Sentiment Analysis for Product Review using Machine Learning and Deep Learning Methods

Submitted by

Md. Minhajur Rahman

ID: CSE 01806748

A thesis submitted in conformity with the requirement for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering

PORT CITY INTERNATIONAL UNIVERSITY

7-14, Nikunja Housing Society, South Khulshi, Chattogram,Bangladesh

January 2023

Sentiment Analysis for Product Review using Machine Learning and Deep Learning Methods

Submitted by

Md. Minhajur Rahman

ID: CSE 01806748

Under the Supervision of

Mrs. Manoara Begum

Senior Lecturer & Chairman (in-charge)

Department of CSE

Port City International University

A thesis submitted in conformity with the requirement for the degree of
Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering

PORT CITY INTERNATIONAL UNIVERSITY

7-14, Nikunja Housing Society, South Khulshi, Chattogram,Bangladesh

January 2023

DECLARATION

I humbly announce that this thesis entitled, “Sentiment Analysis for Product Review using Machine Learning and Deep Learning Methods” and this work is completely my claim for my undergrad degree. All area in this proposal has been thesis recognized.

(Signature of the candidate)

Name: Md. Minhajur Rahman

ID: CSE 01806748

B.Sc. in CSE

Port City International University

RECOMMENDATION

It is therefore confirmed that the thesis, **" Sentiment Analysis for Product Review using Machine Learning and Deep Learning Methods "** has been submitted by the criteria for the degree of B.Sc. in the Department of Computer Science and Engineering. This is a valid interpretation of the research effort carried out by **Md. Minhajur Rahman** student **ID: CSE 01806748**, under my instruction and oversight.

Supervisor
Mrs. Manoara Begum
Senior Lecturer & Chairman (in-charge)
Department of Computer Science and Engineering
Port City International University
Email: manoara.cse@portcity.edu.bd
Cell: 01881-075015

ACKNOWLEDGEMENT

I want to start by expressing our deep appreciation to Allah, the Almighty, for his wondrous gift, which made it possible for me to finish my final year thesis. I had the good fortune to study under the esteemed supervision of **Mrs. Manoara Begum**. I am extremely appreciative of her advice and support. Because of her great experience, I have the chance to learn more and make substantial progress. All of our teachers in the Department of Computer Science and Engineering at Port City International University gave me the support I needed to get my degree in Computer Science and Engineering. Last but not least, I must sincerely thank my parents for their constant support and tolerance.

ABSTRACT

Today's world is becoming increasingly digitalized. In today's digitalized world, e-commerce is growing popularity since it brings things closer to customers without requiring them to leave their homes. A customer must read hundreds of reviews before making a purchase. The number of internet evaluations for a single product can easily approach millions, making tracking and understanding client feedback difficult. However, in the age of machine learning and deep learning, it would be much easier to gain thousands of inputs and knowledge from them if a model was used to polarize and understand them. As a result, sentiment analysis is a new research field that combines natural language processing and text analysis to extract subjective information from sources and classify the polarity of expressed sentiments. Using Sentiment Analysis, 55000 reviews were classified as positive, negative or neutral in the proposed work. Among the numerous classification models, Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Convolutional neural network (CNN) and Long short-term memory (LSTM) have been employed for classification of reviews. RF outperformed all other algorithms with the highest accuracy of 93.55%.

Keywords: Sentiment Analysis, Natural Language Processing, Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Convolutional neural network (CNN), Long short-term memory (LSTM).

TABLE OF CONTENT

CHAPTER **PAGE**

CHAPTERS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

The list Describes several abbreviations and will be used later within the documentation.

MNB Multinomial Naive Bayes

LR Logistic Regression

DT Decision Tree

RF Random Forest

SVM Support Vector Machine

KNN K-Nearest Neighbors

CNN Convolutional Neural Network

LSTM Long Short-Term Memory

Chapter 1
Introduction

1.1 Introduction

Because online marketplaces have grown in popularity over the last few decades, online sellers and merchants now ask their customers to provide feedback on the products they have purchased. Every day, millions of evaluations are written about various products, services, and locations online. This has made the Internet the most significant of acquiring concepts and beliefs about a product or a service.

However, as the number of reviews available for a product increases, it becomes more difficult for a potential consumer to make an informed decision about whether or not to purchase the product. Different opinions on the same product on the one hand, and ambiguous reviews on the other, make it more difficult for customers to make the right decision. Here, it appears that all e-commerce enterprises must analyze these contents.

Sentiment analysis and classification is a computational study that attempts to address this problem by extracting subjective information such as opinions and sentiments from natural language texts. Natural language processing, text analysis, computational linguistics, and biometrics are just a few of the methods that have been explored to approach this issue. In recent years, Machine learning And Deep learning methods have been well-liked in the semantic and review analysis for their simplicity and accuracy.

People frequently shop online at Amazon, one of the biggest e-commerce sites, where they may read thousands of reviews evaluations left by other consumers about the things they want to buy. These reviews provide valuable opinions about a product such as its property, quality, and recommendations, which aids purchasers in understanding nearly every detail of a product. This is not only beneficial to consumers, but it also assists sellers who manufacture their own products in better understanding consumers and their needs.

This thesis investigates the sentiment classification problem for online reviews, employing supervised approaches to determine the overall semantic of customer reviews by classifying them as positive, negative, or neutral. Data on reviews of clothing, shoes, and jewelry were taken from the Amazon.com customer reviews dataset for this study. The reviews were classified using deep learning and machine learning classification models, including Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), and they were cross validated to determine the best classifier for this purpose.

1.2 Problem Statement

Traditional sentiment analysis considers the intensity of feelings, emotions, urgency, and motives, whereas models take into account positive, negative, and neutral emotions. The system can track specific words and concepts to provide detailed information for those interested in such analysis by working with questions, opinions, and feedback. In some instances, before making a purchase, customers will use sentiment analysis to research a product or service.Customers may use sentiment analysis to research a product or service before making a purchase in some cases. Marketers will use surveys to learn about the public's ideas and tastes in order to determine how well their products are performing and how much or little consumers want them. Corporations and marketing agencies used the results of this methodology to gather valuable input on newly released products. Businesses and firms can also use this methodology to gather valuable input on newly released products. Sorting through the emotions associated with ratings and social media discussions and processing them more quickly and accurately broadens the options for making more specific and precise decisions. Due to insufficient data, labeled information, and the inherent complexity of complex sentences, current methods are completely ineffective, inaccurate, and unable to perform sentiment analysis without an extensive collection of Sentiment Analysis.

1.3 Motivation

Sentiment analysis is currently at the leading position of scientific research. Its applications are beneficial in many cases. It is helpful for achieving results without wasting time or brainpower. This thesis investigates the sentiment classification problem for online reviews, employing supervised approaches to determine the overall semantic of customer reviews by classifying them as positive, negative, or neutral. this model could be assist possible clients with settling on an informed decision on their purchase and organizations to improve their items or services.

1.4 Objectives

My research aims to do this by conducting sentiment analysis on customer reviews and classifying them as positive, negative, or neutral. I believe that my suggested approach would reduce consumer frustration while shopping online since they will be able to examine product reviews based on the ratio of previous customers' positive, negative, and neutral comments. It can also be beneficial to the vendor because it allows him to spot product flaws and give better customer service.

1.5 Thesis Outline

I covered the basics of how the detection system works in

Chapter 1 – I now present an outline of the content of the next chapters in this thesis.

Chapter 2 –Literature Review. This chapter discusses previous research on sentiment analysis and natural language processing on review text.

Chapter 3 – Methodology. This chapter describes how the whole system works for the classification of Product Review. This chapter presents a short overview of our proposed methodology and also discusses Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Convolutional Neural Network(CNN), Long Short-Term Memory (LSTM) and the feature extraction process.

Chapter 4 – Hardware and Tools. This chapter describes the Tools and Hardware we have used to implement the system. A short description was given of the tools we have used in the research.

Chapter 5 – Experimental Result & Discussion. This chapter presents the results of the performance of our proposed model for Product Review detection and provides visual representation.

Chapter 6 – Conclusion and Future Work. This chapter concludes the evaluated results and observations. It also points out some limitations and future scope for this research.

Chapter 2
Literature Review

2.1 Introduction

Previous research on sentiment analysis and natural language processing on review text is discussed in this chapter.

2.2 Sentiment Analysis

Sentiment analysis is textual contextual mining that identifies and extracts subjective information from the source material, assisting businesses in understanding the social sentiment of their brand, product, or service while monitoring online conversations. However, most social media analysis is limited to basic sentiment and count-based metrics. This is analogous to only scratching the surface and missing out on high-value insights that are waiting to be discovered [12].

2.3 Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence that deals with the interaction of computers and humans through the use of natural language. The ultimate goal of NLP is to read, decipher, comprehend, and make sense of human languages in a valuable way. To derive meaning from human languages, most NLP techniques rely on machine learning [13].

2.4 Related work

[1] In 2019 Najma Sultana, Pintu Kumar, Monika Rani Patra, Sourabh Chandra and S.K. Safikul Alam worked on a research paper named "Sentiment Analysis For Product Review" where they work on a dataset including 50000 movie reviews & got accuracy respectively NB-89.85%, LR- 89.57%, LSVC- 89.36%, DT 87.78% .

[2] In 2019, Levent Guner, Emilie Coyne and Jim Smit worked on a research paper named "Sentiment Analysis For Amazon.Com Reviews" Using 60000 random reviews between 4 million reviews & got accuracy respectively LSVM-86%, MNB-85%, LSTM:90%.

[3] In 2018, Tanjim Ul Haque ,Nudrat Nawal ,Saber and Faisal Muhammad Shah worked on a research paper named ‘Sentiment Analysis on Large Scale Amazon Product Reviews’

Using 48500 product reviews & also using SVM, MNB, SGD, RF,LR,DT and got overall accuracy of 90%.

[4] In 2019, Rajkumar S. Jagdale, Vishal S. Shirsat and Sachin N. Deshmukh worked on a research paper named ‘Sentiment Analysis on Product Reviews Using Machine Learning Techniques’ using NB, SVM, and got overall accuracy 98.17%.

[5] In 2017, Zeenia Singla, Sukhchandan Randhawa and Sushma Jain on a research paper named ‘Sentiment Analysis Of Customer Product Reviews Using Machine Learning’ using SVM, NB,DT and got overall accuracy of 81.77%.

[6]In 2017, Palak Baid, Apoorva Gupta and Neelam Chaplot on a research paper named ‘Sentiment Analysis of Movie Reviews using Machine Learning Techniques’ using NB,KNN,RF and got overall accuracy of 81.4%.

[7]In 2020, LI YANG ,YING LI , JIN WANG AND R. SIMON SHERRATT ‘Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning’ using NB,SVM,CNN,BiGRU,SLCABG and got overall accuracy of 93.5%.

[8]In 2017, Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat and Priyanka Badhani ’ Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python’ using DAN2,SVM, Bayesian Logistic Regression, Naïve Bayes, Random Forest Classifier, Neural Network, Maximum Entropy , Ensemble classifier,and got overall accuracy of 90.0%.

[9]In 2017, Zeenia Singla, Sukhchandan Randhawa, and Sushma Jain,’ STATISTICAL AND SENTIMENT ANALYSIS OF CONSUMER PRODUCT REVIEWS’ using SVM got accuracy of 84.87%.

[10]In 2021, Mst. Tuhin Akter , Manoara Begum and Rashed Mustafa ‘ Bengali Sentiment Analysis of E-commerce Product Reviews using K-Nearest Neighbors’ using RF,LR,SVM,KNN,XGBoost and got overall accuracy of 96.25%.

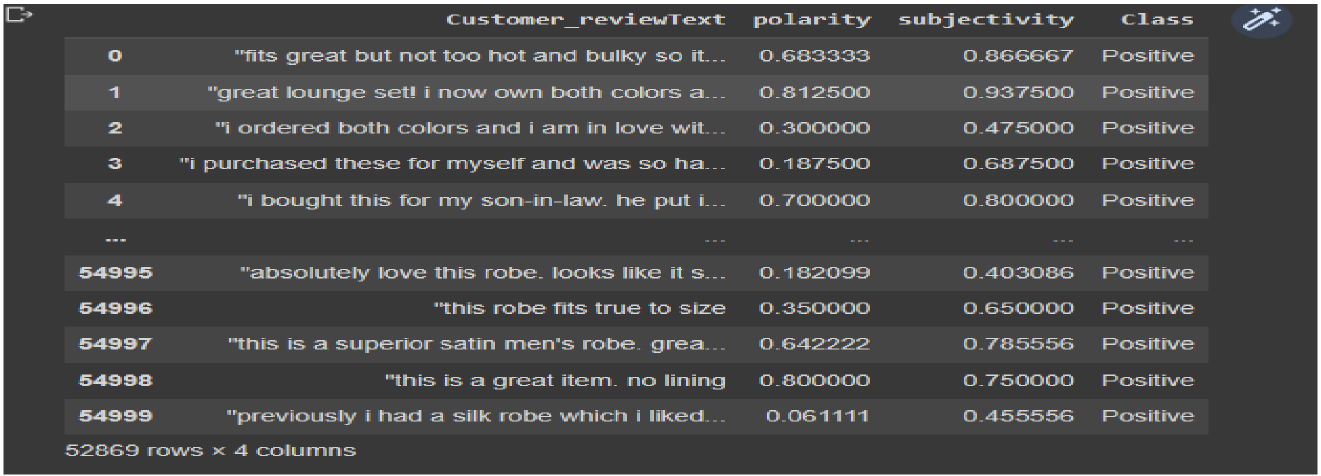
Chapter 3

Methodology

This chapter describes how the whole system works which has been used for the Detection of Product Review also gives detailed information about the system architecture.

3.1 Dataset Collection

I collect the Amazon Review dataset from Julian McAuley, UCSD website. "Clothing, Shoes, and Jewelry" is the name of this dataset. This dataset contains 278 677 reviews. From this dataset, I took a total of 55000 reviews, which I used on my thesis. This dataset has been labeled with three classes, which are positive, negative, and neutral [11].



```
                          Customer_reviewText   polarity  subjectivity     Class
0       "fits great but not too hot and bulky so it...   0.683333      0.866667  Positive
1      "great lounge set! i now own both colors a...   0.812500      0.937500  Positive
2       "i ordered both colors and i am in love wit...   0.300000      0.475000  Positive
3     "i purchased these for myself and was so ha...   0.187500      0.687500  Positive
4       "i bought this for my son-in-law. he put i...   0.700000      0.800000  Positive
...                                             ...        ...           ...       ...
54995    "absolutely love this robe. looks like it s...   0.182099      0.403086  Positive
54996                   "this robe fits true to size   0.350000      0.650000  Positive
54997    "this is a superior satin men's robe. grea...   0.642222      0.785556  Positive
54998                "this is a great item. no lining   0.800000      0.750000  Positive
54999   "previously i had a silk robe which i liked...    0.061111      0.455556  Positive
52869 rows × 4 columns
```

Figure 3.1: Sample of the dataset

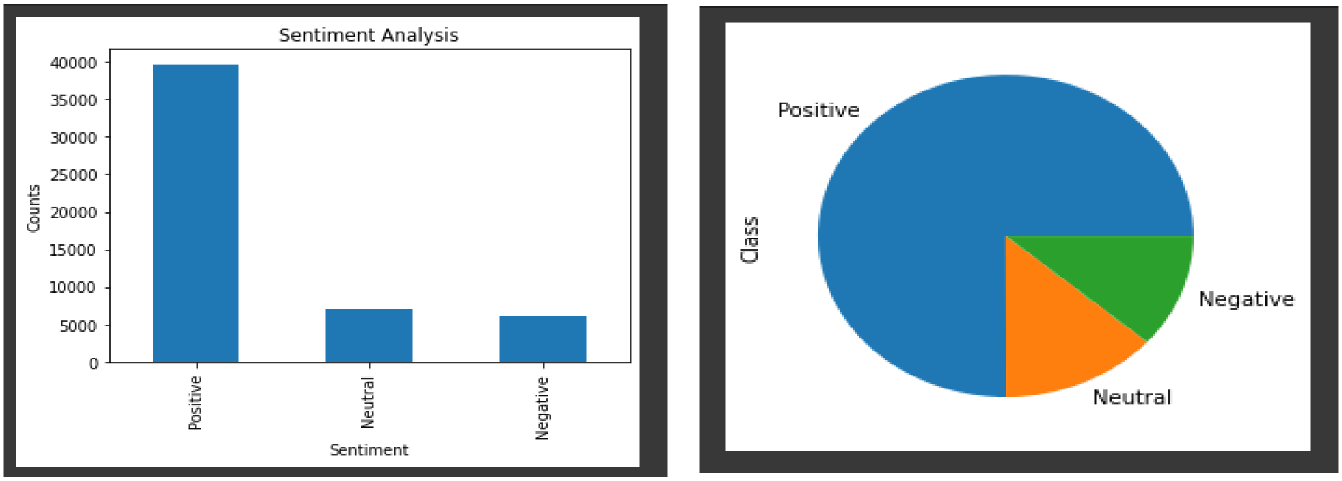


Figure 3.2: Visualization of Dataset

3.2 Working Procedure

The main goal of my research is to detect Product Reviews from English text comments and classify what type of Review comment it is. So, I can say that it is a multiclass classification problem. This work is accomplished through the use of various machine learning and deep learning models. I preprocessed the comments to remove noisy data before reviewing the classifiers. Then I extracted features from preprocessed data. After this, I used a technique called Oversampling to deal with the issues of imbalanced data which I have in our dataset. Machine learning and deep learning kept 80% of training and 20% for testing purposes. Finally fed the training data into classifiers and deep learning models. Then used different evaluation methods to predict the outcome.

3.2.1 System Workflow

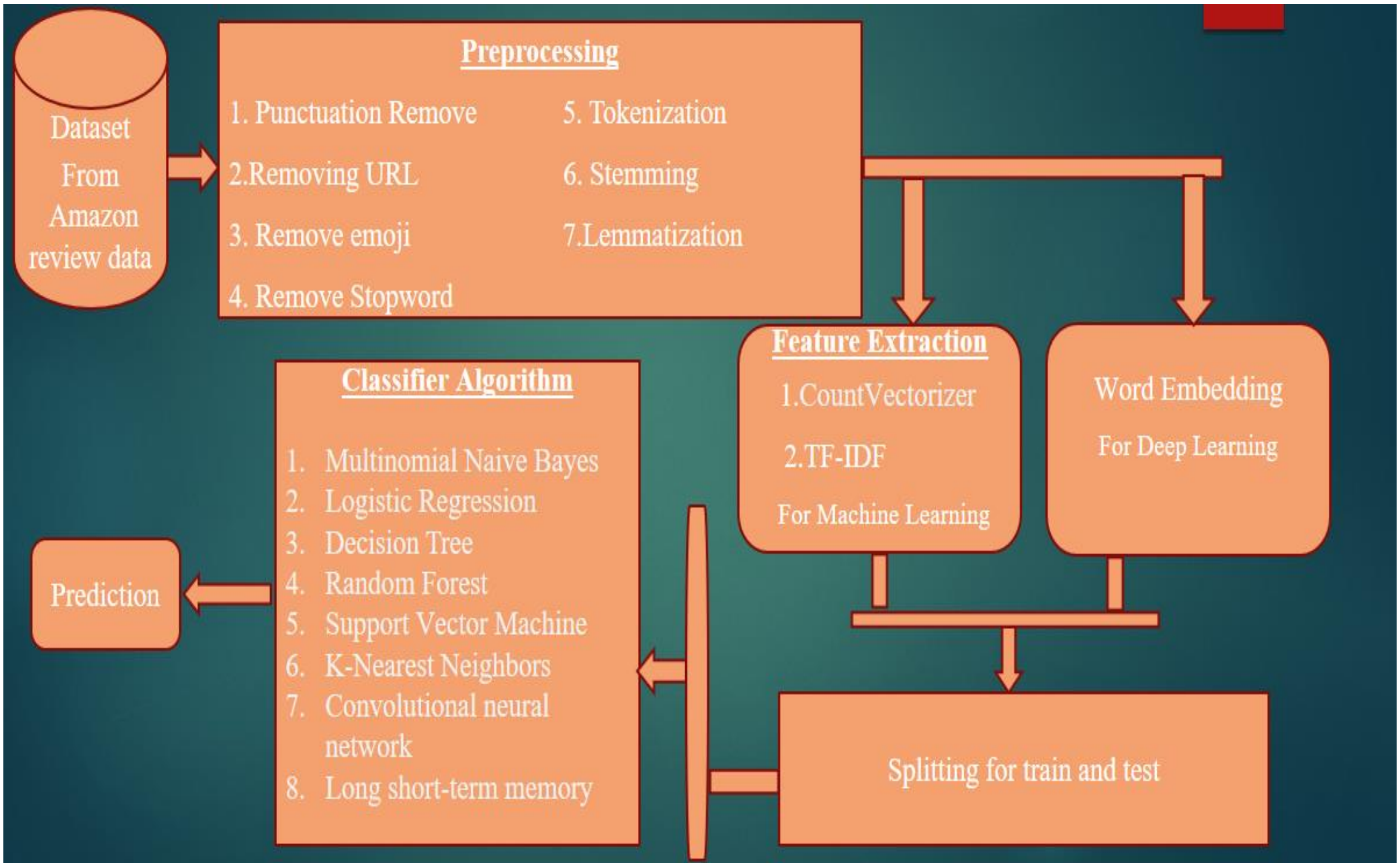
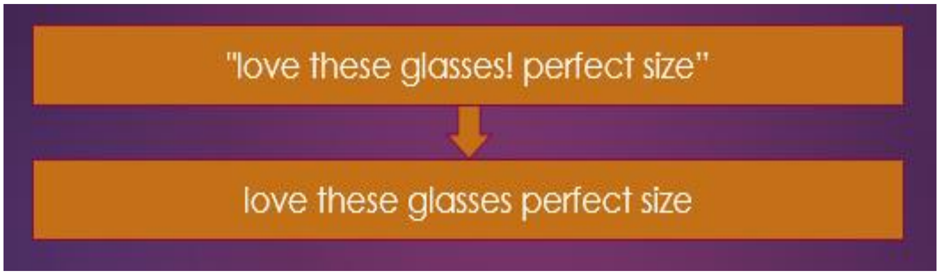


Figure 3.3: A flowchart of the proposed system

3.3 Data Preprocessing

In addition to the original data, I have some unnecessary data in this dataset that I do not require. In the beginning, I have to remove that unnecessary thing. The processes will be discussed below.

□ **Punctuation Remove** : The removal of punctuation from textual data is the most common text processing technique. The process of removing punctuation will aid in treating each text equally. For example, after the punctuation is removed, the words data and data! are treated equally. We must also exercise extreme caution when selecting the list of punctuations to exclude from the data based on the use cases. Python's string. punctuation contains these symbols! "#$%&\'()*+,-./:;?@[\\]^ {|}~`

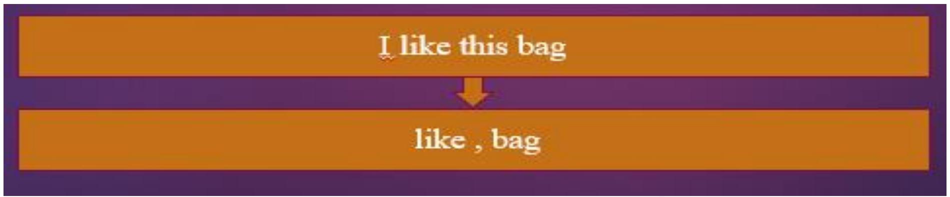

"love these glasses! perfect size"

love these glasses perfect size

Figure 3.4: Punctuation Remove

□ **Removing Stopwords :** Stopwords are words that are commonly used to describe less meaningful concepts, such as article, pronoun, preposition, and so on. Stop using NLP and text mining.

In general, words are removed. Different stop terms are used in various formats depending on the country, language, and other factors [6]. There may be many meaningless words in a document. If we do not remove these words, the classifier will have a difficult time determining the correct result. There will undoubtedly be a large number of articles and pronouns in a document. If we do not remove this, the classifier will prioritize these words because they appear frequently in the document and will not produce an accurate result. As a result, these stop words must be removed.


I like this bag

like , bag

Figure 3.5: Removing Stopwords

❑ **Tokenization :** Tokenization is the process of dividing long strings of input text into smaller chunks. It converts a string or document to tokens (smaller chunks). Tokenization is most commonly used to divide values such as a document, sentence, or paragraph into smaller units such as words or subwords. These smaller units are referred to as tokens. It is a step in the process of preparing a text for natural language processing. Tokenization has several advantages, including making it easier to map parts of speech, matching common words, and removing unwanted tokens.

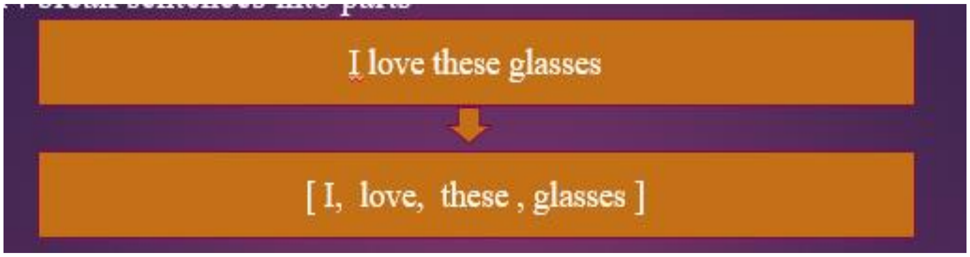


Figure 3.6: Tokenization

❑ **Stemming :** Stemming is a strategy for reducing emphasis in words to their root shapes, such as mapping a group of words to the same stem. Indeed, the stem is not a significant word in the language. Stemming is a method of lessening intonation towards their root shapes; this occurs so that portraying a group of related words beneath the same stem, even if the root has no suitable meaning. The goal of stemming is to reduce our lexicon and dimensionality for Natural language processing tasks, as well as to improve speed and effectiveness in data recovery and data handling tasks. For stemming, both English and non-English stemmers are available. Porter and Lancaster are the most commonly used English language stemmers. Porter stemmer is the oldest of them all.

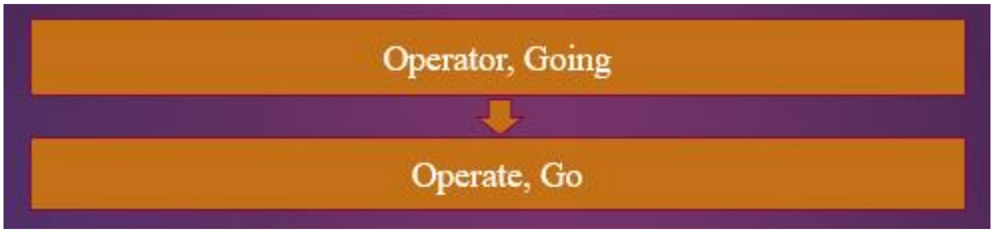
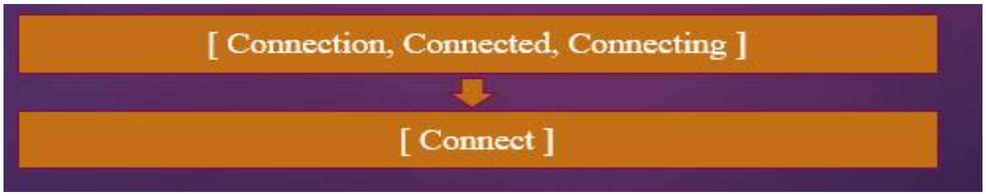


Figure 3.7: Stemming

❑ **Lemmatization :** The lemmatization technique is similar to stemming. The result of lemmatization is known as a 'lemma,' which is a root word rather than a root stem, which is the result of stemming. We will get a valid word that means the same thing after lemmatization.



[Connection, Connected, Connecting]

[Connect]

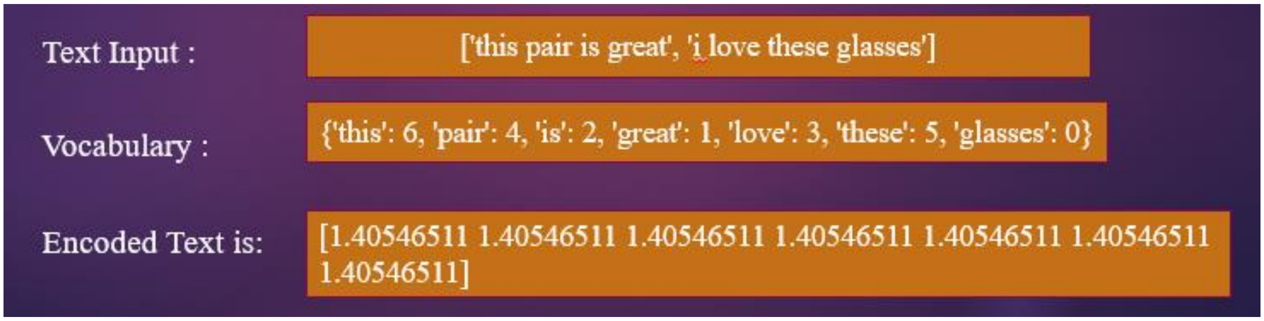
Figure 3.8: Lemmatization

3.4 Feature Extraction

The feature allows text data to be represented numerically. We all know that machines can only understand 0 and 1. Machine learning, similarly, cannot understand without numerical data. However, my Dataset only contains string data. So, I need to convert this string data to a numerical format, which will be covered in this section.

3.4.1 Term Frequency and Inverse Document Frequency (TF-IDF)

TF-IDF is an abbreviation for term frequency (TF) and inverse document frequency (IDF) (IDF). It primarily defines a word's significance in a document. The frequency or rapidity of a term in a document is determined by TF, while the unusualness of a term is determined by IDF. The TFIDF value increases in proportion to the number of times a word appears in the database, but is offset by the number of current records containing the word. TF-IDF weights each term by isolating the term repeat by the number of documents in the corpus containing the word , rather than referring to a phrase in a record by its rough repeat (number of occurrences) or relative repeat. Despite data preprocessing, repeated terms that are irrelevant to sentiment analysis may exist. For example, the terms "buy" and "sell" are overused in a product buy and sell review package, but they are meaningless in sentiment analysis. Each word is assigned a distinct TF-IDF score. A term's weight is determined by comparing the scores of its TF and IDF products. Furthermore, this score is important in determining how rare or common a word is. The higher the TF*IDF score, the rarer the word; conversely, the lower the TF*IDF score, the more frequent the word in the document.



```
Text Input :        ['this pair is great', 'i love these glasses']

Vocabulary :        {'this': 6, 'pair': 4, 'is': 2, 'great': 1, 'love': 3, 'these': 5, 'glasses': 0}

Encoded Text is:    [1.40546511 1.40546511 1.40546511 1.40546511 1.40546511 1.40546511
                    1.40546511]
```

Figure 3.9: TF-IDF

3.4.2 Count Vectorizer

The term CountVectorizer refers to the process of breaking down a sentence or any text into words by performing preprocessing tasks such as converting all words to lowercase, thereby removing special characters. Because NLP models cannot understand textual data and only accept numbers, textual data must be vectorized.

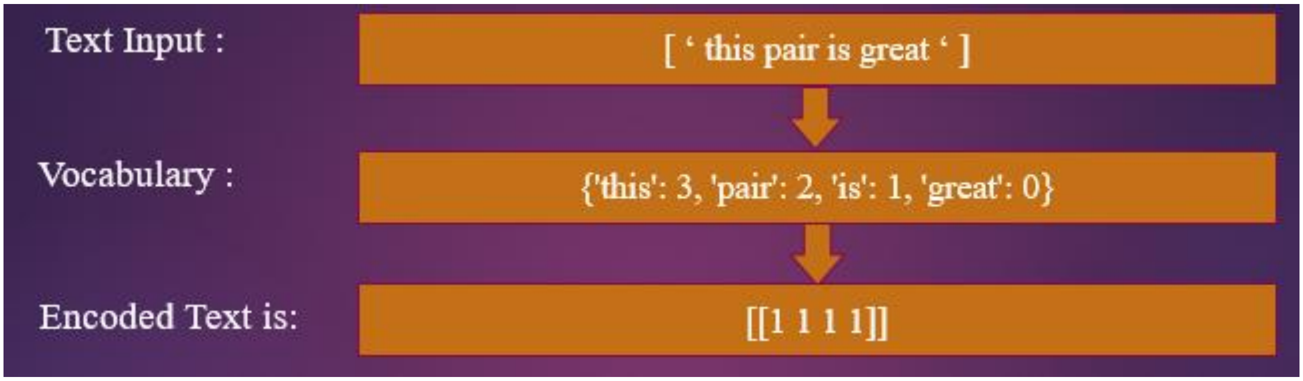

Text Input :
[‘ this pair is great ‘]
Vocabulary :
{'this': 3, 'pair': 2, 'is': 1, 'great': 0}
Encoded Text is:
[[1 1 1 1]]

Figure 3.10: Count Vectorizer

3.4.3 Word Embedding

Word embedding produces a numerical representation of textual data. It offers equivalent representations for words with similar semantic features to help in word differentiation [14].

3.5 Classifiers

This section gives a brief overview of the algorithms used in my thesis study. Here, I evaluated two deep learning models and six machine learning models in our system. MNB, LR, DT, RF,SVM and KNN are the six algorithms under machine learning, and CNN and LSTM are used under Deep learning models.

3.5.1 Multinomial Naive Bayes (MNB)

When categorizing texts based on a statistical examination of their contents, the multinomial naive Bayes algorithm is frequently utilized. It offers an alternative to "heavy" AI-based semantic analysis and significantly streamlines the classification of textual material. The classification aims to assign text fragments (i.e. documents) to classes by determining the likelihood that a document belongs to the same class as other documents with the same subject.

Each document contains multiple words (i.e. terms) that help the reader understand the document's contents. A class is a tag that refers to the same subject in one or more documents. Labeling documents with one of the existing classes is accomplished through statistical analysis, which tests the hypothesis that the terms in a document have already occurred in other documents from the same class.

3.5.2 Support Vector Classifier (SVC)

A Linear SVC (Support Vector Classifier) is to fit the data you provide, returning a "best fit" hyperplane that divides or categorizes your data. After you've obtained the hyperplane, you can feed some features into your classifier to see what the "predicted" class is. The Linear SVC model has more parameters than the SVC model, such as penalty normalization (L1 or L2) and loss function.

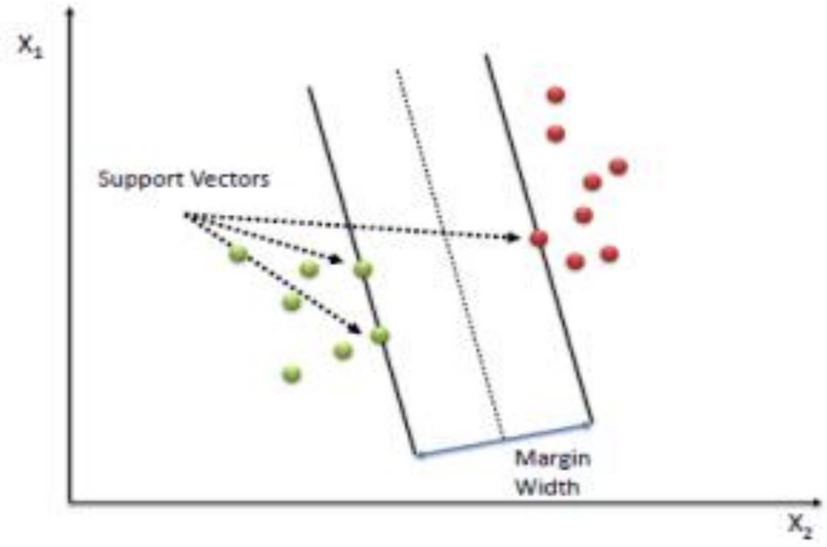


Figure 3.11: SVC Classifier

Kernel functions are a method of converting input data into the format required for processing. The Kernel Function modifies the training set of data in this way so that a non-linear decision surface can be converted into a linear equation in higher dimension spaces. The implementation makes use of the SVC library. The fit time may become prohibitive after tens of thousands of data points because it scales at least quadratically with the number of samples. When dealing with large datasets, consider using Linear SVC or SGD Classifier. A one-to-one system is used to manage multiclass support.

3.5.3 Logistic Regression (LR)

The most basic version of logistic regression, though there are many more complex variations, employs a logistic function to describe a binary dependent variable. Regression analysis employs the logistic regression method (also known as logit regression) to estimate the parameters of a logistic model (a form of binary regression). At the turn of the twentieth century, the biological sciences began to employ logistic regression. Later, it was used for a variety of social science applications. Logistic regression is a technique for estimating the probability of a discrete outcome from an input variable. Types of Logistic Regression-

- Binary Logistic Regression-The categorical response has only two 2 possible outcomes. Example: Spam or Not.
- Multinomial Logistic Regression-Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan).
- Ordinal Logistic Regression-Three or more categories with ordering. Example: Movie rating from 1 to 5.

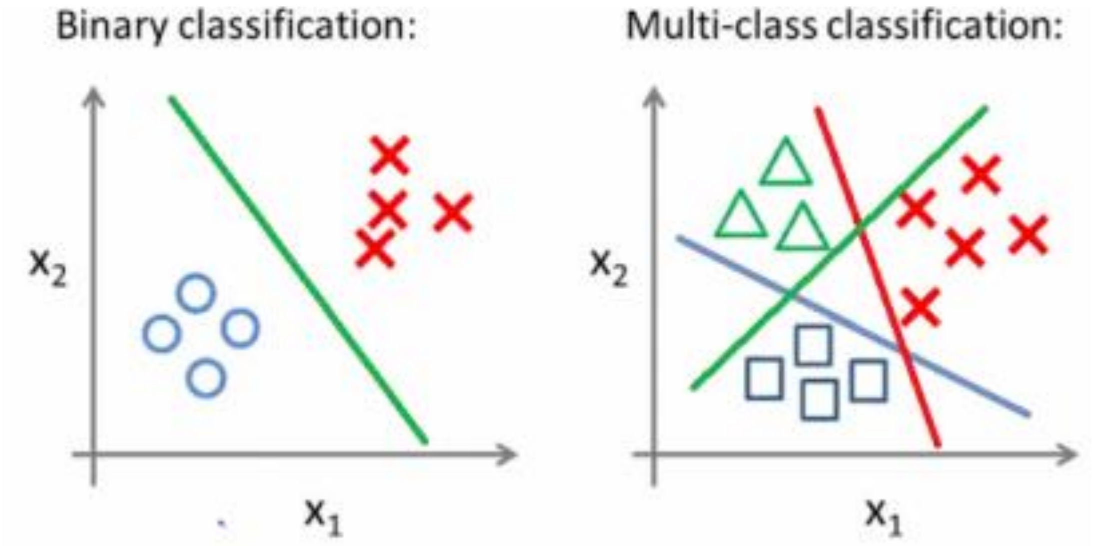


Figure 3.12: Logistic Regression

To model the probability of a given class or occurrence, a supervised machine learning method known as logistic regression can be used. When the outcome is binary or dichotomous and the data can be separated linearly, this strategy is used. As a result, logistic regression is frequently used to solve problems involving binary classification.

3.5.4 Decision Tree (DT)

The Decision Tree algorithm is a member of the supervised learning algorithm family. The goal of using a decision tree is to build a training model that can be used to predict the class or value of the target variable by learning simple choice rules generated from prior data (training data). Decision trees are a type of supervised machine learning that divides data indefinitely based on a parameter. Using the binary tree from earlier, one can comprehend a decision tree. Most decision tree algorithms work top-down, selecting a variable that best divides the set of objects at each stage. Decision trees use a variety of techniques to determine whether to divide a node into two or more sub-nodes. Sub-node formation increases the homogeneity of newly formed sub-nodes. The decision tree divides the nodes based on all of the available factors, then selects the split that produces the most homogeneous sub-nodes. It can help with both decision making and decision representation. A condition is represented by each internal node in a decision tree, and a choice is represented by each leaf node in a decision tree.

3.5.5 Random Forest (RF)

During the training phase of the random forests or random decision forests ensemble learning approach, which is used for classification, regression, and other tasks, a large number of decision trees are built. For classification problems, the random forest output is the class that the majority of the trees chose. For regression tasks, the mean or average prediction of each tree is returned. In RF, the classifier data variables are drawn at random from a large number of trees. The RF is built using the four streamlined stages that follow. There are N instances (cases) in the training data, and M attributes in the classifier. The number of examples (or "cases") in the training data is N, while the number of attributes (or "attributes") in the classifier is M.

3.5.6 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm and technique can be used for both regression and classification tasks. K-Nearest Neighbors examines the labels of a predetermined number of data points surrounding a target data point to predict which class the data point belongs to. K-Nearest Neighbors (KNN) is a conceptually simple yet extremely powerful algorithm, and it is one of the most widely used machine learning algorithms. Let's take a closer look at the KNN algorithm and see how it works. Understanding how KNN works will allow you to appreciate the best and worst use cases for KNN.

3.6 Deep Learning Algorithms

3.6.1 Convolutional neural network (CNN)

Deep learning has emerged as a very useful approach in recent years due to its ability to handle massive amounts of data. Hidden layers have surpassed traditional methods in popularity, particularly in pattern recognition. The convolutional neural network is a popular deep neural network. A Convolutional Neural Network (CNN) is a Deep Learning system that can take in an input image, assign importance to different elements and objects in the image (learnable weights and biases), and distinguish between them.

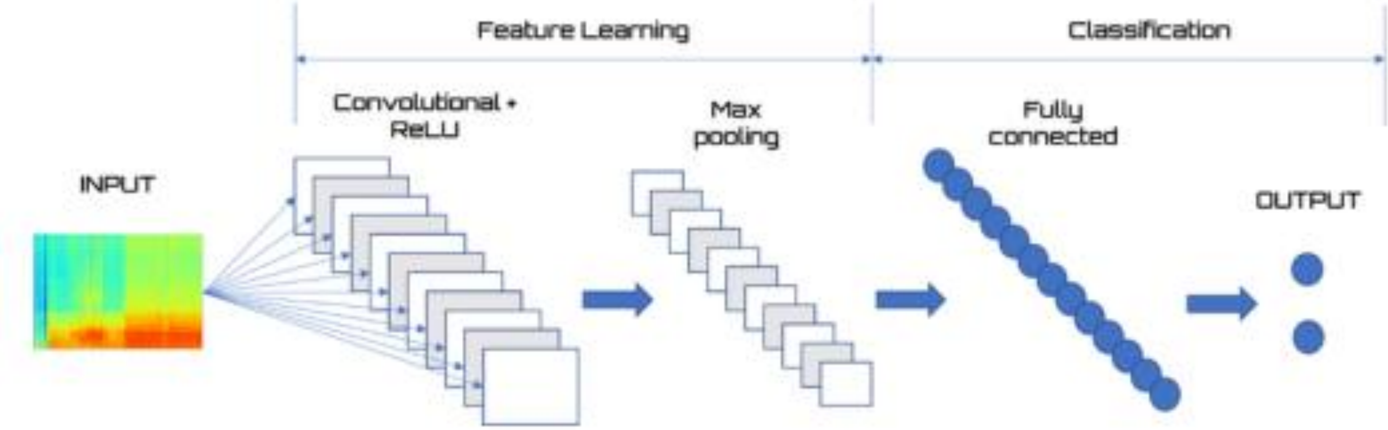


Figure 3.13: Structure of CNN

❑ **Convolutional layer**

The first and most important layer is the convolution layer. The convolutional layer's K filters look for phrases in the input that are similar to the filter. A one-dimensional convolution is an operation that involves a weight vector, m R m, and an input vector, s R s, where m is the convolution's filter. Convolution is the degree of overlap when one function is transferred to another. In this instance, the filter scans the entered sentence in search of word patterns that match the filter and the phrase. Furthermore, filters are taught during training, allowing us to train them on more data and build up their strength to match the demands of our network. We fix certain parameters (hyper), which are independent of training, such as the number of filters and the stride (the offset of how far to shift the filter across the text).

❑ **Max pooling layer**

Pooling layers often come after convolutional layers. The input that is supplied to the pooling layers is divided into subsamples. The most common pooling technique involves doing a max operation on the output of each filter. Pooling in NLP often produces just one value for each filter and is applied to the total output. For a 22 window, the following example shows maximum pooling. Pooling reduces storage and over-fitting.

❑ **Fully connected layer**

A totally connected input layer "flattens" the output of the preceding layers into a single vector that may be utilized as an input for the following layer. To forecast the correct label, the first fully connected layer applies weights to the feature analysis inputs. A fully connected output layer provides the final probability for each label.

3.6.2 Long Short-Term Memory (LSTM)

LSTMis a fundamental RNN extension. It reduces the problem of disappearing gradients and can track
dependencies across large gaps.

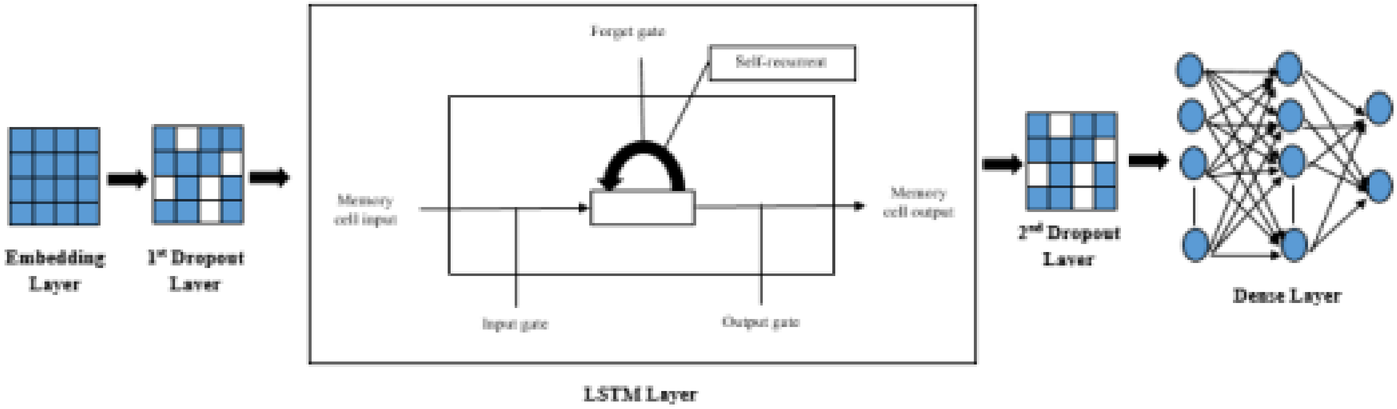


Figure 3.14: Layer combination of the LSTM architecture

An LSTM's recurrent hidden layer contains distinct units known as memory blocks. Memory blocks include memory cells with self-connections that store the network's temporal state, in addition to specific multiplicative units called gates that regulate the flow of information. In the original architecture, each memory block had one of three different gate types, namely:

- Input gate: The input gate regulates the flow of input activations into the memory cell.
- Output gate: this gate regulates how cell activations exit the network and enter other nodes.
- Forget gate: scales the internal state of the cell before adding it as input through the self-recurrent connection of the cell, adaptively forgetting or resetting the memory of the cell.

Chapter 4

Hardware and Tools

This chapter describes the tools used to implement the system to detect Product Review. It gives detailed information about the tools used to implement the system.

4.1 Tools

The following tools will be used in the implementation of the designed system:

- Colaboratory Notebook
- NumPy
- Pandas
- OS
- Scikit-Learn
- Matplotlib
- Anaconda
- Python
- Jupyter Notebook
- Google Drive

4.1.1 Colaboratory Notebook

The Collaboratory, or "Colab," is a Google Research creation. Because of the web-based Python editor Colab, anyone can write and run Python programs. It has applications in education, data analysis, and machine learning. Colab is a hosted Jupyter notebook service that does not require installation and provides free access to computing resources such as GPUs. We can use and share Jupyter notebooks with others through Colab without downloading, installing, or running anything. Colab's resources are not always available or limitless, and usage guidelines are subject to change at any time. This is required for Colab to provide free materials. Go to Resource Limits for more information.

4.1.2 NumPy

NumPy, a Python library that works with arrays, has a variety of applications. It also includes tools for working with a multidimensional array object with high performance. This is the most important Python package for scientific computing. It has several characteristics, including the following: An N-dimensional array's strong object (Broadcasting) C/C++ and Fortran code integration tools with advanced features Linear algebra, the Fourier transform, and random number generation are all capabilities. In addition to its obvious scientific applications, NumPy can be used as a multidimensional container of generic data. NumPy can connect to a variety of databases quickly and cleanly because it allows the creation of any data type.

4.1.3 Pandas

Pandas, an open-source Python toolkit, uses solid data structures to provide high-performance data manipulation and analysis. The term "Pandas" comes from the term "Panels Data," which is used in econometrics to refer to multidimensional data. Python was mostly used for data preparation and munging before Pandas. It was only marginally helpful for data analysis. This problem was solved by pandas. We can use Pandas to import, prepare, alter, model, and analyze data from any source. Finance, economics, statistics, analytics, and other academic and professional fields use Python with Pandas. These critical elements are just a few of the many features it has:

- Data Frame object with rapid and effective default and customized indexing.
- Instruments for transferring data between different file formats and in-memory data

objects. • Integrated processing for missing data and data alignment. Data sets can be repositioned and modified. • Label-based slicing, indexing, and subsetting of large data sets.

- We can add or remove columns from a data structure.
- Group by data while aggregating and transforming data. High-performance joining and merging of data.
- Functionality for Time Series.

4.1.4 OS

Python's OS module contains tools for interacting with the operating system. OS is one of Python's most common utility modules. This module allows us to use operating system-specific features while traveling. A large number of file system interface functions are available in the *os* and *os. path* modules.

4.1.5 Scikit-Learn

Using a consistent interface, a variety of machine learning, pre-processing, cross-validation, and visualization techniques can be built using the open-source Python package Sklearn. The primary characteristics of Scikit-Learns are as follows:

- Efficient and simple data mining and analysis methods. Other classifications, regression, and clustering algorithms are also mentioned, such as support vector machines, random forests, gradient boosting, and k-means.
- It can be utilized by anyone and repurposed in several circumstances.
- Because it is built on top of NumPy, SciPy, and matplotlib, it is quick.

4.1.6 Matplotlib

Matplotlib is an excellent Python visualization library for 2D array charts. Matplotlib is a cross-platform data visualization toolkit built on NumPy arrays and designed to work with the entire SciPy stack. John Hunter first mentioned it in 2002. One of the most important advantages of visualization is that it provides us with visual access to massive amounts of data in easily understandable images. Line, bar, scatter, histogram, and other plot types are available in Matplotlib.

4.1.7 Anaconda

Anaconda is a Python and R language distribution designed to simplify package management and deployment in scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, and other related fields). Anaconda is a popular choice for quick and easy implementation because it includes many of the tools needed for data science and machine learning. Anaconda, like a virtual environment, uses environments to separate different libraries and versions.

4.1.8 Python

Python is a high-level programming language that is dynamically semantic, interpreted, and object-oriented. Its high-level data structures, combined with dynamic typing and dynamic binding, make it ideal for Rapid Application Development as well as as a scripting or glue language for connecting existing components. Python's concise, simple syntax prioritizes readability, reducing software maintenance costs. Python supports modules and packages, which encourages program modularity and code reuse.

4.1.9 Jupyter Notebook

JupyterLab is the most recent web-based interactive development environment for notebooks, code, and data. Using the interface's flexibility, users can create and arrange workflows for machine learning, computational journalism, scientific computing, and data science. A modular architecture encourages extensions to increase and improve functionality.

4.1.10 Google Drive

Google Drive, a free cloud storage service, allows users to upload files and access them from anywhere on the planet. Documents, photos, and other data are synced between all of the customers' devices, including PCs, cellphones, and tablets. Google Docs, Gmail, Android, Chrome, YouTube, Google Analytics, and Google+ are among the other services and platforms that Google Drive is compatible with. Microsoft OneDrive, Apple iCloud, Box, Dropbox, and Sugar Sync compete with Google Drive.

4.1.11 TensorFlow

TensorFlow, a robust open-source machine learning framework, enables developers and academics to use cutting-edge machine learning from start to finish. Its extensive, scalable ecosystem of tools, libraries, and community resources enables rapid development and deployment of ML applications.

4.2 Hardware

- Processor: Intel i5-8265U @ 1.60GHz 1.80 GHz
- Ram: 8 GB DDR4 2600MHz
- OS: Windows 10(22000.978)

Chapter 5

Experimental Result & Discussion

This chapter will represent the result of what I got from my system and also represent the discussion about this system. Moreover, it will represent classification report, confusion matrix which is the evidence of my system, and also comparative analysis among the datasets I used in my system.

5.1 Metrics and Evaluation

Calculation of Accuracy (%) =����× 100%

Where n is the total number of test samples and c is the proportion of correctly identified test samples. The performance metrics shown below have binseen calculated for each classifier:

- ❑ **Precision:** The percentage of positive and negative instances correctly classified by the classifier out of all examples presented under each tag. In this case, the samples in the positive class have been identified as bullying victims, while the samples in the negative class have not.
- ❑ **Recall:** The proportion of examples correctly classified into a given class out of all cases where the classifier should have correctly classified.
- ❑ **F1Score:** The F1 score is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score

F1 Score = 2 * Precision * Recall / (Precision + Recall)

5.2 Confusion Matrix

Figure 5.1 Describes the confusion matrix of the MNB model with the TF-IDF. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the MNB model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem. In this confusion matrix of the MNB model, we can see that the true positive and true negative and true neutral numbers are 3164 and 3168 and 2422, respectively, while the false positive and false negative and false neutral numbers are significantly lower. Here, true positive and negative and neutral are much higher than false positive and negative and neutral, indicating that our model performs well in terms of class prediction.

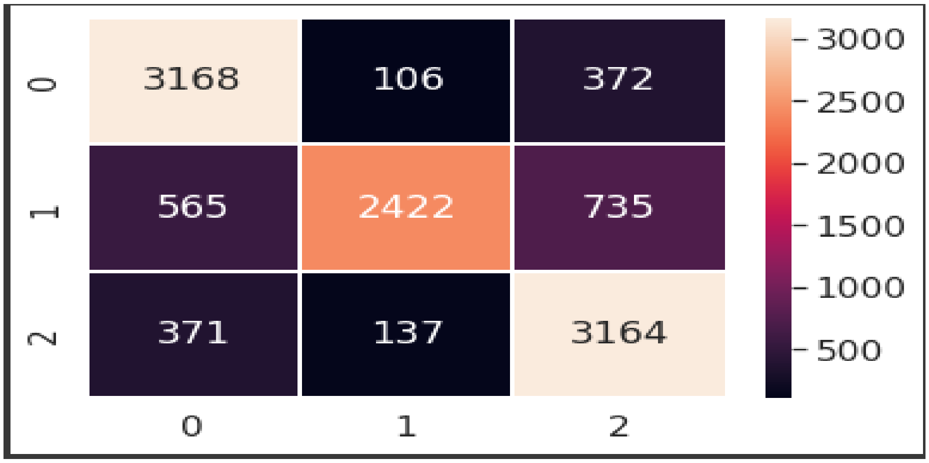


Figure 5.1: Confusion Matrix of Multinomial Naive Bayes

Figure 5.2 Describes the confusion matrix of the LR model with the TF-IDF. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the LR model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem. In this confusion matrix of the LR model, we can see that the true positive and true negative and true neutral numbers are 3127 and 3111 and 3559, respectively, while the false positive and false negative and false neutral numbers are significantly lower. Here, true positive and negative and neutral are much higher than false positive and negative and neutral, indicating that our model performs well in terms of class prediction.

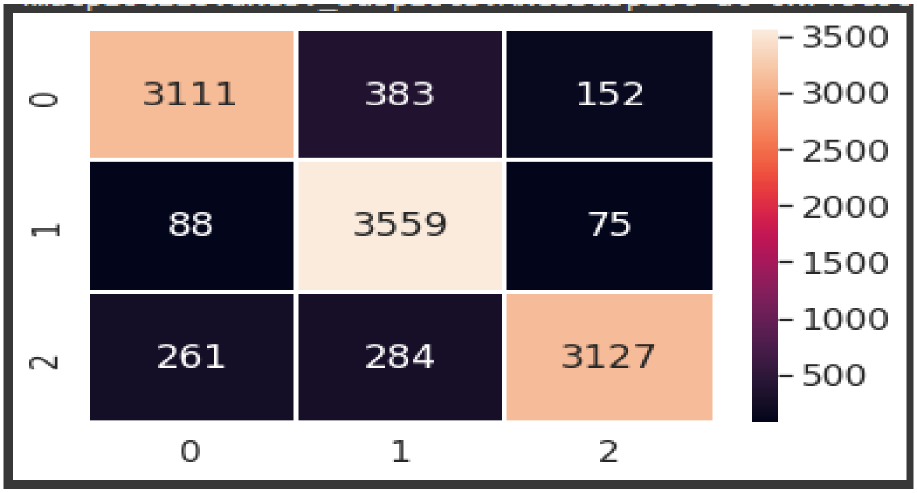


Figure 5.2: Confusion Matrix of Logistic Regression

Figure 5.3 Describes the confusion matrix of the DT model with the TF-IDF. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the DT model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem. In this confusion matrix of the DT model, we can see that the true positive and true negative and true neutral numbers are 3103 and 3257 and 3513, respectively, while the false positive and false negative and false neutral numbers are significantly lower. Here, true positive and negative and neutral are much higher than false positive and negative and neutral, indicating that our model performs well in terms of class prediction.

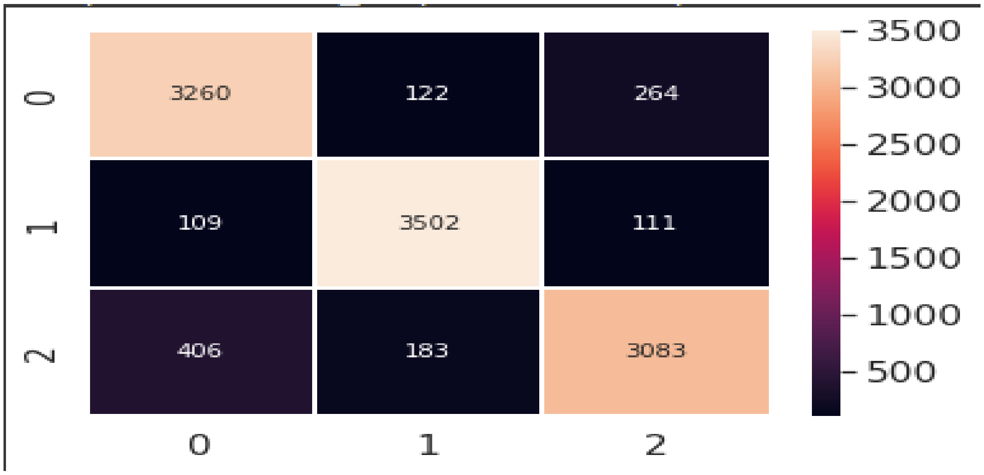
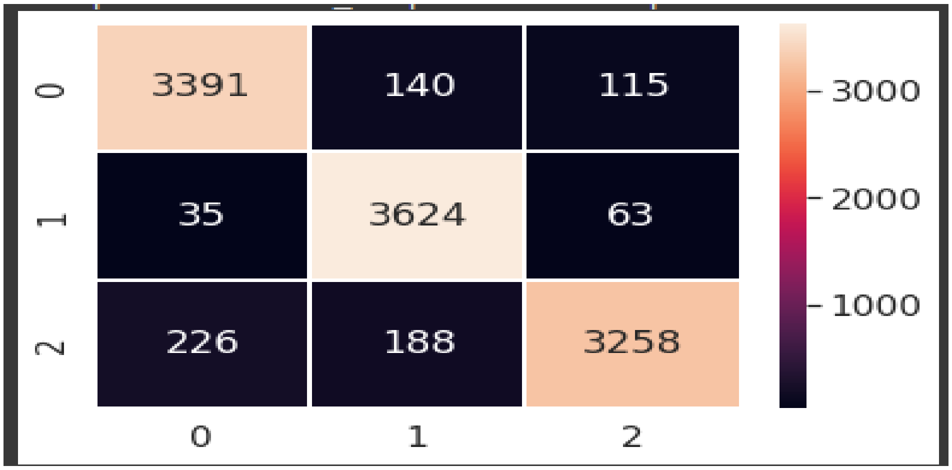


Figure 5.3: Confusion Matrix of Decision Tree

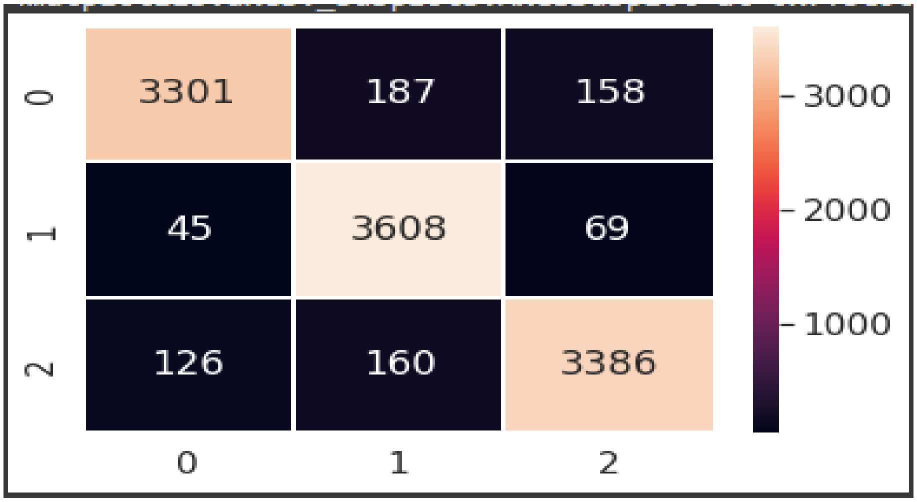
Figure 5.4 Describes the confusion matrix of the RF model with the TF-IDF. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the RF model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem. In this confusion matrix of the RF model, we can see that the true positive and true negative and true neutral numbers are 3258 and 3391 and 3624, respectively, while the false positive and false numbers and false neutral are significantly lower. Here, true positive and negative and neutral are much higher than false positive and negative and neutral, indicating that our model performs well in terms of class prediction



	0	1	2
0	3391	140	115
1	35	3624	63
2	226	188	3258

Figure 5.4: Confusion Matrix of Random Forest

Figure 5.5 Describes the confusion matrix of the SVM model with the TF-IDF. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the SVM model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem. In this confusion matrix of the SVM model, we can see that the true positive and true negative and true neutral numbers are 3386 and 3301 and 3608, respectively, while the false positive and false negative and false neutral numbers are significantly lower. Here, true positive and negative and neutral are much higher than false positive and negative and neutral, indicating that our model performs well in terms of class prediction.



	0	1	2
0	3301	187	158
1	45	3608	69
2	126	160	3386

Figure 5.5: Confusion Matrix of Support Vector Machine

Figure 5.6 Describes the confusion matrix of the KNN model with the TF-IDF. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the KNN model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem. In this confusion matrix of the KNN model, we can see that the true positive and true negative and true neutral numbers are 290 and 2914 and 3503, respectively, while the false positive and false negative and false neutral numbers are significantly lower. Here, true positive and negative and neutral are much higher than false positive and negative and neutral, indicating that our model performs well in terms of class prediction.

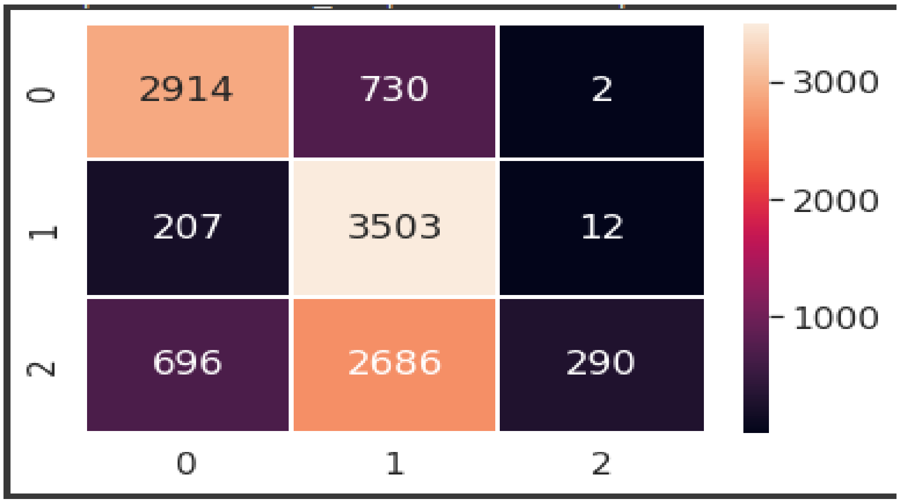


Figure 5.6: Confusion Matrix of K-Nearest Neighbors

Figure 5.7 Describes the confusion matrix of the LSTM model with the Word Embedding. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the LSTM model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem. In this confusion matrix of the LSTM model, we can see that the true positive and true negative and true neutral numbers are 3000 and 2468 and 2468, respectively, while the false positive and false negative and false neutral numbers are significantly lower. Here, true positive and negative and neutral are much higher than false positive and negative and neutral, indicating that our model performs well in terms of class prediction.

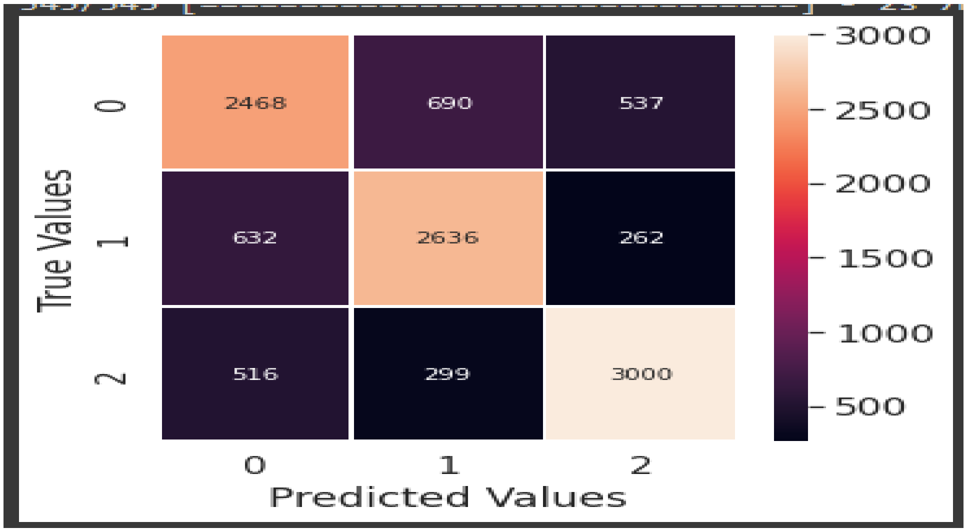


Figure 5.7: Confusion Matrix of Long Short-Term Memory

Figure 5.8 Describes the confusion matrix of the CNN model with the Word Embedding. From the confusion matrix, I calculated the Accuracy, Precision, Recall, and F1 scores of the CNN model. Here I used the weighted average for calculating Precision, Recall, and F1 score for our multiclass classification problem.

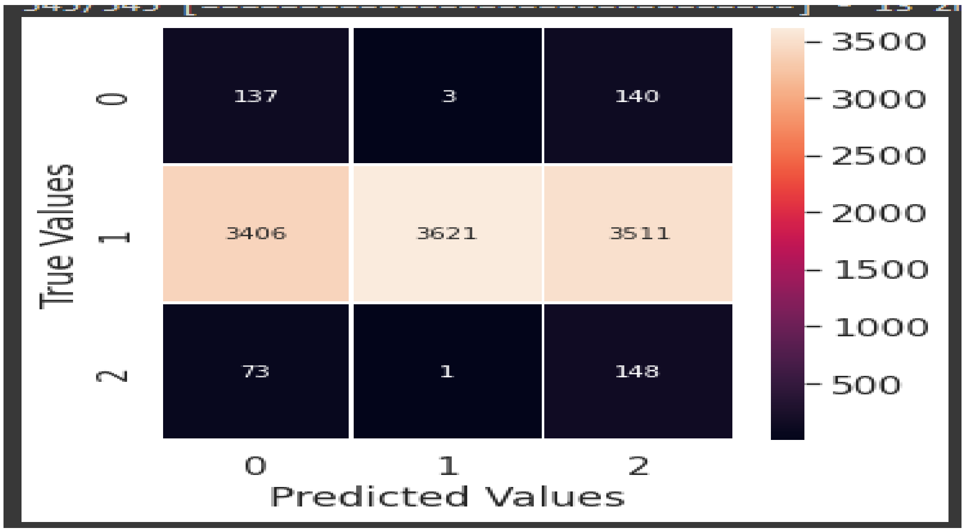
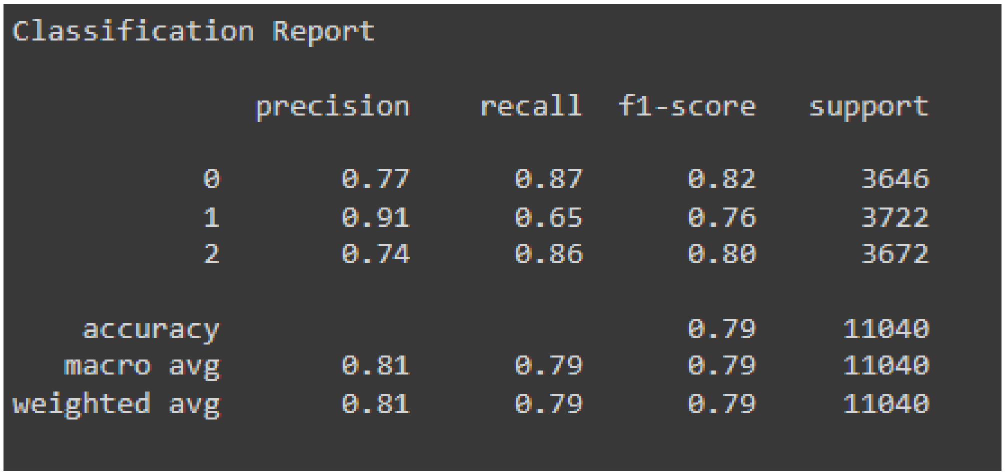


Figure 5.8: Confusion Matrix of Convolutional Neural Network

5.3 Classification Report

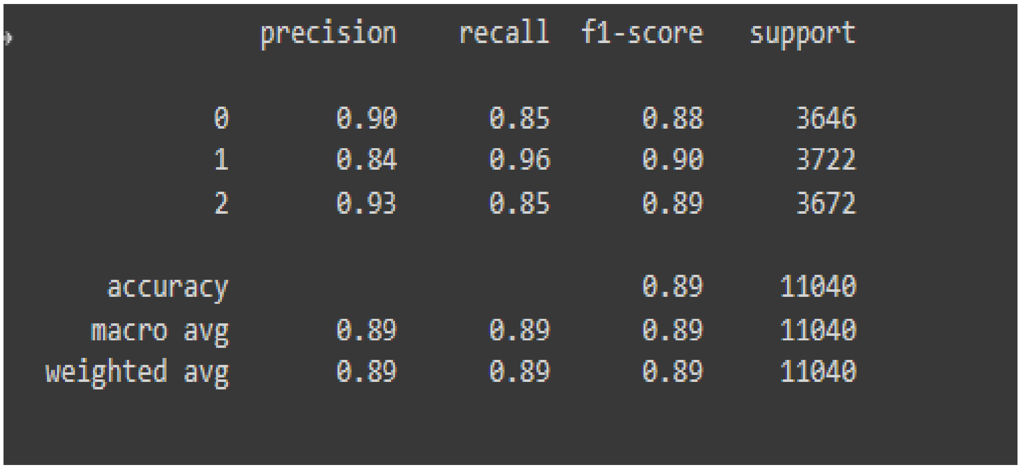
Figure 5.9 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 79.45%.



```
Classification Report

              precision    recall  f1-score   support

           0       0.77      0.87      0.82      3646
           1       0.91      0.65      0.76      3722
           2       0.74      0.86      0.80      3672

    accuracy                           0.79     11040
   macro avg       0.81      0.79      0.79     11040
weighted avg       0.81      0.79      0.79     11040
```

Figure 5.9: Classification report of Multinomial Naive Bayes

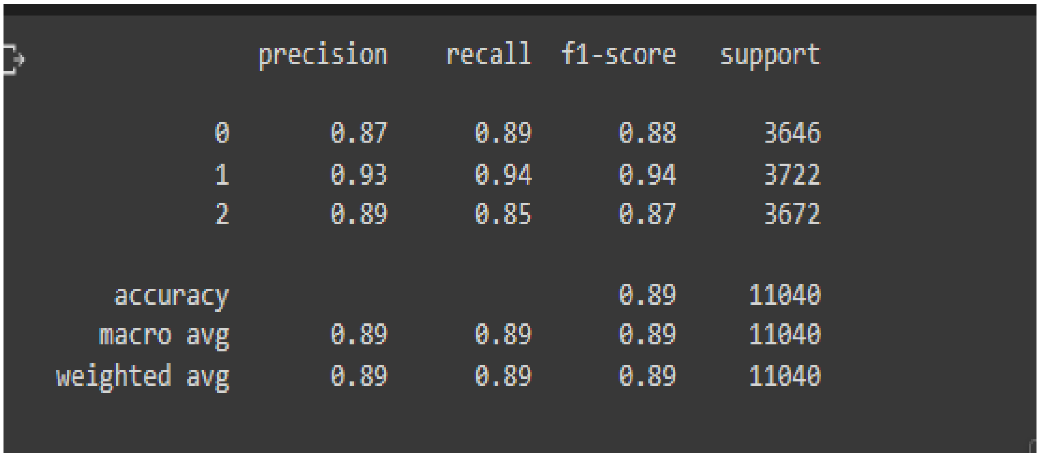
Figure 5.10 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 89%.



```
              precision    recall  f1-score   support

           0       0.90      0.85      0.88      3646
           1       0.84      0.96      0.90      3722
           2       0.93      0.85      0.89      3672

    accuracy                           0.89     11040
   macro avg       0.89      0.89      0.89     11040
weighted avg       0.89      0.89      0.89     11040
```

Figure 5.10: Classification report of Logistic Regression

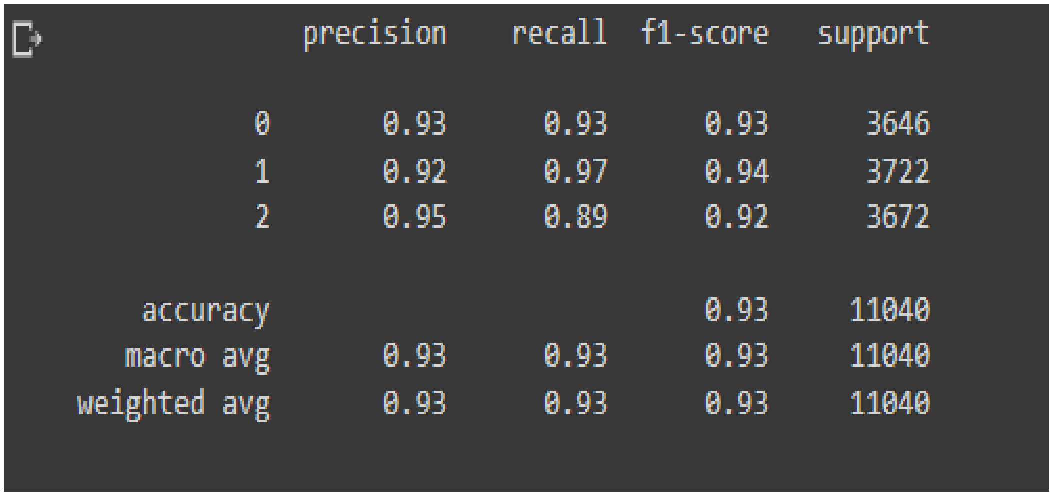
Figure 5.11 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 89.14%.



```
              precision    recall  f1-score   support

           0       0.87      0.89      0.88      3646
           1       0.93      0.94      0.94      3722
           2       0.89      0.85      0.87      3672

    accuracy                           0.89     11040
   macro avg       0.89      0.89      0.89     11040
weighted avg       0.89      0.89      0.89     11040
```

Figure 5.11: Classification report of Decision Tree

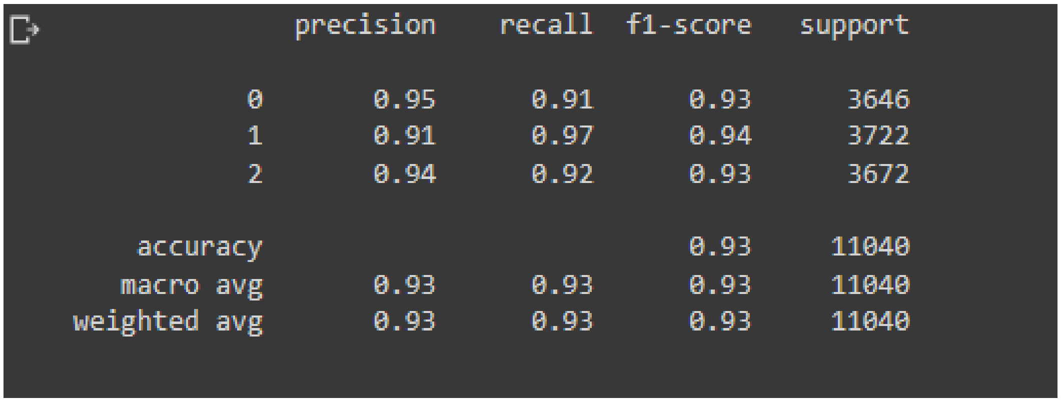
Figure 5.12 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 93.70%.



```
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      3646
           1       0.92      0.97      0.94      3722
           2       0.95      0.89      0.92      3672

    accuracy                           0.93     11040
   macro avg       0.93      0.93      0.93     11040
weighted avg       0.93      0.93      0.93     11040
```

Figure 5.12: Classification report of Random Forest

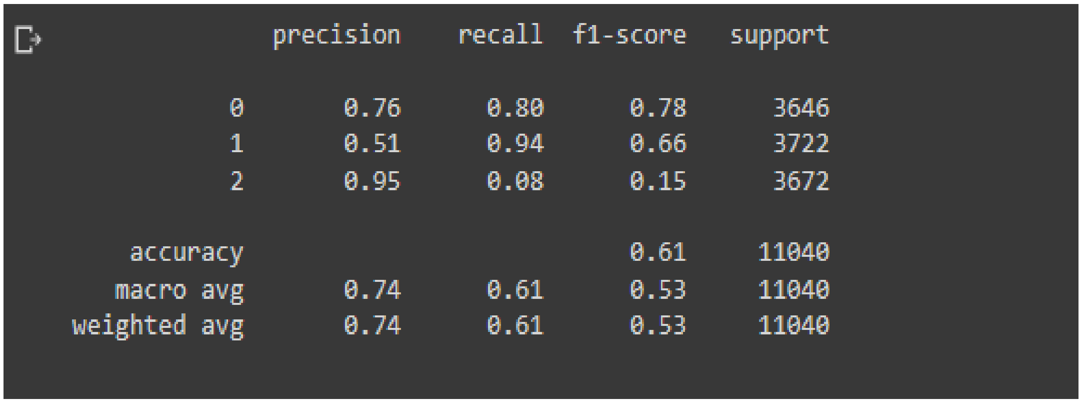
Figure 5.13 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 93.13%.



```
              precision    recall  f1-score   support

           0       0.95      0.91      0.93      3646
           1       0.91      0.97      0.94      3722
           2       0.94      0.92      0.93      3672

    accuracy                           0.93     11040
   macro avg       0.93      0.93      0.93     11040
weighted avg       0.93      0.93      0.93     11040
```

Figure 5.13: Classification report of Support Vector Machine

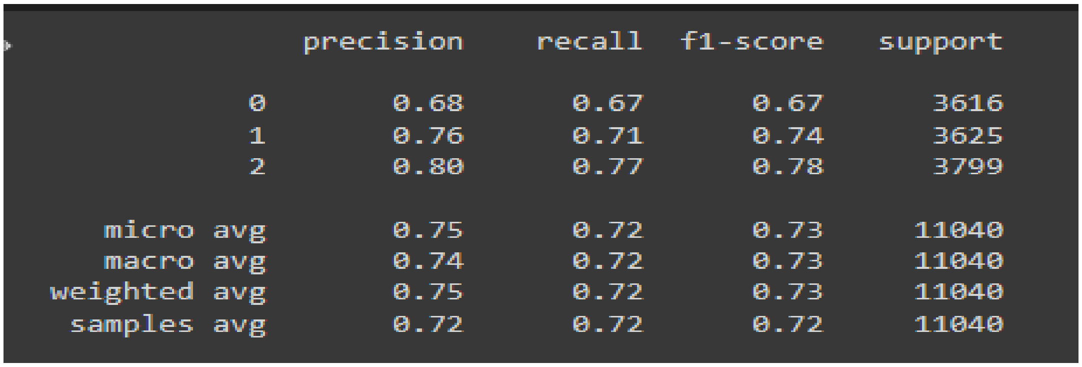
Figure 5.14 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 61.18%.



```
              precision    recall  f1-score   support

           0       0.76      0.80      0.78      3646
           1       0.51      0.94      0.66      3722
           2       0.95      0.08      0.15      3672

    accuracy                           0.61     11040
   macro avg       0.74      0.61      0.53     11040
weighted avg       0.74      0.61      0.53     11040
```

Figure 5.14: Classification report of K-Nearest Neighbors

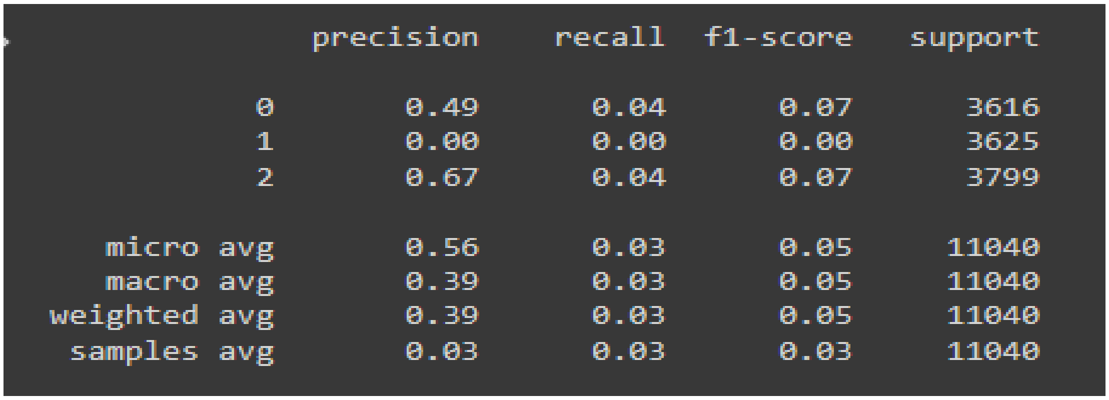
Figure 5.15 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 73.41%.



```
              precision    recall  f1-score   support

           0       0.68      0.67      0.67      3616
           1       0.76      0.71      0.74      3625
           2       0.80      0.77      0.78      3799

   micro avg       0.75      0.72      0.73     11040
   macro avg       0.74      0.72      0.73     11040
weighted avg       0.75      0.72      0.73     11040
 samples avg       0.72      0.72      0.72     11040
```

Figure 5.15: Classification report of Long Short-Term Memory

Figure 5.16 Describes the classification report, I got the overall overview of our model by getting the value of precision, recall, accuracy, and f1-score and also got an accuracy of 35.38%.



```
              precision    recall  f1-score   support

           0       0.49      0.04      0.07      3616
           1       0.00      0.00      0.00      3625
           2       0.67      0.04      0.07      3799

   micro avg       0.56      0.03      0.05     11040
   macro avg       0.39      0.03      0.05     11040
weighted avg       0.39      0.03      0.05     11040
 samples avg       0.03      0.03      0.03     11040
```

Figure 5.16: Classification report of Convolutional Neural Network

Performance Comparison of ML algorithm based on the TF-IDF given below:

Table 5.1 based on the TF-IDF, it's found out that all the ML has different accuracy due to their way of working methodology. Since RF has the highest accuracy (93.70%) among all the tested models, so in this case, I can say RF is the best classifier for our dataset based on TF-IDF.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
RF	93.55%	94%	94%	94%
SVM	93.13%	93%	93%	93%
DT	89.14%	89%	89%	89%
LR	89%	89%	89%	89%
MNB	79.45%	81%	80%	79%
KNN	61.18%	74%	62%	54%

Table 5.1 Comparison of ML algorithm based on the TF-IDF

Performance Comparison of ML algorithm based on the Count Vectorizer given below:

Table 5.2 based on the count vectorizer, it's found out that all the ML has different accuracy due to their way of working methodology. Since RF has the highest accuracy (82.78%) among all the tested models, so in this case, I can say RF is the best classifier for our dataset based on Count Vectorizer.

Algorithm	Accuracy	Precision	Recall	F1-Score
RF	82.78%	83%	83%	83%
SVM	81.92%	83%	82%	81%
DT	79.77%	80%	80%	80%
LR	83.31%	84%	83%	83%
MNB	71.59%	73%	72%	71%
KNN	61.29%	71%	61%	58%

Table 5.2 Comparison of ML algorithm based on the Count Vectorizer

Performance Comparison of Deep Learning Algorithms given below:

Table 5.4 based on the Word Embedding, it's found out that all the Deep Learning has different accuracy due to their way of working methodology. Since LSTM has the highest accuracy (73.41%) among all the tested models, so in this case, I can say LSTM is the best classifier for our dataset based on Word Embedding.

Algorithm	Accuracy	Precision	Recall	F1-Score
LSTM	73.41%	74%	72%	73%
CNN	35.38%	39%	0	0

Table 5.3 Comparison of Deep Learning Algorithm

5.4 Comparative Analysis

In this section my research was tried to be compared with target paper works. The comparative analysis was based on accuracy. The comparison can be seen in the table below-

Paper Title	RF	SVM	DT	LR	MNB	KNN	LSTM	CNN
Sentiment Analysis For Product Review[1]	85%	89.88%	71.91%	89.51%	86.26%	78.54%	87.97%	81.66%
Proposed Model	93.55%	93.13%	89.14%	89%	79.59%	61.18%	73.41%	35.38%

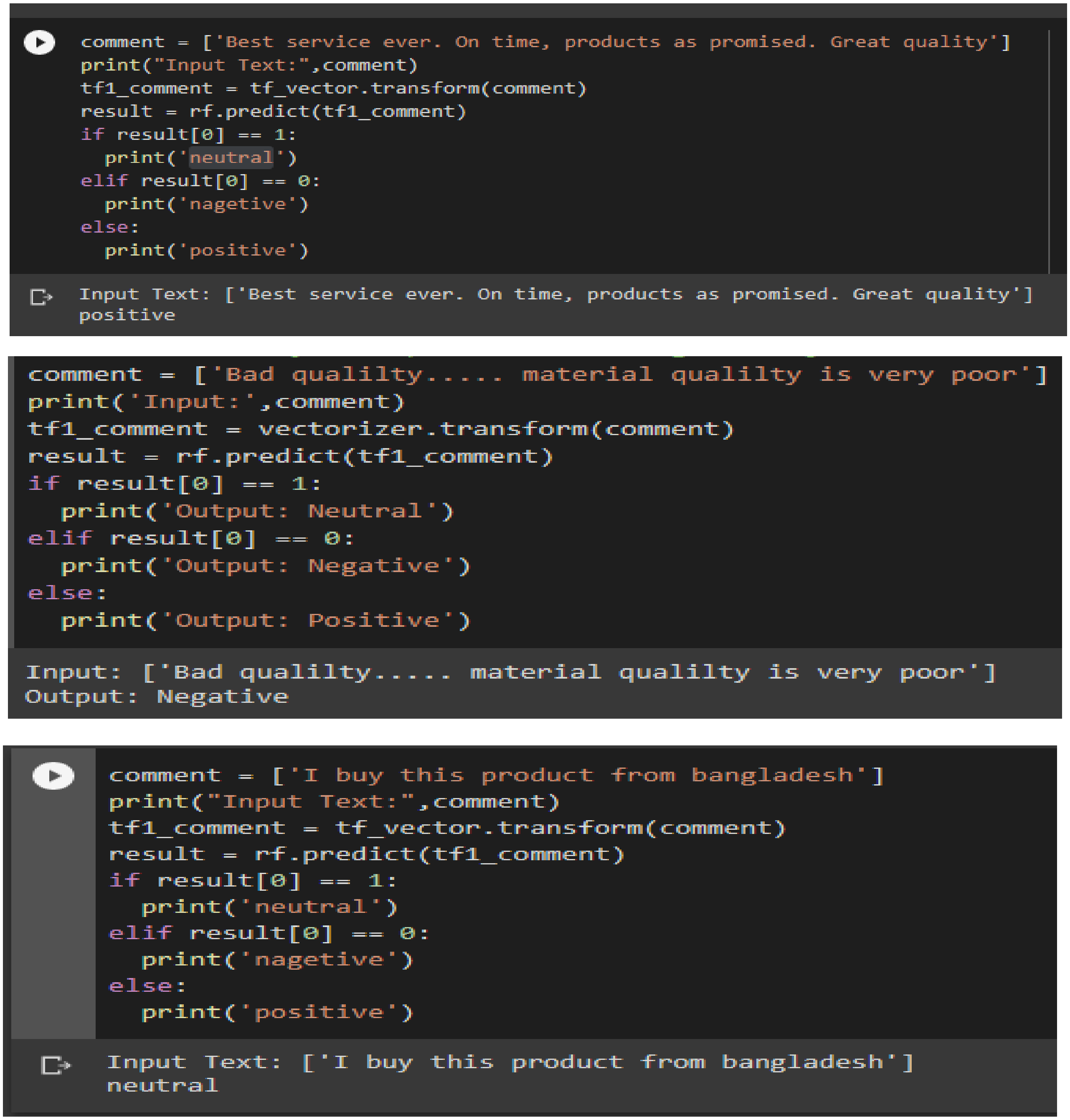
Table 5.1: Comparative Analysis

Different researches listed in the table have conducted different pre-processing steps and feature extraction processes. As part of my research, I had to improve all of the extraction processes and preprocessing steps and select the ones with the highest accuracy. Pull based active learning process have contributed labeling and selecting the best reviews as our training and testing data. Use of different preprocessing process helped sorting out unnecessary words. And finally taking the best features extracted from the datasets and learning through proper classifiers it was possible to attain greater accuracy. From the table it can be decided that the approaches used in approaches my proposed model shows more effectiveness and could achieve a better result than target paper works.

5.5 Model’s Prediction

Three classes:

- Positive: 2
- Negative: 0
- Neutral: 1



```
comment = ['Best service ever. On time, products as promised. Great quality']
print("Input Text:",comment)
tf1_comment = tf_vector.transform(comment)
result = rf.predict(tf1_comment)
if result[0] == 1:
  print('neutral')
elif result[0] == 0:
  print('nagetive')
else:
  print('positive')
```

```
Input Text: ['Best service ever. On time, products as promised. Great quality']
positive
```

```
comment = ['Bad qualilty..... material qualilty is very poor']
print('Input:',comment)
tf1_comment = vectorizer.transform(comment)
result = rf.predict(tf1_comment)
if result[0] == 1:
  print('Output: Neutral')
elif result[0] == 0:
  print('Output: Negative')
else:
  print('Output: Positive')
```

```
Input: ['Bad qualilty..... material qualilty is very poor']
Output: Negative
```

```
comment = ['I buy this product from bangladesh']
print("Input Text:",comment)
tf1_comment = tf_vector.transform(comment)
result = rf.predict(tf1_comment)
if result[0] == 1:
  print('neutral')
elif result[0] == 0:
  print('nagetive')
else:
  print('positive')
```

```
Input Text: ['I buy this product from bangladesh']
neutral
```

Figure 5.17: Prediction for the Machine learning algorithm

Chapter 6

Conclusion and Future Work

This chapter develops conclusions and evaluates the outcomes based on the observations. It also identifies certain limits as well as the future works of the research.

6.1 Conclusion

Customers are increasingly dependent on online evaluations as a result of an evolutionary transition from offline to online markets. Online reviews have evolved into a forum for establishing trust and influencing customer purchasing habits. With such reliance, there is a need to handle such a vast amount of evaluations and deliver genuine reviews to the consumer. My research aims to do this by conducting sentiment analysis on customer reviews and classifying them as positive, negative, or neutral. After balancing the data with almost equal ratio of positive, negative and neutral reviews, eight classification models have been used to classify reviews. Out of the eight classifiers, i.e. Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Convolutional neural network (CNN), and Long short-term memory (LSTM), predictive accuracy of RF is found to be the best. The accuracy results have been cross validated and the highest value of accuracy achieved was 93.55% for RF among the eight models. I believe that my suggested approach would reduce consumer frustration while shopping online since they will be able to examine product reviews based on the ratio of previous customers' positive, negative, and neutral comments. It can also be beneficial to the vendor because it allows him to spot product flaws and give better customer service.

6.2 Future work

This Work can be extended in the following manner in future

- Try to use more feature extraction.
- To use more Deep Learning Models.
- To use more Deep Learning Models.
- Improve deep learning model accuracy
- Adding more data.

References

1. Najma Sultana, Pintu Kumar, Monika Rani Patra, Sourabh Chandra And S.K. Safikul Alam,“Sentiment Analysis For Product Review ”, researchgate.net,ICTACT Journal On Soft Computing, April 2019.
2. Levent Guner, Emilie Coyne, Jim Smit, “Sentiment analysis for Amazon.com reviews ”, researchgate.net, Big Data in Media Technology ,March 2019.
3. Tanjim Ul Haque, Nudrat Nawal Saber, Faisal Muhammad Shah, “Sentiment Analysis on Large Scale Amazon Product Reviews ” ,IEEE, International Conference on Innovative Research and Development (ICIRD),June 2018.
4. Rajkumar S.Jagdale, Vishal S.Shirsat and Sachin N.Deshmukh,Sentiment Analysis on Product Reviews Using Machine Learning Techniques,researchgate.net,January 2019.
5. Zeenia Singla, Sukhchandan Randhawa, Sushma Jain, “Sentiment Analysis Of Customer Product Reviews Using Machine Learning ”, International Conference on Intelligent Computing and Control (I2C2),June 2017.
6. Palak Baid, Apoorva Gupta and Neelam Chaplot on a research paper named ‘Sentiment Analysis of Movie Reviews using Machine Learning Techniques’, researchgate.net, International Journal of Computer Applications (0975 – 8887), December 2017.
7. LI YANG ,YING LI , JIN WANG AND R. SIMON SHERRATT ‘Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning’ ,IEEE, January 27, 2020.
8. Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat and Priyanka Badhani ’ Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python’, researchgate.net, International Journal of Computer Applications, May 2017.
9. Zeenia Singla, Sukhchandan Randhawa, and Sushma Jain,’ STATISTICAL AND SENTIMENT ANALYSIS OF CONSUMER PRODUCT REVIEWS’,IEEE, December 2017.
10. Mst.Tuhin Akter, Manoara Begum and Rashed Mustafa ‘ Bengali Sentiment Analysis of E-commerce Product Reviews using K-Nearest Neighbors’IEEE, International Conference on Information a nd Communication Technology for Sustainable Development (ICICT4SD), April 2021.
11. Dataset name “ Clothing, Shoes, and Jewelry ”. Available : http://jmcauley.ucsd.edu/data/amazon/index_2014.html.
12. “Sentiment Analysis,” towardsdatascience,https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17.

13. “Natural Language Processing,” becominghuman.ai, https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32.
14. How to Use Word Embedding Layers for Deep Learning with Keras,https://machinelearningmastery.com/use-word embedding-layers-deep-learning-keras/.
15. A guide to the types of machine learning algorithms and their applications, https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html.
16. A Simple Introduction to Natural Language Processing. (2021). Retrieved 8 June 2021, from https://becominghuman.ai/a-simple-introduction-to-naturallanguage-processing-ea66a1747b32.
17. P. V. Rajeev and V. S. Rekha, “Recommending products to customers using opinion mining of online product reviews and features,” in 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], IEEE, 2015.
18. M. S. Elli and Y.-F. Wang, “Amazon reviews, business analytics with sentiment analysis,” Elwalda, Abdulaziz, et al.“Perceived Derived Attributes of Online Customer Reviews, 2016.
19. M. U. Islam, F. B. Ashraf, A. I. Abir, and M. Mottalib, “Polarity detection of online news articles based on sentence structure and dynamic dictionary,” in 2017 20th International Conference of Computer and Information Technology (ICCIT), IEEE, 2017.
20. S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, “Improving the accuracy of pretrained word embeddings for sentiment analysis,” arXiv preprint arXiv:1711.08609, 2017.