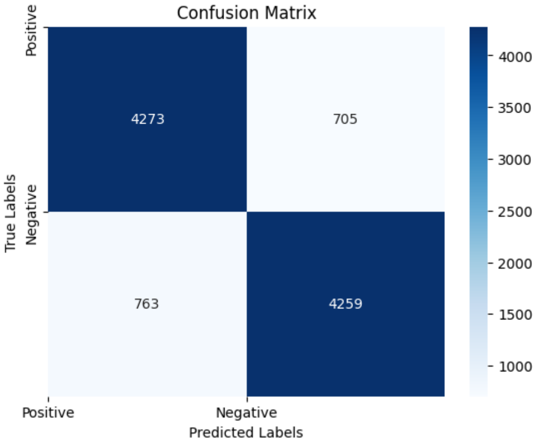


Figure 24. SVM with TF IDF Confusion Matrix

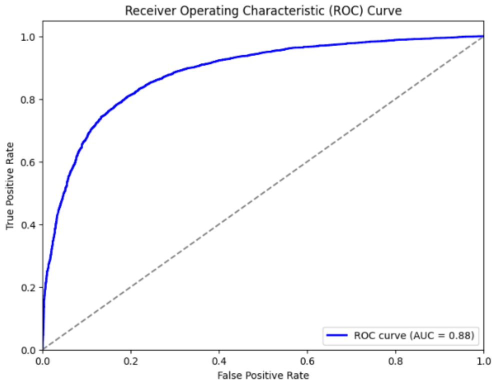


Figure 25. SVM Wrod2Vec ROC Curve

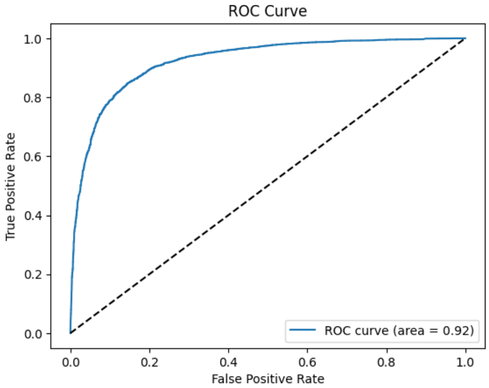


Figure 26. SVM TF-IDF ROC Curve

The side-by-side confusion matrices in figures 23 and 24 provide a deeper understanding of how both the version of the model were able to correctly identify the positives and negatives. A high number of true positives which is 3808 and true negatives which is 4241 for SVM with Word2Vec, and 4273 true negatives with 4259 true positives for SVM with TF IDF indicate that the models were proficiently able to identify the positives and negatives. However, the presence of false positives and false negatives for SVM with Word2Vec (737, 1214), and SVM with TF-IDF (705, 763) respectively indicate that there is further room for improvement for these models.

Moreover, from figures 25 and 26, it is easy to interpret that both the models can distinguish between positive and negative samples. The SVM model with TF-IDF achieved an impressive AUC of 92%, while the SVM model with Word2Vec obtained an AUC of 88%. The higher AUC score for the SVM model with TF-IDF indicates that it is better at distinguishing between positive and negative reviews compared to the SVM model with Word2Vec.

4.3.2 LSTM

It is established that LSTM is a type of RNN that is capable of capturing sequential pattern in textual data. The LSTM model for this experiment comprises of 100 LSTM units referring to 100 memory cells within LSTM, this allows the model learn and retain important data and sequential patterns. Input length is set to maximum sequence to maintain uniformity. The LSTM model architecture also incorporated a dense layer with 64 units, activated using the Rectified Linear Unit (ReLU) function, which introduced non-linearity into the network to enhance its representational power. The output layer consisted of a single unit activated by the sigmoid function, ensuring that the model produces a probability score between 0 and 1, signifying the sentiment polarity (positive or negative) of the input text.

The performance is further optimized by employing binary cross entropy loss function, it is commonly used in cases of binary classification use cases. The Adam optimizer, known for its adaptive learning rates and fast convergence, was chosen to update the model's parameters during training. During the training phase, we utilized a batch size of 32, allowing the model to update its parameters after processing each batch of 32 review samples. We ran the training process for 10 epochs, representing the number of times the model iteratively processed the entire dataset during training. In conclusion, our LSTM model with a carefully designed architecture and well-tuned hyperparameters demonstrates its efficacy in sentiment analysis for e-commerce review data

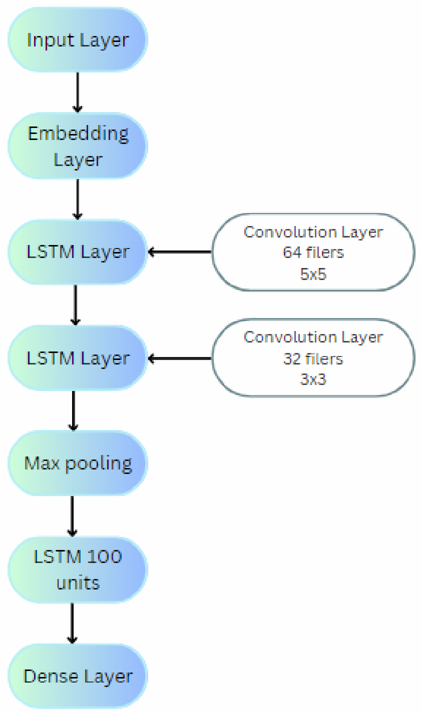


Figure 27. LSTM model setup

Parameter	Value
Input Length	Maximum sequence length
Embedding Output Dimension	100
LSTM Units	100
Dense Units	64
Dense Activation Function	relu
Output Layer Units	1
Output Layer Activation	sigmoid
Loss Function	binary_crossentropy
Optimizer	Adam
Batch Size	32
Epochs	10

Table 9. LSTM model parameters

Model	Feature extraction	Accuracy	Precision	Recall	F1 Score	AUC ROC Score
LSTM	TF IDF	0.823	0.841	0.800	0.820	0.883
LSTM	Word2Vec	0.849	0.858	0.837	0.847	0.920

Table 10. LSTM results using TF IDF and Word2Vec feature selection methods

For the LSTM model with TF-IDF, we observed an accuracy of 82.3%. This means that the model correctly predicted the sentiment of approximately 82.3% of the reviews in the test set. The precision of 84.1% suggests that when the model predicted a review as positive or negative, it was accurate 84.1% of the time. The recall of 80.0% indicates that the model successfully captured 80.0% of the positive and negative sentiments present in the test set. The F1 score of 82.0% represents a balanced measure of precision and recall, and an AUC ROC score of 88.3% shows the model's ability to distinguish between positive and negative sentiments. On the other hand, the LSTM model with Word2Vec achieved a higher accuracy of 84.9%, indicating improved overall performance compared to the TF-IDF model. The precision of 85.8% suggests that this model's predictions were accurate 85.8% of the time. The recall of 83.7% indicates that the model successfully identified 83.7% of the positive and negative sentiments in the test set. The F1 score of 84.7% represents a balanced measure of precision and recall, and the AUC ROC score of 92.0% demonstrates the model's strong ability to distinguish between positive and negative sentiments.

The side-by-side confusion matrices in figure 28 and 29, provide a better understanding of how both the version of the models were able to correctly identify the positives and negatives. A high number of true positives which is 4205 and true negatives which is 4287 for LSTM with Word2Vec, and 4018 true negatives with 4221 true positives for LSTM with TF IDF indicate that the models were proficiently able to identify the positives and negatives. However, the presence of false positives and false negatives for LSTM with Word2Vec (691, 817), and LSTM with TF-IDF (757, 1004) respectively indicate that there is further room for improvement for these models.

Moreover, from figure 30 and 31, it is easy to interpret that both the models can distinguish between positive and negative samples. The LSTM model with Word2Vec achieved an impressive AUC of 92%, while the LSTM model with LSTM obtained an AUC of 88.3%. The higher AUC score for the LSTM model with TF-IDF indicates that it is better at distinguishing between positive and negative reviews compared to the SVM model with Word2Vec.

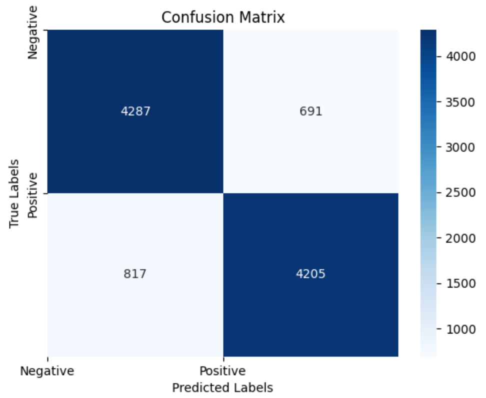


Figure 28. LSTM with Word2Vec Confusion Matrix

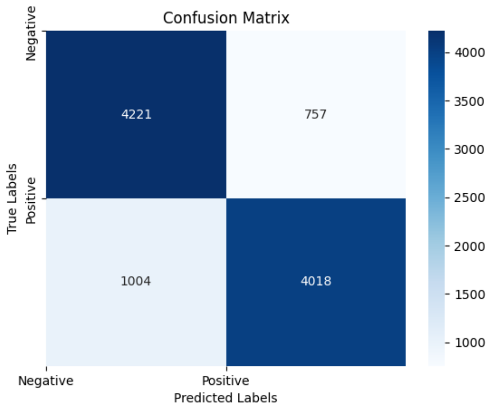


Figure 29. LSTM with TF-IDF Confusion Matrix

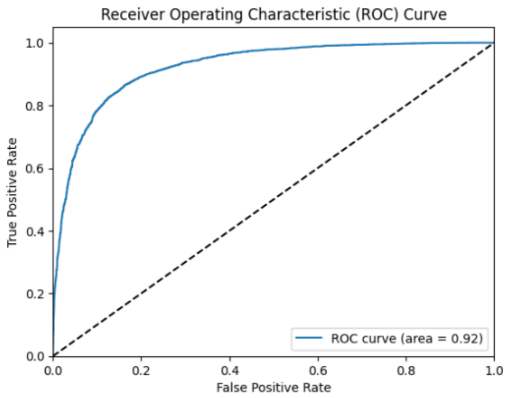


Figure 30. LSTM with Word2Vec ROC Curve

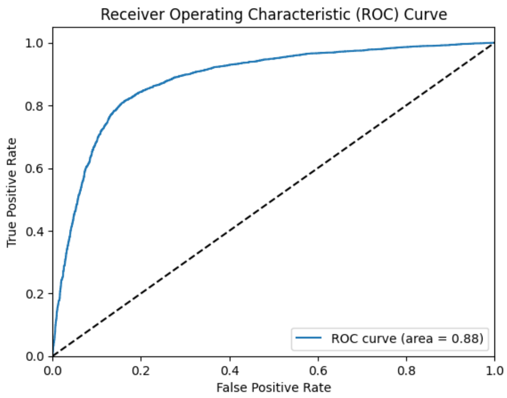


Figure 31. LSTM with TF-IDF ROC Curve

4.3.3 CNN

Below is a detailed overview of the employed setup and architecture for Convolutional Neural Network (CNN). The input shape is specified as (Number of samples, Number of features, 1), each sample representing the feature and review text that are transformed into one dimensional vector. The model consists of multiple convolutional layers, each characterized by kernel size, number of filters and an activation function. Subsequently, to down sample the learned features, pooling layers are employed with an option for wither MaxPooling, and AveragePooling.

Overfitting is handled by introducing dropout layers. The dropout rate is specified (e.g. 0.2, 0.5). Dropout deactivates random neurons during training, this encourages the network learn features which are not overly dependent on specific neurons. Adam, SGD, and RMSprop optimizers were experimented with. With a specified learning rate of 0.001 or 0.01, this controls the step size during gradient descent.

For training, binary cross entropy was employed as the loss function, widely used for binary classification. Batch size was set to 32 or 64 for updating the model parameters. The batch size depended on the compute resources availability. Throughout the training, specified number of epochs was set to make sure the model processed iteratively and uniform number of times. Finaly early stopping was employed which halted the training if an epoch where there was no improvement, this prevented overfitting.

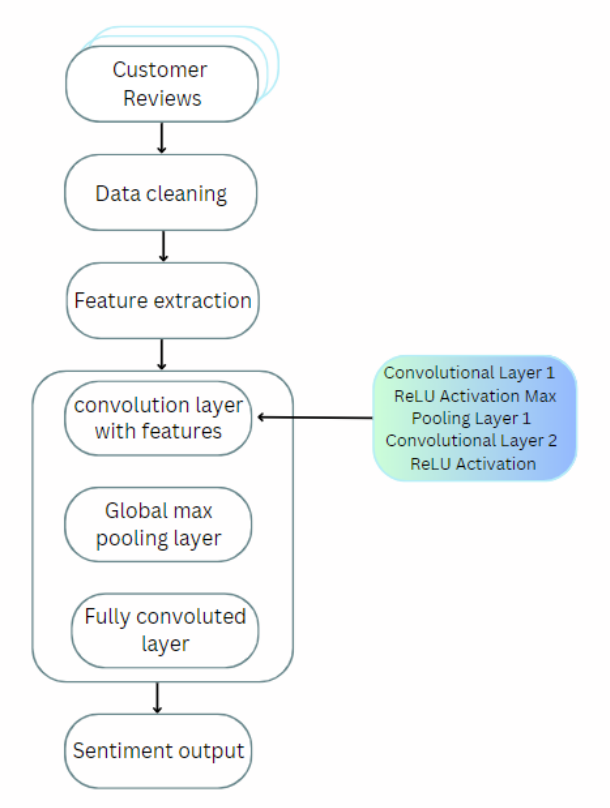

Customer Reviews
Data cleaning
Feature extraction
convolution layer with features
Global max pooling layer
Fully convoluted layer
Sentiment output
Convolutional Layer 1 ReLU Activation Max Pooling Layer 1 Convolutional Layer 2 ReLU Activation

Figure 32. CNN model setup

Parameter	Value
Input Shape	(Number of samples, Number of features, 1)
Convolutional Layers	Number, Filters, Kernel Size, Activation
Pooling Layers	Type (MaxPooling, AveragePooling), Pool Size
Dense Layers	Number, Units, Activation
Dropout	Dropout rate (e.g., 0.2, 0.5)
Optimizer	Adam, SGD, RMSprop, etc.
Learning Rate	Value (e.g., 0.001, 0.01)
Loss Function	Binary Crossentropy, Categorical Crossentropy
Batch Size	Value (e.g., 32, 64)
Epochs	Number of training epochs (10)
Early Stopping	Patience (number of epochs with no improvement)
Model Architecture	CNN architecture, custom or predefined

Table 11. CNN model setup

Table 12. CNN results using TF IDF and Word2Vec feature selection methods(below table)

Model	Feature extraction	Accuracy	Precision	Recall	F1 Score	AUC ROC Score
CNN	TF IDF	0.578	0.606	0.481	0.4	0.591
CNN	Word2Vec	0.81	0.797	0.850	0.823	0.895

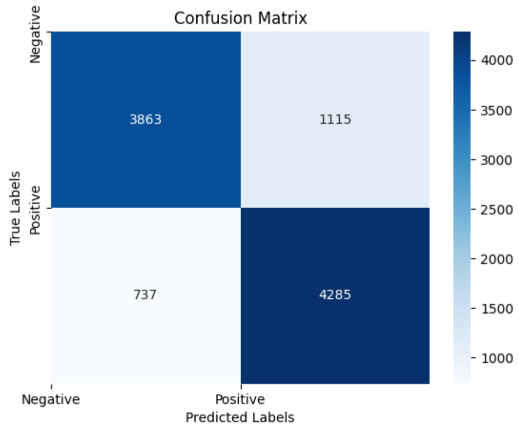


Figure 33. CNN with Word2Vec Confusion Matrix

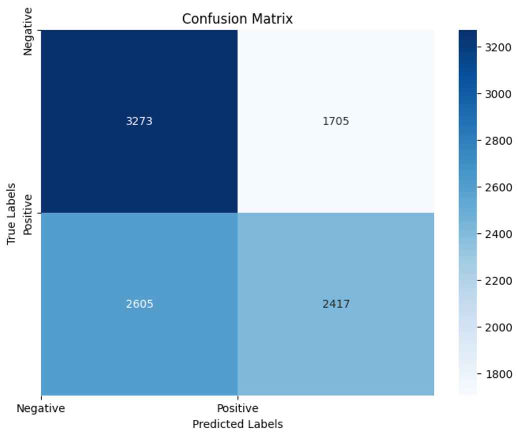


Figure 34. CNN with TF-IDF Confusion Matrix

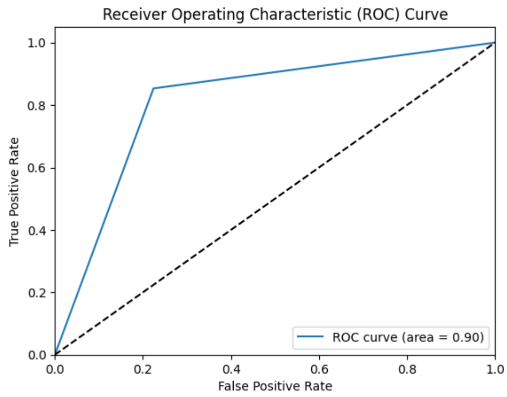


Figure 35. CNN Word2Vec ROC Curve

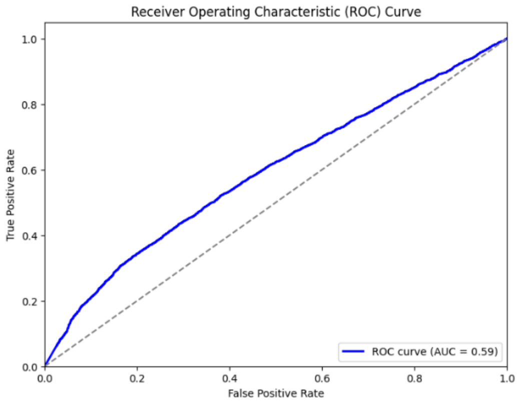


Figure 36. CNN TF-IDF ROC Curve

Based on the provided results, it is evident that the CNN model with Word2Vec as the feature extraction method outperformed the CNN model with TF-IDF for sentiment analysis. The CNN-Word2Vec achieved an impressive accuracy of 0.81, indicating that it correctly predicted the sentiment for approximately 81% of the test samples. Moreover, the model demonstrated strong precision (0.797), suggesting that it made fewer false positive predictions in classifying positive sentiment instances. The recall score of 0.850 indicates that the model effectively identified a large proportion of the actual positive sentiment instances present in the dataset. The F1-score of 0.823, a harmonic mean of precision and recall, reveals a balanced performance between the two metrics.

Most notably, the CNN-Word2Vec model attained a high AUC-ROC score of 0.892, indicating its ability to discriminate between positive and negative sentiment instances with high accuracy. In contrast, the CNN-TF IDF model showed lower overall performance, achieving an accuracy of 0.578, precision of 0.606, recall of 0.40, F1-score of 0.482, and AUC-ROC score of 0.607. In conclusion, the CNN-Word2Vec model proved to be the superior choice for sentiment analysis, outperforming the CNN-TF IDF model across all evaluation metrics.

4.3.4 BERT

BERT is a powerful pre-trained language model, it is capable of capturing contextual information from large corpora. For this BERT setup "bert-base-uncased" variant was employed. This variant is pre-trained on 12 transformer layers and 110 million parameters. The BERT tokenizer is used in the case which is capable of segmenting the input text and generates input embeddings. The model architecture adopted the “BertForSequenceClassification”, which specifically designed for classification tasks. A single output is produced by this which represents the sentiment label, the input to do the same is the tokenized input sequence.

The maximum sequence length is set to 150, this makes sure to have a uniform length and subsequently pads or truncates sentences at the time of tokenization. Binary cross entropy loss function was employed which is a standard choice when it comes to binary classification, where the aim to predict the sentiment either negative or positive. Adam optimizer was used, with a learning rate set to 0.00002 this controls the size of the step to update the model parameter while training. At the time of training the data is divided into batches, each batch containing 16 samples. It is essential to divide the data into these batches to mitigate memory constraints and in facilitating efficient gradient updates. Finally, the training processes is iterated over 10 epochs.

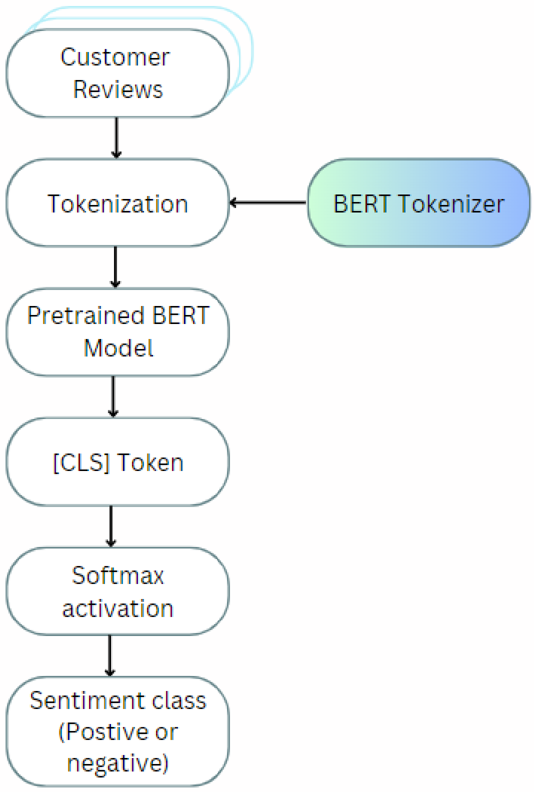


Figure 37. BERT model setup

Parameter	Value
Model Name	Bert-base-uncased (BERT pre-trained Model)
Tokenizer	Bert Tokenizer
Model Architecture	BertForSequenceClassification
Maximum Sequence Length	150
Loss Function	Binary Cross-Entropy (BCE) Loss
Optimizer	Adam
Learning Rate (Step size of Optimizer)	0.00002
Batch Size	16
Epochs	10

Table 13. BERT model setup

Model	Feature extraction	Accuracy	Precision	Recall	F1 Score	AUC ROC Score
BERT	BERT	0.863	0.838	0.903	0.869	0.928

Table 14. BERT sentiment analysis performance result

The above results obtained by the BERT model are indicative of its strong performance in this capability to classify sentiment for the provided amazon ecommerce dataset. The accuracy of 86.3% suggests that the model can correctly predict the sentiment label for a significant portion of the review samples. This high accuracy indicates the model's effectiveness in understanding and capturing the underlying sentiment patterns present in the text data.

A recall value of 90.3% indicates that the model can identify 90% of actual positives, a high recall essential when it comes to sentiment analysis as it helps avoid false negatives. The F1 score of 86.9%, which is also a harmonic mean of precision and recall suggests that the model strikes a good balance in appropriately classifying positive sentiments.

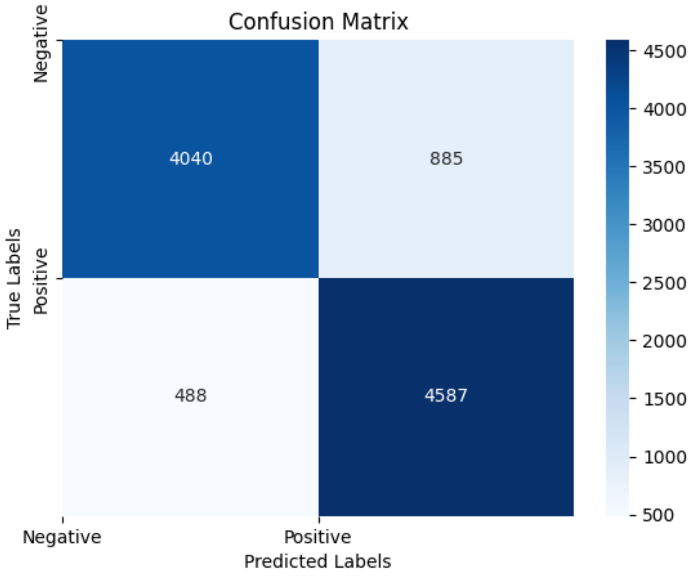


Figure 38. BERT confusion matrix

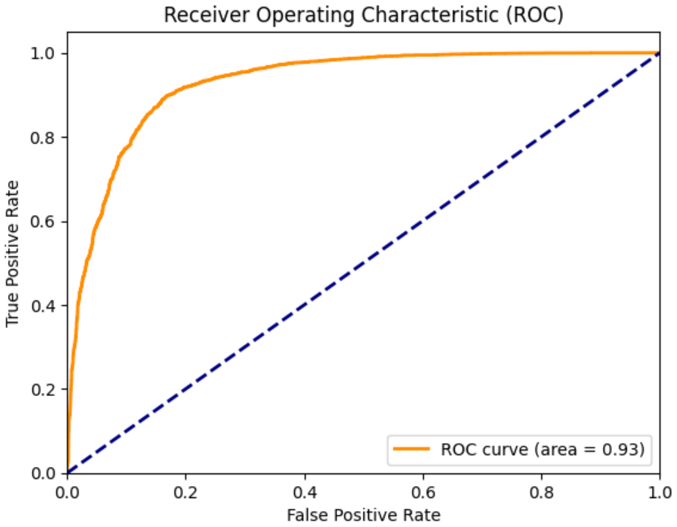


Figure 39. BERT ROC Curve, AUC = 93%

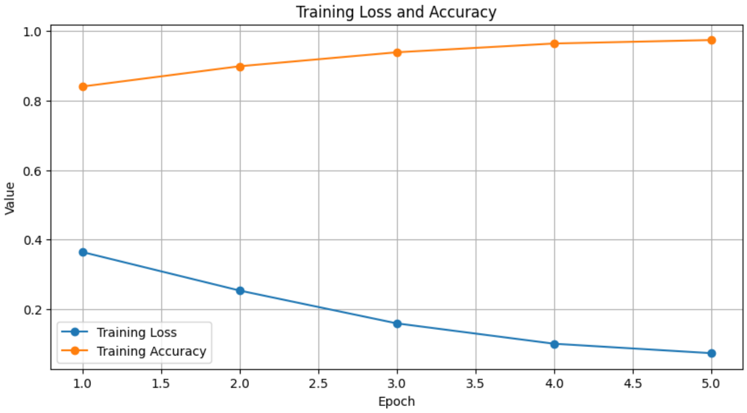


Figure 40. BERT Training loss and training accuracy over 5 epochs

The AUC ROC score is an essential metric for binary classification as it indicates the model's capability to distinguish between positive and negative classes of sentiment. 92.8% score means the higher is the power of this model to discriminate between the 2 sentiment classes, leading to confident predictions.

4.4. Performance comparison and addressing the research questions

In the ‘Performance evaluation’ section there will a comparison and analysis of the results of the trained models. An assessment of each model on the test dataset using the different evaluation techniques. Additionally, the purpose of this section to gain deeper insight in the effectiveness of the models in achieving the best accuracy.

In the below table are the performances of SVM, LSTM, CNN, and BERT to understand which model yields the best result for the specific dataset and the task of sentiment analysis.

Model	**Feature extraction**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC ROC Score**
SVM	TF IDF	0.853	0.857	0.848	0.852	0.925
SVM	Word2Vec	0.804	0.837	0.763	0.799	0.805
LSTM	TF IDF	0.823	0.841	0.800	0.820	0.883
LSTM	Word2Vec	0.849	0.858	0.837	0.847	0.920
CNN	TF IDF	0.578	0.606	0.481	0.400	0.591
CNN	Word2Vec	0.81	0.797	0.850	0.823	0.895
BERT	**BERT**	**0.863**	**0.838**	**0.903**	**0.869**	**0.928**

Table 15. Final evaluation matrix, comparing performances of all models

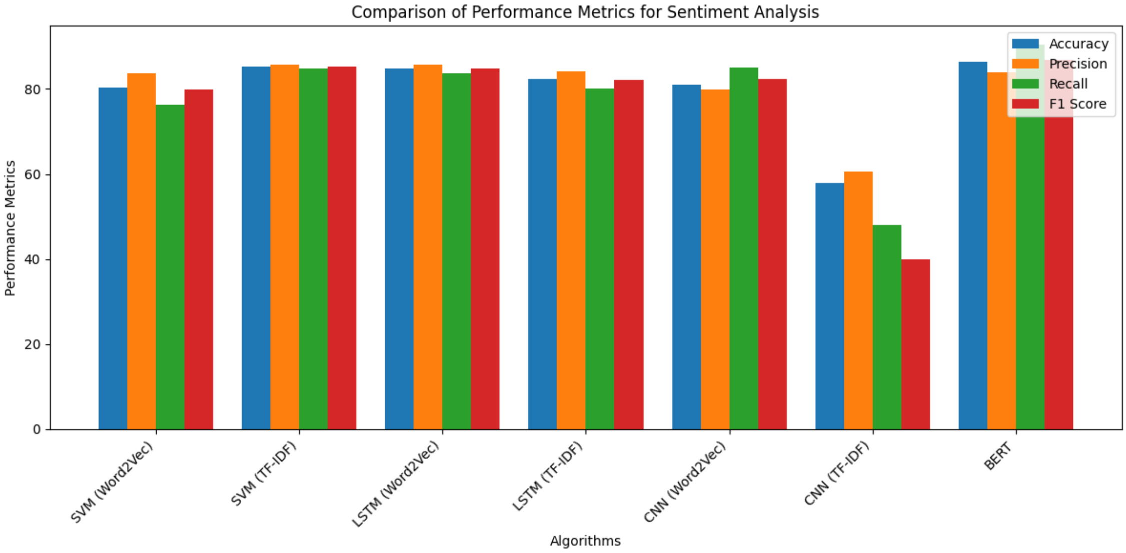


Figure 41. Comparison of performance metrics for sentiment analysis

Table 15 and figure 41, give a good overview to compare the different models employed for sentiment analysis for this research. This enables the research to answer the research questions posed in section 1.2. Accuracy, will serve as the most pivotal score to determine of these has the best performance. From a quick glance of the table, BERT has performed exceptionally well in comparison to other methods.

Furthermore, answering the research question will help dive deep into the performances of these models.

RQ1: *How do different machine learning models compare in terms of their performance and effectiveness for sentiment analysis on e-commerce review data?*

Based on the comparison done in section 4.4 with various machine learning models to perform sentiment analysis on the selected Amazon ecommerce review dataset, the research has pointed that BERT outperformed other included models. In the experiment BERT achieved the highest accuracy of 86.3%, this goes on to showing that it has a better classification capability of the customer reviews. Furthermore, BERT solidifies its superiority by remarkable demonstration of precision, recall and F1score.

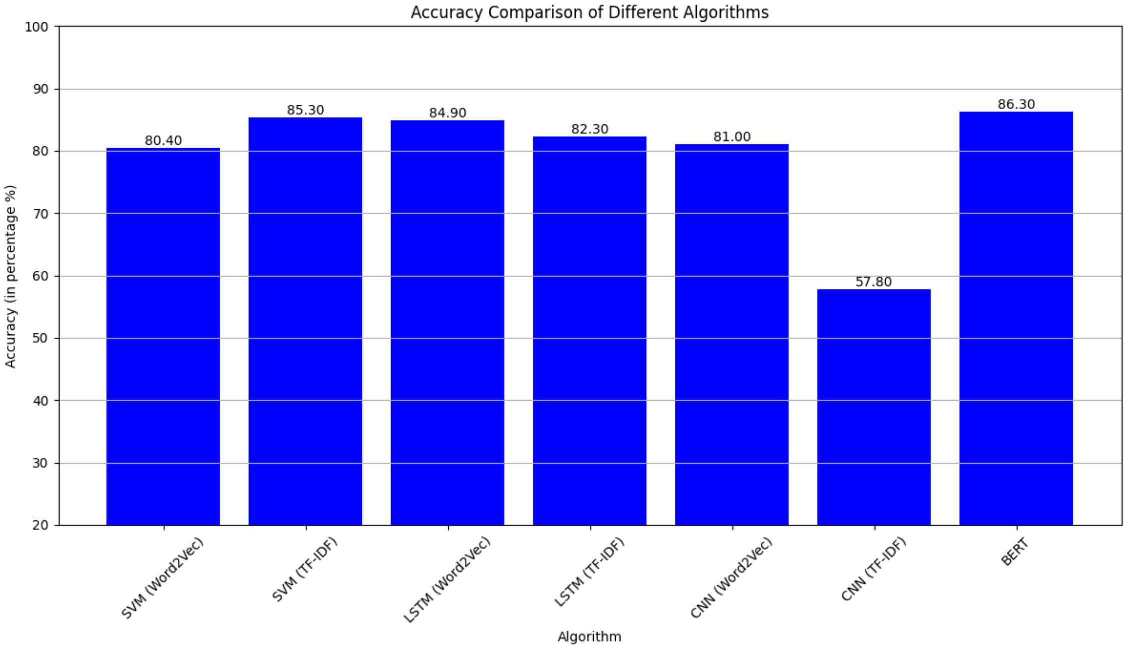


Figure 42. BERT with highest accuracy compared to other models

SVM with TF-IDF ranked second in performance, achieving an accuracy of 85.3%. While it showcased slightly lower Precision, Recall, and F1 Score compared to BERT, it still delivered commendable results in sentiment analysis.

LSTM with Word2Vec secured the third position with an accuracy of 84.9%. It exhibited high Precision and Recall, but the F1 Score was slightly lower compared to BERT and SVM with TF-IDF. LSTM with TF-IDF ranked fourth, achieving an accuracy of 82.3%. While it showcased competitive Precision and Recall, the F1 Score was slightly lower compared to the top-performing models.

CNN with Word2Vec performed at the fifth position, achieving an accuracy of 81.0%. It demonstrated good Precision and AUC ROC Score but lagged behind in terms of Recall and F1 Score. CNN with TF-IDF had the lowest overall performance, achieving an accuracy of 57.8%. This model exhibited the lowest Precision, Recall, F1 Score, and AUC ROC Score among all the models evaluated.

RQ2: *Which feature extraction method is best suited for sentiment analysis of Amazon reviews (TF-IDF or Word2Vec)?*

In the context of sentiment analysis for the amazon customer reviews, BERT has performed tremendously in comparison to the traditional feature extraction methods like Word2Vec and TF-IDF. However, one of the aims of this research was to determine which feature extraction between Word2Vec and TF-IDF would perform the best. It is evident that in comparison Word2Vec has performed better in most cases.

LSTM with Word2Vec achieved a higher accuracy of 84.9% compared to LSTM with TF-IDF, which attained an accuracy of 82.3%. Similarly, CNN with Word2Vec outperformed CNN with TF-IDF, with accuracies of 81% and 57.8%, respectively.

4.4 Summary

In section 4, the research has implemented the theory that was discussed in the earlier sections. The result of the models is based on their performance and the model setup. In order to have a fair evaluation the models used with Word2Vec and TF-IDF were not altered, only their feature extraction methods were changed. As a result, the final matrix in table xx gives the comparison of the chosen models, where BERT comes out as the top performer amongst the rest.

5 Conclusion

In this thesis, main focus is on the sentiment analysis i.e. the classification of consumer reviews, along with challenges faced in sentiment analysis. A comparative study, portraying the capability of different machine learning models is reported in this literature. As a part of this thesis's research, four different machine learning models are discussed to add to the growing knowledge of classification accuracy of consumer reviews.

Dataset of amazon consumer reviews is used for this research. It is a labelled dataset with reviews ranging from 1 to 5 stars. For this research reviews with 0 to 3 stars were considered negative, those from 4 to 5 as positive. As there was a class imbalance between positive and negative reviews, a balanced dataset was obtained by the process of data under-sampling in order to not introduce biases. Four different machine learning approached were employed i.e. SVM (Support Vector Machine), LSTM (Long Term Short Memory), CNN (Convolutional Neural Network), and BERT (Bi-directional Encoder Representations from Transformers) have been implemented for classification of amazon consumer reviews.

The results clearly indicate that BERT outperformed the other models, achieving the highest accuracy of 86.3% and demonstrating excellent precision, recall, and AUC ROC Score. This finding highlights the power of deep learning models, particularly BERT, in capturing complex patterns and contextual information within text data. BERT is one of the latest state of the art model as a result, it was expected to have performed better.

Interestingly, SVM with TF-IDF and LSTM with Wordd2Vec performed the best following BERT. Both obtaining an accuracy of 85.3% and 84.9% respectively. It is also to be considered that, many of these models can be resource intensive. Especially, BERT and LSTM demanded more resources when compared to SVM and CNN. Moreover, the comparison between TF-IDF and Word2Vec also helped determine which of these feature extraction methods was best suited. Word2Vec came out on top, indicating that Word2Vec embeddings are suited better for sentiment analysis for amazon customer review dataset.

In conclusion, the study aimed to compare the effectiveness and performance of different machine learning models for the purpose of sentiment analysis on the e-commerce consumer review data. Through the experiments and evaluation, the study has obtained valuable insights into the strengths and limitations of the chosen models. This research contributes to the ever-expanding domain of natural language processing for sentiment analysis using machine learning. The insights gained from the study can help businesses move towards the direction of data driven decision making. The stakeholders can make better decisions and employ one these techniques by understanding the comparative strengths of these models.

5.1. Scope for Further research

As with any research, it is essential to consider the limitations of this study. For instance, the performance of the models may change on different datasets, domain specificity, and the generalizability of the findings should be verified across multiple domains and data sources. Additionally, further optimization of hyperparameters and tuning of model architectures may enhance the performance of some models. This study can be further extended and the further research can be carried out in the following direction.

- Fine tuning BERT: While BERT demonstrated exceptional performance in sentiment analysis, further exploration can be done to fine-tune the BERT model specifically for e-commerce review data. Fine-tuning involves adapting the pre-trained BERT model to better suit the characteristics and language patterns of the target domain, which may lead to even higher accuracy and efficiency.
- Ensemble methods: Investigating the possibility of combining multiple models via ensemble techniques. Ensemble learning can help improve the overall performance and robustness for the task of sentiment analysis by leveraging the strength of individual models.
- Multi-lingual Sentiment Analysis: Extend the study to accommodate multi-lingual sentiment analysis if your e-commerce platform has reviews in multiple languages. Investigate pre-trained models and techniques that can handle sentiment analysis in different languages effectively.
- Real-time Sentiment Analysis: Explore the implementation of real-time sentiment analysis on live data streams. Investigate how the model can be deployed and updated in a production environment to provide up-to-date sentiment insights for businesses.
- Deployment and scalability: Consider the practical implications of deploying the sentiment analysis models in a real-world setting. Address challenges related to model deployment, scalability, and integration into existing business processes.
- Handling of textual data: Addition to above points, there is also a room to handle modern day reviews that include more than textual data. Not restricted to ecommerce data but also different sources that generate huge amount of data daily for example Twitter, and Facebook. Different reviews contain emoticons or in simple terms symbols like (☺, ☹) assist the user to express their emotions in a more impactful manner. Moreover, the models also need to take into consideration the stress that user tend to give on certain words that is directly proportional to the strength of their emotion. For example, words like ‘greatttt!’ and ‘niiice!!!’. Usually, they might not have a proper meaning but should be further processed to help identify the sentiment that is associated with the sentence.

- Evaluation methods: One of the matrices used to evaluate the models is the confusion matrix. Going ahead, this performance can also be evaluated using statistical tests like, ANOVA, Wilkinson test and t-test can be included to evaluate the performances of the employed systems.

References

Aline Bessa (2022) ‘Lexicon-Based Sentiment Analysis: A Tutorial’, 17 March [Online]. Available at https://www.knime.com/blog/lexicon-based-sentiment-analysis.

Andrienko, N., Andrienko, G., Miksch, S., Schumann, H. and Wrobel, S. (2021) ‘A theoretical model for pattern discovery in visual analytics’, *Visual Informatics*, vol. 5, no. 1, pp. 23–42.

Bengfort, B., Bilbro, R. and Ojeda, T. (between 1891 and 1894) *Russell, Hon. Benj. [Benjamin]: Enabling language-aware data products with machine learning*, Sebastopol CA, O'Reilly Media Inc.

Chen, N. (2022) ‘E-Commerce Brand Ranking Algorithm Based on User Evaluation and Sentiment Analysis’, *Frontiers in psychology*, vol. 13, p. 907818.

Cho, K., van Merrienboer, B., Bahdanau, D. and Bengio, Y. (2014) *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches* [Online]. Available at https://arxiv.org/pdf/1409.1259.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014) *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling* [Online]. Available at https://arxiv.org/pdf/1412.3555.

Cortes, C. and Vapnik, V. (1995) ‘Support-vector networks’, *Machine Learning*, vol. 20, no. 3, pp. 273–297.

Dave, K., Lawrence, S. and Pennock, D. M. (2003) ‘Mining the peanut gallery’, *Proceedings of the twelfth international conference on World Wide Web - WWW '03*. Budapest, Hungary, 20-05-2003 - 24-05-2003. New York, New York, USA, ACM Press, p. 519.

Denny Britz (2015) ‘Understanding Convolutional Neural Networks for NLP’, 2015 [Online]. Available at https://dennybritz.com/posts/wildml/understanding-convolutional-neural-networks-for-nlp/.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019) ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’, *Proceedings of the 2019 Conference of the North*. Minneapolis, Minnesota. Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 4171–4186.

Eke, C. I., Norman, A. A., Shuib, L. and Nweke, H. F. (2020) ‘Sarcasm identification in textual data: systematic review, research challenges and open directions’, *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4215–4258.

Elzeheiry, S., Gab-Allah, W. A., Mekky, N. and Elmogy, M. (2023) *Sentiment Analysis for E-commerce Product Reviews: Current Trends and Future Directions*.

Ethan Cramer-Flood (2020) *Ecommerce Decelerates amid Global Retail Contraction but Remains a Bright Spot* [Online], Insider Intelligence. Available at https://www.insiderintelligence.com/content/global-ecommerce-2020.

Fausett, L. (1994) *Fundamentals of neural networks: Architectures, algorithms, and applications / Laurene Fausett*, Englewood Cliffs, N.J., Prentice-Hall.

Ghose, A. and Ipeirotis, P. G. (2011) ‘Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 10, pp. 1498–1512.

Haque, T. U., Saber, N. N. and Shah, F. M. (2018) ‘Sentiment analysis on large scale Amazon product reviews’, *2018 IEEE International Conference on Innovative Research and Development (ICIRD).* Bangkok, 11-05-2018 - 12-05-2018, IEEE, pp. 1–6.

Hochreiter, S. and Schmidhuber, J. (1997) ‘Long short-term memory’, *Neural computation*, vol. 9, no. 8, pp. 1735–1780.

Hota, H. S., Sharma, D. K. and Verma, N. (2021) ‘Lexicon-based sentiment analysis using Twitter data’, in *Data Science for COVID-19,* Elsevier, pp. 275–295.

Iqbal, A., Amin, R., Iqbal, J., Alroobaea, R., Binmahfoudh, A. and Hussain, M. (2022) ‘Sentiment Analysis of Consumer Reviews Using Deep Learning’, *Sustainability*, vol. 14, no. 17, p. 10844.

Javaid Nabi (2019) *Recurrent Neural Networks (RNNs): Implementing an RNN from scratch in Python.* [Online], Towards Data Science. Available at https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85.

Jemimah Ojima Abah (2021) *Sentiment Analysis of Amazon Electronic Product Reviews using Deep Learning*, Master’s dissertation, Ireland, Dublin Business School [Online]. Available at https://esource.dbs.ie/handle/10788/4291.

Jonathon Read (2005) ‘Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification’, 43--48 [Online]. Available at https://aclanthology.org/P05-2008.

Kaur, J. and Sidhu, B. K. (2018) ‘Sentiment Analysis Based on Deep Learning Approaches’, *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS).* Madurai, India, 14-06-2018 - 15-06-2018, IEEE, pp. 1496–1500.

Kim, Y. (2014) ‘Convolutional Neural Networks for Sentence Classification’, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Doha, Qatar. Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 1746–1751.

Li, H., Ma, Y., Ma, Z. and Zhu, H. (2021) ‘Weibo Text Sentiment Analysis Based on BERT and Deep Learning’, *Applied Sciences*, vol. 11, no. 22, p. 10774.

Liu, B. ‘Sentiment Analysis and Subjectivity’.

Liu, Q., Wang, J., Zhang, D., Yang, Y. and Wang, N. (2018) ‘Text Features Extraction based on TF-IDF Associating Semantic’, *2018 IEEE 4th International Conference on Computer and Communications (ICCC).* Chengdu, China, 07-12-2018 - 10-12-2018, IEEE, pp. 2338–2343.

Madhusudhan Aithal, C. T. (2021) *On Positivity Bias in Negative Reviews.*

Michael Phi (2018) *Illustrated Guide to LSTM’s and GRU’s: A step by step explanation* [Online], Towards Data Science. Available at Illustrated Guide to LSTM’s and GRU’s: A step by step explanation.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) *Efficient Estimation of Word Representations in Vector Space* [Online]. Available at http://arxiv.org/pdf/1301.3781v3.

Mohammad, S. M. and Turney, P. D. (2013) ‘CROWDSOURCING A WORD-EMOTION ASSOCIATION LEXICON’, *Computational Intelligence*, vol. 29, no. 3, pp. 436–465.

Ni, J., Li, J. and McAuley, J. (2019) ‘Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects’, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Hong Kong, China. Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 188–197.

Pang, B. and Lee, L. (2008) ‘Opinion Mining and Sentiment Analysis’, *Foundations and Trends® in Information Retrieval*, vol. 2, 1–2, pp. 1–135.

Rambocas, M. and Pacheco, B. G. (2018) ‘Online sentiment analysis in marketing research: a review’, *Journal of Research in Interactive Marketing*, vol. 12, no. 2, pp. 146–163.

Rani Horev (2018) *BERT Explained: State of the art language model for NLP* [Online], Towards Data Science. Available at https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

Read, J. (2005) ‘Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification’, *Proceedings of the ACL Student Research Workshop.* Ann Arbor, Michigan, Association for Computational Linguistics, pp. 43–48.

Rita Kurban (2019) *CNN Sentiment Analysis CNN Sentiment: AnalysisUse Convolutional Neural Networks to Analyze Sentiments in the IMDb Dataset* [Online], Towards Data Science. Available at https://towardsdatascience.com/cnn-sentiment-analysis-9b1771e7cdd6.

SEPIDEH PAKNEJAD (2018) *Sentiment classification on Amazon reviews using machine learning approaches*, Stockholm, Sweden, KTH ROYAL INSTITUTE OF TECHNOLOGY [Online]. Available at https://web.archive.org/web/20200610153943/http://kth.diva-portal.org/smash/get/diva2:1241547/FULLTEXT01.pdf.

Srinivas, A. C. M. V., Satyanarayana, C., Divakar, C. and Sirisha, K. P. (2021) 'Sentiment Analysis using Neural Network and LSTM', *IOP Conference Series: Materials Science and Engineering*, vol. 1074, no. 1, p. 12007.

Stephanie Chevalier (2022) *Global retail e-commerce sales 2014-2026* [Online], Statista. Available at https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/#:~:text=In%202021%2C%20retail%20e%2Dcommerce,8.1%20trillion%20dollars%20by%202026.

Tan, J. Y., Chow, A. S. K. and Tan, C. W. (2022) 'A Comparative Study of Machine Learning Algorithms for Sentiment Analysis of Game Reviews', *The Journal of The Institution of Engineers, Malaysia*, vol. 82, no. 3.

Thompson, M., Duda, R. O. and Hart, P. E. (1974) 'Pattern Classification and Scene Analysis', *Leonardo*, vol. 7, no. 4, p. 370.

Tripathy, A. and Rath, S. K. (2017) 'Classification of Sentiment of Reviews using Supervised Machine Learning Techniques', *International Journal of Rough Sets and Data Analysis*, vol. 4, no. 1, pp. 56–74.

turbolab (2021) *Feature Extraction in Natural Language Processing* [Online]. Available at https://turbolab.in/feature-extraction-in-natural-language-processing-nlp/.

Turney, P. D. (2002) 'Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews'.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017) *Attention Is All You Need* [Online]. Available at http://arxiv.org/pdf/1706.03762v5.

Wankhade, M., Rao, A. C. S. and Kulkarni, C. (2022) 'A survey on sentiment analysis methods, applications, and challenges', *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780.

Werbos, P. J. (1990) 'Backpropagation through time: what it does and how to do it', *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560.

Wikipedia (2023) *Recurrent neural network* [Online]. Available at https://en.wikipedia.org/w/index.php?title=Recurrent_neural_network&oldid=1162017246 (Accessed 17 July 2023).

Statutory Declaration

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper.

I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content.

I am aware that the violation of this regulation will lead to failure of the thesis.

Himanshu Dharm
Student’s name

Himanshu Dharm
Student’s signature

581332
Matriculation number

06.08.2023
Berlin, date