Bengali Sentiment Analysis Based on Product Reviews: Unveiling Consumer Voices

by

Mehedi Hassan
20201148
Mujtaba Wasif Pritom
20201130
Shahidul Islam Fuad
20201055
Saif Ahmmed Sifat
21101341

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

Department of Computer Science and Engineering
Brac University

Declaration

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

Student's Full Name & Signature:

Mujtaba Wasif Pritom	Mehedi Hassan
20201130	20201148
Saif Ahmmed Sifat	Shahidul Islam Fuad
21101341	20201055

Approval

The thesis/project titled "Bengali Sentiment Analysis Based on Product Reviews: Unveiling Consumer Voices" submitted by

1. Mujtaba Wasif Pritom (20201130)
2. Mehedi Hassan (20201148)
3. Saif Ahmmed Sifat (21101341)
4. Shahidul Islam Fuad (20201055)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October, 2024.

Examining Committee:

Supervisor:
(Member)

Dewan Ziaul Karim

Lecturer
Department of Computer Science and Engineering
Brac University

Co Supervisor:
(Member)

Md Faisal Ahmed

Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Thesis Coordinator)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

These days, customers are more keen to buy products online rather than going to a shop or market. However, they often fear about the quality of products, as there is no way to measure them before buying them. As a result, most buyers rely on the reviews of other customers who have already purchased the product. For this reason, customer reviews are very crucial for the e-commerce industry. A popular method for assessing the quality of a product is opinion mining, which is also called sentiment analysis. It is a method of extracting emotion from a text using natural language processing (NLP). In this study, we have collected data from an e-commerce site named Daraz and introduced a new dataset that contains Bengali reviews. A total of 48000 reviews were collected, of which 22000 were Bengali. 15000 are in English, while the rest of 9000 are in "Banglish" (Romanized Bengali). Several data preprocessing techniques were used to introduce a new clean dataset that only contains Bengali reviews. Five machine learning algorithms—Naive Bayes, Random Forest, Gradient Boosting Classifier, Logistic Regression, and Support Vector Machine (SVM)—and three deep learning models—BiLSTM, Multilingual BERT, and BanglaBERT—were implemented to evaluate our work. Our work should help the sellers filter out the best products that are popular among consumers.

Keywords: NLP, Naive Bayes, Random Forest, Gradient Boosting Classifier, Logistic Regression, Support Vector Machine, e-commerce.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the years, the number of e-commerce sites has been growing exponentially. If we search for our favorite products in the browser, we will see hundreds of e-commerce sites selling our desired material. These online marketplaces serve a vital part in our day-to-day lives by giving us access to our preferred goods and fostering a good connection with consumers.

From food to hardware equipment, every day we feel the need to purchase something out of necessity. As a result, we have to go to different shops to purchase our necessary commodities. But people these days are busier than ever. Therefore, for the majority of us customers, traveling to a real store to purchase our essential materials becomes a burden. This is where e-commerce steps in. E-commerce sites have introduced the concept of buying and selling products without being physically present. Additionally, these platforms have their own delivery networks, so users don't even need to go to a market to acquire their purchases. Besides, the review system also helps consumers check the quality of a product.

The method of examining perspective, thought, and perception in a text—which may take the shape of many languages—is known as sentiment analysis. It is the process of analyzing public opinions on the internet and detecting the emotions those opinions carry. The main purpose of this approach is to categorize the tone or feeling of text into several variables, like positive, negative, and neutral. The research on this process provides the customer and sellers with a whole viewpoint of the market from which they can make better decisions on their purchases and sales, respectively. As a huge number of e-commerce sites exist online, numerous product reviews can be found online, which makes it perfect for sentiment analysis.

The main goal of this study is to gather consumer reviews from an e-commerce site and classify each review as positive, negative, or neutral. We have mainly focused on the Bengali reviews. For this research, we have collected the required data from the largest e-commerce site in Bangladesh, named Daraz. Various data pre-processing and NLP approaches and different types of machine learning and deep learning models were implemented. Our aim is to contribute to a positive outcome in online business and also to the Natural Language Processing (NLP) field, as there is less research conducted in terms of Bengali. Moreover, this thesis should help the sellers filter out the best products that are popular among consumers.

1.1 Research Problem

In our day and age, the number of online shoppers is increasing and is currently higher than ever. At the end of 2024, it is estimated, the number of online subscribers in Bangladesh will reach about 130 million. With this increasing number of online users each day, various e-commerce sites collect huge amounts of data consisting of reviews from various categories of products. Handling this enormous amount of data and processing and categorizing these reviews based on emotional sentiment is a necessity. It provides a better perception for both sellers and customers by presenting crucial insight and assisting in well-informed decision-making on products. Due to the history of online scamming in the past, people nowadays put more faith in products that have a larger number of reviews. The large number of positive or neutral reviews on any product is a sign of authenticity, whether the product is a smartphone, laptop, book, or a mere pencil. A lower quantity of reviews on any product, regardless of the majority of its reviews being positive, puts a negative perspective of the product on customers. People prioritize reviews as they show their opinions and personal experiences with products and It is the only way to fully understand the other user's impression of the products.

In this large dataset of reviews, there are great possibilities for observing instances where the reviews do not convey the reality of the products. This can lead to noise in the data, thus making it complicated to extract the true sentiment. Online spammers post irrelevant content, fake opinions, and other things that are meaningless and have no value in research. Moreover, it skews the result and shows a completely different outcome than the actual sentiment. The usage of sarcasm and multipolarity is also a challenge to identifying the true sentiment. Besides, ground truth labels that provide sentiment polarity are not usually available, which helps evaluate consistent results. Ground truth is key for the training and testing of sentiment analysis models. It provides a tag of opinions that helps categorize the review into a sentiment whether it be neutral, positive, or negative.

Bengali is a broadly spoken language in Southeast Asia, with over 250 million people having it as their primary language. Despite the language being spoken by a vast majority of people, there is very little research being conducted on this language. There is also an inadequate number of effective and built-in pre-processing tools and models in the context of Bengali. For Western Roman languages like English, Spanish, etc there are features for example POS tagging, lemmatizing, stemming, etc. but none for Bengali. Furthermore, there is a scarcity of proper datasets, as most of the collection has issues of overused slang, spelling mistakes, use of romanized Bengali, etc.

Sentiment analysis is currently a widely used approach to determining public emotion. Much research has been conducted on this topic. Most research was able to provide a decent result and approach to detecting public opinion. Implementing these approaches to business analysis has been pretty successful so far. The e-commerce sites are filled with customer reviews, which are valuable resources for opinion mining. These reviews can be used to detect consumer sentiment, which will help businesses improve their product quality, product recommendations, customer

service, and market research.

1.2 Research Objective

Keeping customers satisfied is necessary for any business that sells products or provides any kind of service. Since people are free to express their points of view on the internet, product reviews have become a key source for understanding whether customers are satisfied with the product. If a product has more negative reviews, then new customers will hesitate to buy that product. Conversely, if a new customer notices a product has positive reviews most of the time, the chances for that customer to buy that product are very high. Hence, customer reviews determine the quality of a service or product. Thus, sentiment analysis becomes a valuable tool for understanding customer satisfaction, as the primary objective of opinion mining is to determine sentiment based on consumer opinions. Our intention is to apply a more effective approach for identifying emotions in Bangla reviews.

Our objectives:

- Contribute to the field of Natural Language Processing for Bengali language.
- Introduce a new dataset that contains Bengali Reviews
- Analyze how much consumers rely on customer reviews
- Implementation of machine learning models capable enough to detect the sentiments: positive, negative, and neutral attached to each review.
- Implementation of deep learning models capable enough to detect the sentiments: positive, negative, and neutral attached to each review.

Chapter 2

Literature review

Opinion mining is one of the most popular NLP techniques used in e-commerce establishments. A lot of research has been conducted on this NLP task. However, most research was done in English, but a few can be found in Bangla, which makes it a good opportunity for new researchers to contribute not only to Bangla sentiment analysis but also to the field of Bangla natural language processing.

The goal of this research is to identify sentiment in user reviews, and their data collection source is an e-commerce platform named Daraz. The dataset consists of 7905 reviews on various products, and they have only focused on Bengali texts. Several data preprocessing tasks were done, such as removing non-Bengali text and separating emoji, removing punctuation and Bangla digits, extracting features, etc. K-Nearest Neighbors, Random Forest Classifier, Logistic Regression, and Support Vector Machine algorithms were implemented for research. KNN outperformed the other algorithms, with a 96.25 accuracy rate as well as 0.96 precision, recall, and f1-score. Although they have managed to get good results using their model, the dataset seems smaller. They could have developed their model for a larger dataset [1].

This study aim to detect sentiment from Bangladeshi restaurant reviews, and the authors collected their data from two well-known food delivery apps: Food Panda and Hungrynaki. 2000 reviews were collected from FoodPanda and 18,000 reviews from Hungrynaki. For data preprocessing, they have removed null values and unnecessary rows, lowercase every word, and removed punctuation and emojis. Furthermore, the Bangla texts were translated, and the Banglish texts were transliterated using the Google Cloud Translation API. Pre-trained models used for this thesis are DistilBERT, AFINN, and RoBERTa. Here, AFINN and RoBERTa are machine learning models and a non-machine learning algorithm used for this thesis is DstilBERT. The accuracy rates of AFINN, RoBERTa, and DistilBERT are 73%, 74%, and 77%, respectively. The reason for these models' poor accuracy is that the amount of data for this NLP task is not enough [13].

Three popular publicly accessible movie review datasets for binary sentiment categorization (MR, IMDB, and SST2) were employed for this research. The researchers have used one layer of bidirectional LSTM architecture to determine the binary sentiments (positive or negative) from these datasets. The MR and IMDB datasets are

balanced. The MR dataset contains 10,622 reviews, among which 5,331 fall under the positive sentiment and the other 5,331 fall under the negative sentiment. In addition, IMDB datasets have 50,000 reviews, of which 25,000 are positive and the other 25,000 are negative. However, the SST2 dataset is not balanced, containing 9,613 reviews. To feed every unique word from the review, they have converted every unique word into low-dimensional vectors using a pre-trained 300-dimensional GloVe vector. The authors used 10-fold cross-validation for the IMDB and MR datasets to train the data. However, the SST2 dataset was divided into a (6920/872/1821) train/valid/test approach. The RMSprop optimizer, cross-entropy loss function, batch size of 64, one BiLSTM layer with 16 nodes, a drop-out rate of 0.3, and the L2 regularizer were used as hyperparameters for this study. After running their model on each of these preprocessed datasets, they received satisfactory results. The F1 score achieved for the MR dataset is 80.495% and the dataset gained 80.50% accuracy rate which was better than the related works during the time they were conducting their research. The IMDB dataset gained accuracy and F1 scores of 90.585% and 90.580%, respectively, which is better than the studies related to their work. Finally, the SST2 dataset achieved accuracy and F1 scores of 85.780% and 85.775%. Their model outperformed all the other models that were implemented in this dataset except the Capule-LSTM, which was 0.62% better than their proposed single-layered BiLSTM model. After completing their study, the authors concluded that their model performs better on balanced datasets. Although they have done a good job building their model, this model could have been more efficient if Adam's optimizer had been used [6].

This paper focuses on sentiment classification, which uses machine learning for Bengali language sentiment analysis. Bangla ranks seventh language all over the world and ranks fifth when it comes to native speakers. There is so much research concerning NLP in the English language but there is less for Bangla. Sentiment analysis is a technique where a machine can identify the sentiment in a statement to make improvements in several areas. 4177 Bengali sentences of the unique dataset were used for the work and the algorithm, Logistic Regression , Decision Tree, Support Vector Machine, K-nearest Neighbors, Random Forest, and Naïve Bayes classifier were applied to make an intelligent system that can detect the emotion of Bengali sentence, comparing the best one with the highest accuracy so to generate most precise output. The Bengali dataset contains two types of emotion which are positive and negative where 2300 are positive and 1800 are negative and all of the data are manually collected from different online sources such as Facebook, YouTube, and various online news portals. There are two columns in the dataset where the first column named sentence is an object type and the second column named `emotion_2` is an integer type. The data is filtered so that all the noise that is not related to the Bengali alphabet such as quotations, hyperlinks, commas special characters, etc are removed. Not to mention, as the machine can not identify short Bengali words, therefore the full form of the corresponding Bengali words is replaced using a Bengali phase tagger. Then after ensuring there are no null values in the two columns, the data are labeled with positive and negative sentiments where positive is denoted by one and negative is denoted by zero. Then for training and testing, among the total number of data, eighty-five percent are used for training and the remaining fifteen percent are used for testing. They observed a score of 350 in the confusion matrix,

however they obtained an accuracy of 62.36% in the Support Vector Machine algorithm. The accuracy of the K-Nearest Neighbour algorithm is 56.29%. With regard to the Decision Tree, the accuracy gained is 58.05%, while the True and False predictions in the confusion matrix are 362, 266, and 627, respectively. With Logistic Regression, the accuracy is 62.20% since there are 237 erroneous predictions and the remaining predictions are true. Comparably, the accuracy obtained for Random Forest is 67.30%, with 422 correct predictions and 205 incorrect predictions out of a total of 205, and for Naïve Bayes, it is 65.70%, with true predictions being 412 and false predictions being 215. Because Random Forest outperformed all other algorithms in terms of accuracy and could correctly identify the sentiment of the words, it was chosen to be the model's algorithm [9].

The paper focuses on a study of SVM and Naive Bayes classifiers for Sentiment Analysis applied on Amazon product reviews. Nowadays, people are mainly interested in buying products on e-commerce websites instead of offline and physical markets as it is more convenient and time-efficient. To know about the product in the e-commerce market, the customer has to go through the reviews other customers gave earlier and here reading thousands of reviews is not suitable for the buyer and that is why this paper targets customers' feedback on different products and then creates a learning model to divide a variety of reviews. Nearly 1,47,000 book reviews have been processed for analysis. As Amazon review feedback comes in the "5-star" ratings, therefore, the "3-star" ratings are discarded because it is regarded as a neutral review and the other ratings are taken for the next step which is data preprocessing. In data preprocessing, there are three steps which are tokenization, removing stop words, and then using the global constant to fill up the missing value. In the tokenization, some characters like punctuation marks are removed. In the removing stop words, the stop words are removed. Then, the system searches in the dataset for the missing value, and then this missing value will be replaced with the relevant constant. During the feature extraction process, the dataset's features are extracted in three stages and they are frequent noun identifier, relevant noun removal, and term frequency-inverse document frequency. There are four categories into which the data from the confusion matrix are classified and they are True Positive, False Positive, True Negative, and False Negative. At the end of this research, a comparison between the Naive Bayes classifier and SVM analyzed the divisive nature of the sentiment expressed in Amazon product evaluations, where the models were trained using about 2250 features from nearly 6000 datasets, while the models also processed nearly 4000 test sets. The study showed a precision, recall, and f1 score of 82.853%, 82.884%, and 82.662% respectively for the SVM classifier and 83.990%, 83.997%, and 83.993% respectively for the Naive Bayes classifier. Therefore, the model generates 84% accuracy for the SVM classifier and 82.875% accuracy for the Naive Bayes classifier. The experimental results therefore confirm that the SVM classifier can polarize the feedback of Amazon products with a higher accuracy rate than the Naive Bayes classifier [5].

This work examines the emotion of Price Hike using LSTM-ANN and Bangla social media comments. The price hike increases as the related products and services expenses increase. For instance, Bangladesh recently saw price increases as a result of an unexpected rise in fuel costs. In this type of situation, social media has

turned out to be a better place to communicate with others, and share opinions and thoughts regarding the current situation, and therefore from this social media, we can analyze public sentiments about the current situation. In this study, a dataset of 2000 sentences with public responses to the most recent price increase is constructed. Next, a hybrid deep learning architecture that outperforms the prior state-of-the-art models is suggested for sentiment analysis. After that, the response of the public to price increases will be examined using the suggested model. Three categories which are positive, negative, and neutral are used to classify public opinions. In the dataset, the transliterated sentences are removed which means the sentences written using the English alphabets to represent Bengali phonetics are removed. In the annotation of the collected dataset, out of 2000 comment instances, 253 are labeled as positive polarity, 1359 as negative polarity, and 388 as neutral polarity. In the data preprocessing, all the special characters, digits, emojis, and punctuations are removed first and then the text is converted to tokens where Keras tokenizer is used. Here, the sentiment level is factored so that positive polarity becomes 0, negative polarity becomes 1, and neutral polarity becomes 2 and then added to the `pad_sequence` function to convert it to a sequence. Then, the associations of words and numbers are then generated using the `fit_on_text` function, with each unique number assigned based on the tokenizer function, where `max_word` is 5000, which means 5000 unique words and numbers will be assigned against words. Then, the words are embedded for the next phase where Sequential() starts the model creation. Then, the network `SpatialDropout1D()` is used to drop the entire feature map and then the dataset is fed to an LSTM layer where the LSTM will work like a recurrent neural network. Not to mention, in this study, a train-test ratio of 70% to 30% is used. Different deep learning architectures are examined where in the F1-score comparison, the proposed model LSTM-ANN outperforms the other three efficient deep learning architectures. After that, the model is contrasted with a few earlier study methodologies, and it is found that, out of all the effective models, the suggested model has the highest accuracy. Although successful, the suggested LSTM-ANN model has many drawbacks. Specifically, while it is reliable in classifying comments or words with positive polarity, it struggles to label negative or neutral statements. One possible explanation is that certain remarks or phrases can be classified as either neutral or negative. Therefore the model picks up one for the expected other due to confusion [3].

The main objective of this thesis is to refine the constraints of sentiment analysis on product reviews written in Bengali, English, and Romanized Bengali.. Exerting focus only on one selected form of language may never contribute to the development of E-commerce. The data set is then labeled into five classes. In this paper, the language has been categorized into negative, positive, slightly positive, slightly negative, and neutral. Applied six machine learning algorithms (MNB, logistic regression, SVM, Random Forest, KNN, and Decision Tree). The dataset has been collected from DARAZ due to it being the most used e-commerce website in Bangladesh. To overcome imbalanced training data (significant disparity in class sizes) random oversampling techniques are applied and make all outnumbered classes equal to the majority (it enlarges the minority class dataset). To tackle imbalanced dataset difficulties, a strategy of oversampling is implemented to resample the data. TF-IDF was used for optimal feature extraction, converting strings to

numerical data for ML models. After training the models it is evaluated with the test dataset. Using the confusion matrix, recall , accuracy, precision, f1-score, and ROC area the performance results were assessed [8].

Here the Word2vec model which includes CBOW and Skip-gram, was used to determine the representation of words. Both the models consist of input, output, and projection layers. CBOW predicts target words based on the context of the text, while on the other hand, based on the target word Skip-gram predicts the context. TF-IDF makes sure whether a word contains sentiment information is deduced by comparison with sentiment dictionaries. In this experiment, NTUSD, Hownet sentiment dictionary, and NTUSD including Li Jun's Chinese commendatory term and derogatory term Dictionary were applied for Chinese sentiment analysis. To overcome the shortcomings in context understanding and Information loss at sequence ends, BiLSTM was proposed in this paper. Hyperparameters such as alpha value, maxLen, duodeNum, etc, were applied for the best classification effect. To figure out the credibility of the word representation, different word representations are fed into the BiLSTM model, which includes vec, TF-IDF, Seninfo, and Seninfo+TF-IDF. Mean precision, recall, and F1 scores of ten repeated experiments were recorded with different word representations. To further solidify the advantage of the proposed method, it is then compared with traditional models [14].

The paper explores the application of deep learning techniques for emotion analysis in Bangla, unlike previous research, they targeted six emotions: joy, depression, fear, anger, love, and surprise. The research draws inspiration from previous studies using lexicon approaches and classical methods and uses BiGRU and CNN-BiLSTM for emotional classification. The dataset was created using Google Translate by converting English text to Bangla, and 7214 sentences were chosen for experimentation. The initial steps involved converting categorical labels into a readable numeric format, checking missing values, and visualizing the minimum, maximum, and average length of texts. To prevent overfitting, the dataset was split into training and testing sets in an 80:20 ratio. During the experimentation, overfitting occurred due to increased epochs, which was resolved by fine-tuning parameters. In their experiment, CNN-BiLSTM performed with an accuracy of 66.62% vs. 64.96%, slightly better than BiGRU. Upon comparisons with Bangla texts that are not included in the dataset, both models performed equally well, with CNN-BiLSTM slightly better than BiGRU. The primary challenge of the research was the lack of a Bangla text dataset, leading to translation errors. The researcher suggests that a larger dataset could significantly increase the model's accuracy and that in the future, focus on the syntactic and semantic features of Bangla text. [12].

The paper presents a study on sentence-level sentiment analysis and opinion mining using a product review dataset from Amazon's website. Six types of product reviews for cameras, laptops, tablets, TVs, cellphones, and video surveillance are included in the dataset. To categorize the reviews, the researchers applied machine learning methods such as Support Vector Machine and Naïve Bayes. The goal was to analyze the opinionated data and extract insights that could help users make decisions and improve business strategies. The researchers gathered 13,057 review datasets in JSON format and preprocessed them using tokenization, stop word removal, stem-

ming, and punctuation mark removal. Sentiment ratings were assigned to phrases using sentiment lexicons containing both positive and negative terms (4783 negative words and 2006 positive words). With an accuracy of 97.17% in tables and 98.71% in cameras, Naïve Bayes beat Support Vector Machine in every area. They come to the conclusion that sentiment analysis can benefit from machine learning techniques, which can also be used to obtain insights from product reviews. Future work could include aspect-level sentiment analysis to understand people's preferences, such as camera quality, megapixels, picture size, structure, lens, and picture quality. The research paper contributes to the field of sentiment analysis using product reviews and highlights potential areas for further exploration. [10].

This paper's main goal is to create a supervised learning model to categorize a large number of reviews and to classify positive and negative user feedback. According to an Amazon study, more than 88% of online buyers place just as much trust in reviews as they do in personal recommendations. Negative reviews have the potential to negatively impact sales, but positive evaluations can make a significant statement about the genuineness of an item. It is critical in business to understand feedback from customers and act appropriately based on thorough data to make well-informed judgments. Support vector machines, multinomial Naïve Bayesian, and Python and R programming languages were the primary classifiers utilized by the previous researchers. One of the previous projects used TF-IDF as an additional experiment, and it was successfully able to predict ratings using a bag of words. However, only a few classifiers were used. Therefore, in this paper, the researchers tried to make their work more efficient by selecting well-performing models and ideas and utilizing them together, which ultimately led them to use Bag of Word, TF-IDF, and Chi-square. About 48500 product reviews from three categories—musical instruments, cell phones and accessories, and electronics—were examined in this study. The experiment used several machine learning algorithms, such as Random Forest, Decision Tree, Naïve Bayes, Linear Support Vector Machine, and Stochastic Gradient Descent. Linear Support Vector Machine classifiers performed best in all three categories, with an accuracy of 93.57% in cell phones and accessories, 94.02% in musicals, and 93.52% in electronics. With the F1 measure, the suggested supervised learning model produced accuracy levels above 90% as well as precision and recall levels above 90%. Cross-validation, various feature extraction techniques, and training-testing ratios were used to test the model. The model's accuracy improved 10-fold, and the Support Vector Machine was chosen for the best classifying results. Future work includes applying Principal Component Analysis in active learning, incorporating the model into programs for customer interaction, and generalizing the model to all types of text-based reviews and comments. [7].

In this paper, the Bert model is used as an input layer feature extraction at the pre-processing stage. Using bidirectional long short-term memory and gated recurrent neural units the hidden layers can contain long-term dependencies, which are now present regardless of the dimension and the frequency in the text. By softmax, the sentiment polarity is generated by pooling to a smaller weight concerning the attention mechanism. In the experiment the BERT model converts the reviews into a numerical matrix, the matrix is then given as input of BERT and trained by two training strategies MLM and next-sentence prediction. The sentence segmentation

method is applied during training, for splitting a long sentence into numerous short text blocks. These are word sequences that are efficient in mapping short sentences into corresponding dimensions. Transformer then receives input from sentence pairs and learns to predict the second sentence in the pair. This paper proposed BiGRU which splits regular GRU neurons into forward and backward states. The connection of two hidden layers with opposite transmission into the same output layer. Enabling the model to obtain information from both the past and the future for more accurate analysis of product reviews. Softmax deduces positive, negative, and neutral scores at the output layer by fusion of different semantics of BERT-BiGRU models. In the end, the attention mechanism derives the linear weight sum of all polarities of sentence sequences. For adequate evaluation of the sentiment analysis efficiency of the proposed model, experiments were conducted on multiple datasets with various domains. The first dataset is from IMDB and the second is from ChnSentiCorp, a Chinese emotional corpus. In the end, the experiment results indicate that models like Weight W2V-ATT-LSTM, Multi-Bi-LSTM, and Bert-BiGRU-Softmax are appropriate for sentiment analysis, in which Bert-BiGRU-Softmax has better performance. In the second experiment, the paper analyzed a large dataset of 500 thousand E-commerce reviews with 150 predefined dimensions, mainly consisting of smartphone reviews from various brands. Covering all aspects of products like polarities and dimensions like "quantity", "services" etc. The paper compared the experiment results of several classification models including RNN, BiGRU, BiLSTM, LSTM, GRU, Bert-BiLSTM, and Bert-BiGRU-Softmax based on loss value and accuracy. The test was taken over about half a million original data from the web reviews corpus and split into training and test datasets in experiments. In the end, the comparison of all the models tested, the Bert-BiGRU-Softmax model at the 8th epoch exhibited the largest value of accuracy of 0.955 [11].

Chapter 3

Work Plan

The required data for our thesis was collected from a website called Daraz by web scraping. By implementing several data preprocessing strategies, the final clean data was created that has a total of 34,800 reviews. The final preprocessed dataset has two features: a "Clean Sentence" that contains the Bengali reviews, and a "Sentiment" column that contains the sentiments. The final dataset was divided into two sets: one was for training and validation, and the other was for the test dataset. The reason for having one dataset for both train data and validation is to apply k-fold cross validation. K fold was used for the BiLSTM model for our thesis. However, the BERT models and the traditional machine learning models did not require k-fold cross-validation, so the dataset for train and validation data was further divided into two sets: train and validation. All of these divided datasets were stored in separate Excel files. At first, for training the traditional machine learning models: Multinomial Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine, and Gradient Boosting Classifier; oversampling was done to the train dataset and to the validation dataset. Then this oversampled dataset's reviews were tokenized and later converted to some numeric representation using TF-IDF Vectorizer. After that, these vectorized reviews along with the sentiments were fed to the models for training and evaluating training performance using different hyperparameter values. If for a certain hyperparameter value the algorithms performed the best, then that value was considered as the best parameter value for training the model for our dataset. Using that value, the code was run on the test dataset again for final evaluation. Before using the deep learning models, GloVe and FastText word embeddings were finetuned on our domain or Bengali dataset. Additionally, both BERT models—Multilingual BERT and BanglaBERT—were also finetuned. So for the deep learning models, we implemented two versions of the models, one without fine tuning on our domain and the other one after fine tuning on our domain. Both were used in the same way. Deep learning models were made to capture complex contextual meaning, so they did not require any oversampling. For BiLSTM models, at first reviews were tokenized and sentiments were converted to integers. Positive: 2, Neutral: 1, Negative: 0. Then GloVe embedding was applied to these tokenized words for reviews. After that, they were sent to the BILSTM model for training. For the BiLSTM model's training, the dataset that contained both training and validation data was used, and k-fold cross-validation was applied to them. For each epoch in a fold, train data was used for training, and validation data was used for evaluating training performance. At last, the test dataset was

used for evaluating the final performance of the model. BiLSTM with FastText word embedding was used in the same way, instead of GloVe FastText. The BERT models (Multilingual and BanglaBERT) did not require any external word embeddings, as they learned word embedding implicitly during training. The reviews were tokenized using BERT's built-in tokenization, and sentiments were converted to integers. Then these data were sent to the model for training using the train dataset, and the validation dataset was used to evaluate the model's training performance for each epoch. Finally, using the test dataset, the final performance of the model was evaluated.

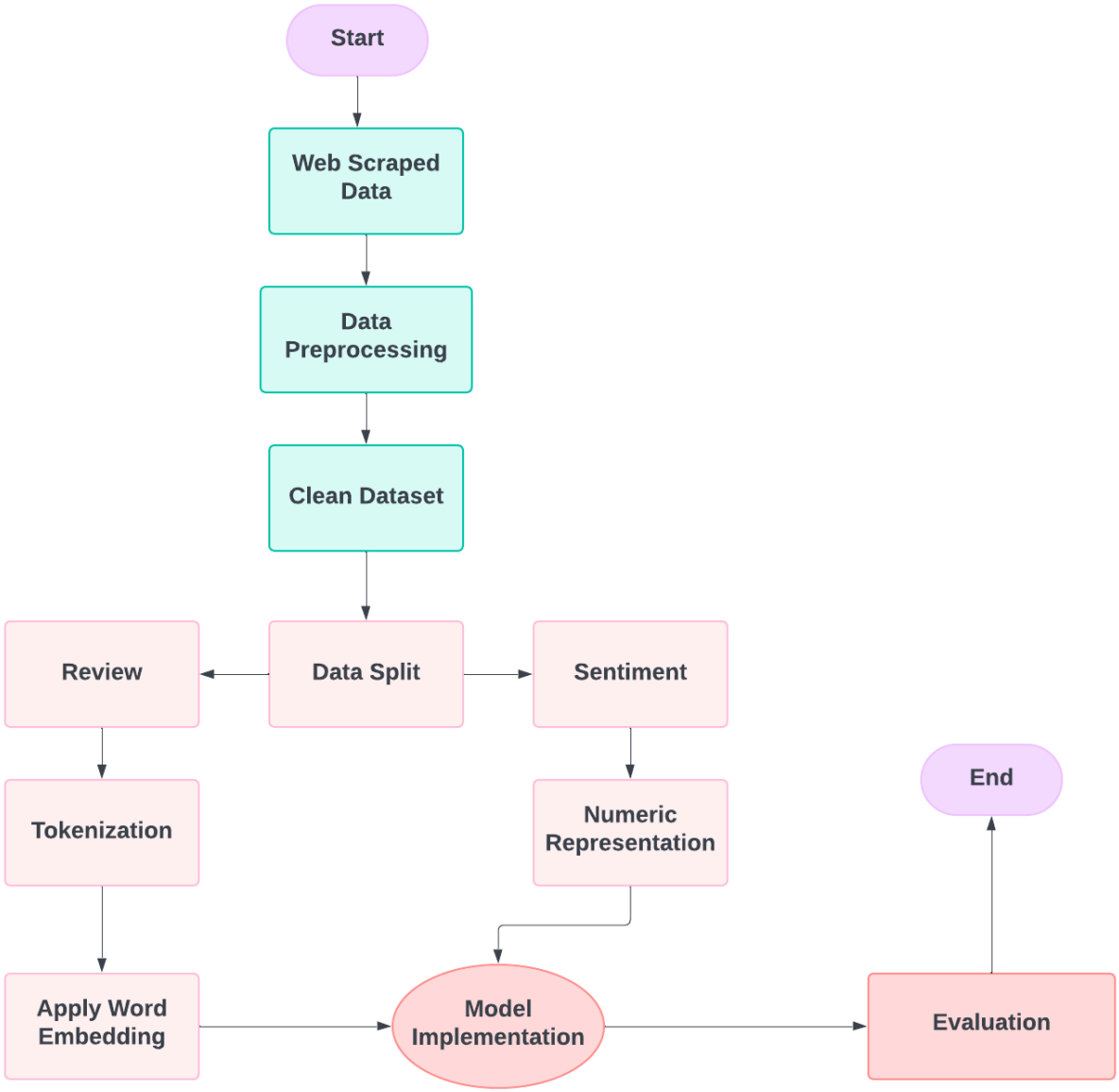

Figure 3.1: Workflow

Chapter 4

Model Analysis

In-total 5 models were implemented. Random Forest, Multinomial Naive Bayes,Logistic Regression,Support Vector Machine (SVM) and Gradient Boosting Classifier. In this chapter there will be a brief discussion of all these models.

4.1 Random Forest

Random forest is based on an ensemble machine learning model that significantly improves a single decision tree by reducing overfitting and thus improving performance. It does this by combining multiple decision trees, and is a more accurate and reliable predictive model. Since our research focuses on classifying sentiment, implementation of random forest may be a suitable option for our application.

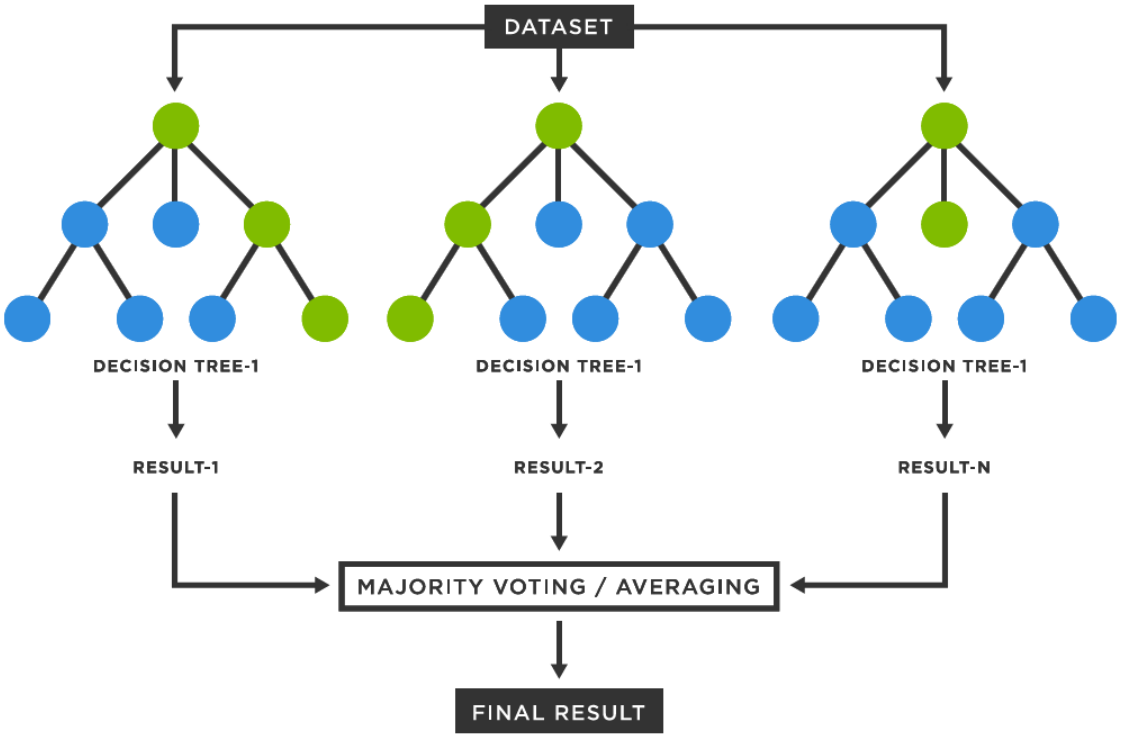

Figure 4.1: Random Forest

$$\mathrm{RF}(x) = \frac{1}{N} \sum_{i=1}^{N} T_i(x)$$

- $\mathrm{RF}(x)$ is the final result for the instance x
- N is the number of decision trees
- $T_i(x)$ is the prediction for the i-th tree for the input x

Advantages:

- Has the ability to handle large datasets with high dimensions.
- Decreases overfitting by taking the average of several decision trees.

Limitations:

- Compared to a single model, it may be more costly to compute and take a longer period of time to predict the result.

4.2 Multinomial Naive Bayes

Multinomial Naïve Bayes is a probabilistic learning method that is based on the Bayes theorem. It can predict the tag of a text or phase by using Bayes theorem. This tag is given after calculating the probability of each tag possible and sets the tag with highest probability. With the help of this strong model, text can be categorised into several classes. As our research involves classifying review into multiple sentiment classes, using Multinomial Naive Bayes can be used efficiently.

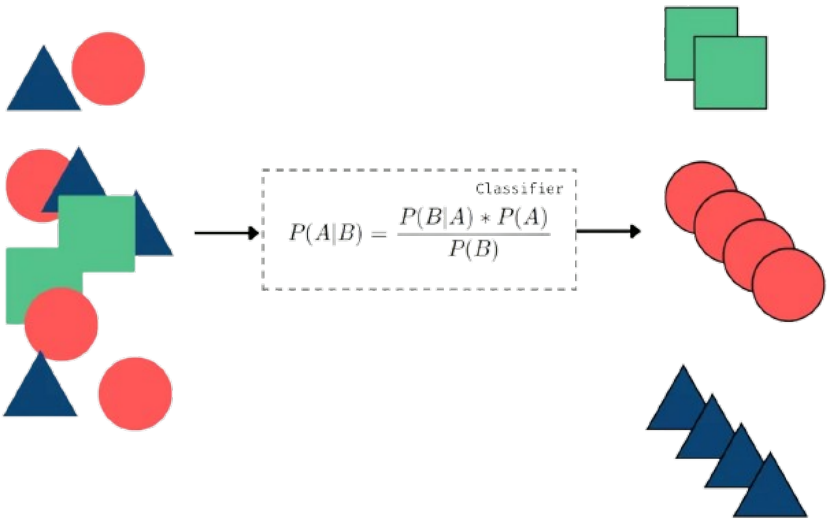

Figure 4.2: Multinomial Naive Bayes

$$P(c|x) \propto P(c) \prod_{i=1}^{n} P(x_i|c)$$

- $P(c)$ is the prior probability of class c
- $P(c|x)$ is the posterior probability of class c given the feature vector x
- x is the feature vector $(x_1, x_2, \ldots, x_n)$
- $P(x_i|c)$ is the likelihood of feature x_i given class c

Advantages:

- Fast and efficient for large datasets.
- Performs well with textual data, especially when features represent word frequencies.

Limitations:

- Assumes feature independence, which often is not the case in real-world data.

4.3 Logistic Regression

Logistic regression is a supervised machine learning algorithm. It classifies an activity by estimating the probability of a certain result or outcome. By analyzing the correlations between one or more independent input variables, logistic regression classifies the data into distinct groups. It gives information about the significance and impact of each feature on the prediction.

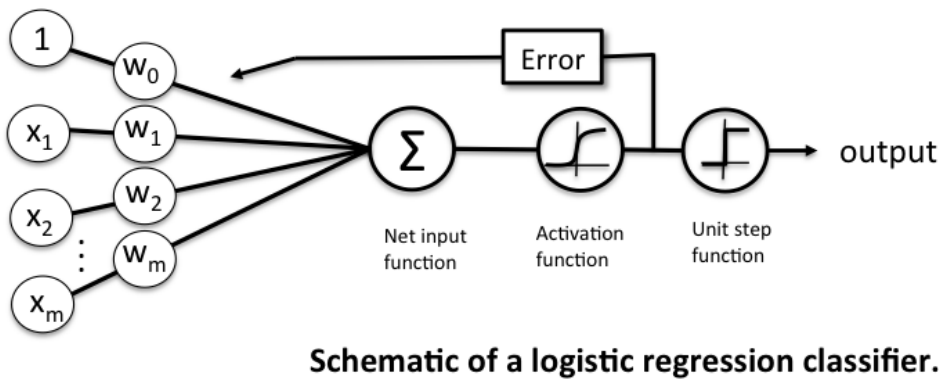

Figure 4.3: Logistic Regression

$$P(y = 1|x) = \sigma(wx + b)$$

- $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function,
- w is the weight vector,
- b is the bias term.

Advantages:

- Can tell which features are most important, with coefficients.
- Effective for large datasets and functions best when the classes are linearly divided.

Limitations:

- Can struggle with non-linear relationships between features and class labels.

4.4 Support Vector Machine (SVM)

Support vector machine is a supervised learning algorithm which is widely used for classification and regression tasks. But it's mostly applied to machine learning classification problems. SVM is a strong and adaptable algorithm that works particularly well with complicated classification issues and high-dimensional data. For its ability to create complicated decision boundaries using support vectors and efficiency in both linear and nonlinear classification scenarios we find it to be a suitable model for our research.

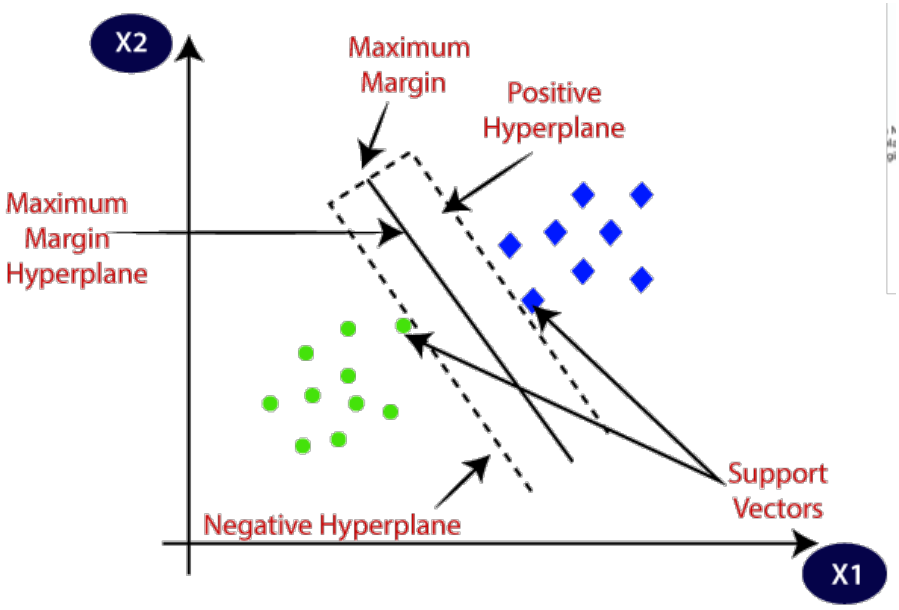

Figure 4.4: Support Vector Machine

$$\min_{w,b} \frac{1}{2}||w||^2 \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1$$

- y_i are the class labels,
- w is the weight vector,
- x_i is the feature vector,
- b is the bias term.

Advantages:

- Effective in high-dimensional spaces and when the number of dimensions exceeds the number of samples.
- Adaptable to various kernel functions for non-linear classification.

Limitations:

- Memory-intensive and costly to compute especially with large datasets.

4.5 Gradient Boosting Classifier

Gradient boosting is a strong boosting technique that trains each new model to reduce the loss function of the previous model by gradient descent. It does this by combining a number of weak classifiers into a strong classifier. At each iteration, the algorithm calculates the gradient of the loss function with respect to the current predictions and generates a new weak model to minimize the gradient of the loss function. The predictions of the new model are then added to the ensemble, and the process runs on until the release requirement is satisfied. As gradient boosting provides great prediction speed and great accuracy in large complex datasets, therefore it would be useful for our research.

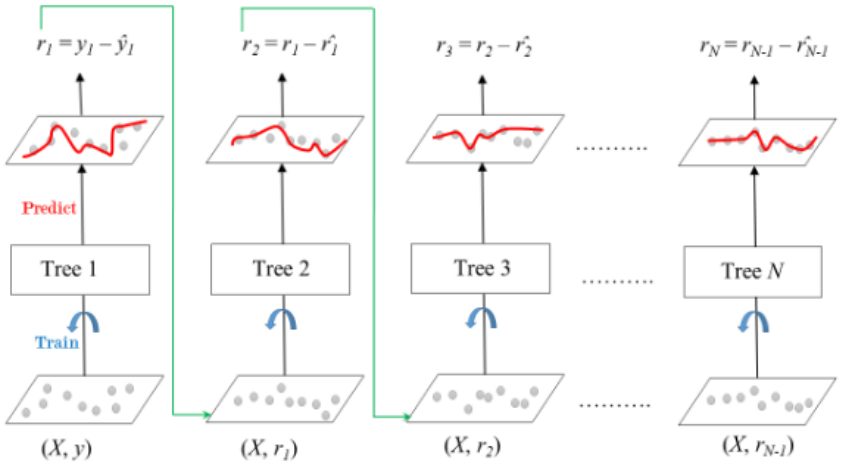

Figure 4.5: Gradient Boosting Classifier

$$F_m(x) = F_{m-1}(x) + h_m(x)$$

- F_m is the ensemble model after m stages,
- $h_m(x)$ is the new model fitted to the residuals of the current ensemble.

Advantages:

- High accuracy and can handle a variety of data types and distributions.
- Can capture complex patterns and correlations in data.

Limitations:

- Prone to overfitting if not properly tuned.
- Computationally expensive and requires careful parameter tuning.

4.6 Bidirectional long short-term memory (BiLSTM)

Bidirectional LSTM (BiLSTM) is an extension of the standard Long Short-Term Memory (LSTM) designed for sequential data processing, particularly in natural language processing (NLP) tasks. In BiLSTM, there are two LSTM layers—one processes the input sequence in the forward direction, while the other processes it in reverse. Each layer uses the output from the previous time step as input for the next time step, allowing the model to capture both past and future context simultaneously. This bidirectional structure helps the model better understand word relationships within a sentence, improving performance on tasks like sentiment analysis and text classification.

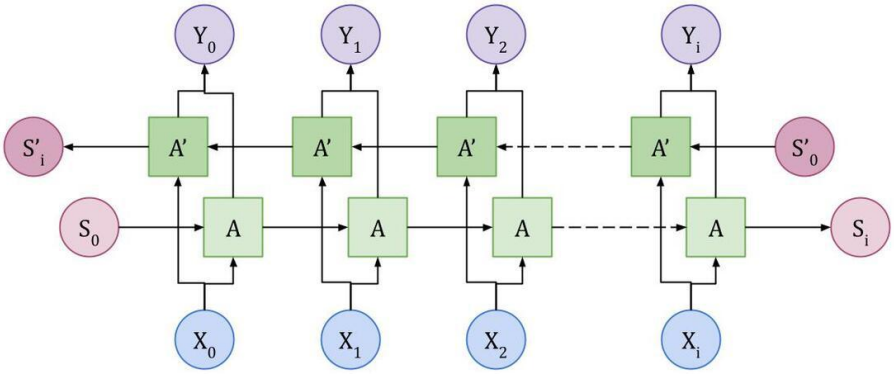

Figure 4.6

Advantages:

- BiLSTM can handle different input-output lengths which makes it ideal for applications such as machine translation, where the input and output sequences may vary in size, or text summarization, where the input text is longer and the summary is shorter.
- Additionally, BiLSTM captures information from both past and future contexts by processing sequences in both forward and backward directions, which improves performance in tasks that benefit from contextual understanding.

Limitations:

- Since BiLSTM requires two sets of LSTM cells, one for the forward pass and one for the backward pass, it effectively doubles the computational cost compared to a standard LSTM, making it more resource-intensive and slower to train.
- Moreover, while BiLSTM excels in many NLP tasks, it may not be the best fit for tasks like speech recognition, where other models, such as convolutional or attention-based architectures, might outperform it due to their ability to capture more localized and hierarchical features.

4.7 Bidirectional Encoder Representations from Transformers (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is a deep-learning framework based on transformer architecture, designed for natural language processing tasks. BERT pre-trains on large datasets using two techniques: Masked Language Modeling (MLM), where random words are masked, and Next Sentence Prediction (NSP), where sentence relationships are predicted. Unlike previous models, BERT is bidirectional, understanding the context of a word by analyzing both preceding and following words. Fine-tuning allows BERT to be adapted for specific tasks. It excels in language understanding due to its transformer-based attention mechanism, enabling it to grasp complex sentence structures and context.

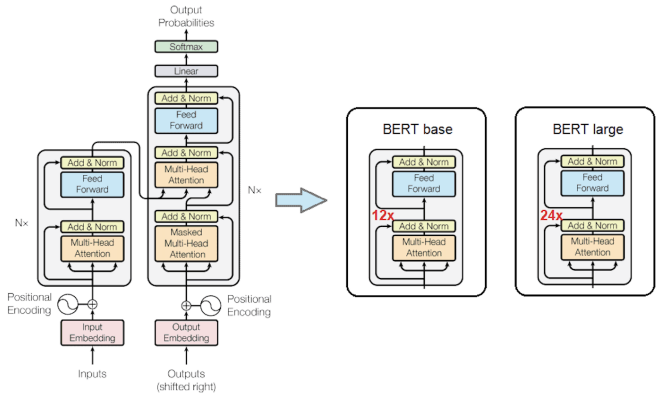

Figure 4.7

Advantages:

- BERT comes pre-trained in multiple languages, making it an excellent choice for projects involving non-English text.
- It is easy to use, as it can be fine-tuned for specific tasks with minimal adjustments.

Limitations:

- Due to its large number of parameters and training structure, BERT is computationally expensive and requires substantial resources, especially for training on large datasets.
- Its large model size also leads to slower training times, as the significant number of weights in the network must be updated during training, making it less efficient for quick iterations or resource-constrained environments.

4.7.1 Multilingual BERT

BERT (Bidirectional Encoder Representations from Transformers) is a powerful natural language processing (NLP) model that revolutionized the field with its bidirectional transformer-based architecture. Unlike earlier models that processed text sequentially, BERT leverages transformers' self-attention mechanism to understand a word in the context of both its preceding and following words. This bidirectional approach enables BERT to capture deeper semantic meaning and dependencies within text, making it highly effective for a wide variety of NLP tasks. BERT's architecture consists of multiple layers of transformers, where each layer processes the input text by generating word embeddings based on context. Its pre-training strategy uses two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM involves randomly masking words within a sentence and requiring the model to predict those masked words based on the surrounding context, teaching it to understand language patterns. NSP helps the model understand relationships between sentences by predicting whether two sentences are consecutive in a given context. BERT's pre-training on massive corpora, combined with its fine-tuning on specific downstream tasks like question answering, sentiment analysis, and named entity recognition, allows it to excel in diverse language tasks. This architecture's strength lies in its attention mechanism, enabling it to focus on important parts of the input text and grasp intricate relationships within and across sentences [4].

4.7.2 BanglaBERT

BanglaBERT is a specialized adaptation of the BERT architecture tailored for Bangla, a low-resource language with relatively less data available for pre-training compared to widely spoken languages like English. The architecture of BanglaBERT closely follows the transformer-based design of BERT, with layers of self-attention mechanisms that process the input text in a bidirectional manner. However, BanglaBERT is pre-trained specifically on Bangla text corpora, allowing it to understand the unique syntactic and semantic structures of the Bangla language. The model is trained using the same MLM and NSP objectives as BERT, but with a focus on

capturing the linguistic nuances specific to Bangla. BanglaBERT is designed to address the challenges of low-resource languages by being efficient in both compute and memory usage while outperforming larger multilingual models like mBERT and XLM-R in Bangla-specific tasks. The research behind BanglaBERT introduces the BLUB benchmark, which evaluates the model on tasks such as sentiment classification (SC), natural language inference (NLI), named entity recognition (NER), and question answering (QA). In these tasks, BanglaBERT demonstrates superior performance, particularly in resource-constrained environments where fewer labeled training samples are available. This model's architecture and pre-training allow it to achieve state-of-the-art results in Bangla NLU, making it a critical tool for advancing Bangla language processing [2].

Chapter 5

Description of Data

5.1 Data Collection Method

For data collection, initially the Daraz Website is visited where manually every product's ratings' HTML files are saved as a Webpage Complete option. Each product page allows to show each star rating, which means there are buttons that show 5 star, 4 star, 3 star, 2 star, and 1 star ratings that can be selected, and if not selected, then all the ratings are shown sorted by relevancy. After navigating through

the product's page, 5 star is selected first, and then scrolled down to a scroll-down menu that says "3/page" and from the menu, the "100/page" option is selected. This is very important as "3/page" indicates that each HTML file would have a maximum of 3 total reviews, whereas "100/page" indicates that a maximum of 100 total reviews can be collected from a single HTML page only if available. Then the HTML file is saved as Webpage Complete option and renamed by concatenating the default file's name with ".5", indicating that the HTML saved file contains the 5-star rating reviews only. This naming format is used because, with four group members collecting these HTML files, duplication of these files can be possible. To avoid this duplication problem, this naming convention is performed so that when adding these HTML files in a folder altogether from all the members, repeated files are skipped and not added to the folder directory. Next, ratings with the 4 star button

are pressed, and since the "100/page" option was already selected beforehand, there was no need to select it again. Then the 4-star rating HTML file is saved similarly to the way the 5-star rating page was saved, following the naming format where in this case ".4" was concatenated after the file's default name. The 3-star, 2-star, and 1-star ratings HTML files are saved just like the 4-star rating. After all the 5-star ratings HTML files are saved and collected from the current product, a different product is navigated next, and all the similar procedures are applied to the next product, and so on. This is how all the HTML files are gathered in a single folder and directory. After gathering all the HTML files, a Python script is prepared to make

an xlsx format file containing the data for the research. First, the folder containing all the HTML files is uploaded to Google Drive, and then a Google Colab ipynb format file is created, which contains a Python script block. This Python script block had firstly Google Colab library that imported Drive to mount and access

the HTML files stored in the uploaded folder. Dataframe libraries like pandas are used for data manipulation and creating data structures. For handling file paths and retrieving the list of HTML files stored in the mount folder, libraries like os and glob are used. Next, each HTML file is parsed using BeautifulSoup from the bs4

package library to extract relevant information such as product name, store, price, user names, reviews, and ratings from each HTML file by using the specific HTML tags such as "span", "div", and "a" and classes to locate and extract the data. These HTML tags and classes were checked and collected from the Daraz product page by the inspect option and finding the specific HTML tags and classes for each piece of information. In Daraz, some reviews are empty, meaning a customer can give only the ratings and not both the reviews and ratings. In such cases, the empty or null value review rows are excluded from the Dataframe. Furthermore, in the case of retrieving ratings, individual stars are counted. For example, 5-star ratings will have 5 "img" HTML tags that each show a star on the frontend page. Therefore, as there are 5 stars shown on the page, the rating is a 5. Similarly, a 1-star rating will have 1 "img" HTML tag, so the frontend page will show 1 star. Next, from the ratings,

sentiment is created where ratings of 4 and 5 are "Positive" sentiment, ratings of 3 are "Neutral" sentiment, and ratings of 1 and 2 are "Negative" sentiment. Finally, any duplicate rows found are removed and saved in the Dataframe, which is then saved as an xlsx file. Here, an xlsx file is used instead of csv because an xlsx file shows Bangla fonts clearly on the local machine, but a csv file distorts Bangla fonts, which does not allow reading the reviews on the local machine.

5.2 Data Preprocessing

For the analysis data were collected primarily from online e-commerce site DARAZ. These data were extracted from the site through scraping. HTML pages of the product page were manually downloaded, which was then run through scraper to extract all the reviews from the product which is then transferred onto the spreadsheet. The dataset initially comprised shop name, product name, customer name, product price, rating, and the review itself. All these data have been merged and inserted into one dataframe

48000 reviews are collected upon which 22000 are Bengali 15000 are English while the rest of 9000 were in "Banglish" (Romanized Bengali). From the data-set sentiment has been determined by the value of the rating. Three sentiments have been classified which are positive, negative and neutral and were stored in a separate column of the dataframe. Rating with less than three was determined as negative, greater than that were positive and reviews with exact value of three were neutral.

After sentiment has been classified, from the data frame all the rows containing null values have been removed. Also the data has gone through the cleaning process. These are as follows.

- Any type of special characters have been removed which do not contain any data beneficial to analysis.
- Reviews containing emojis are also removed.
- URLs and day-time formatted data have also been removed from the dataset.
- Punctuation marks and alphanumeric characters have been removed. These sorts of data create noise and have no impact on the analysis.
- Non-word characters have been removed, such as hashtags and punctuation marks.

The dataframe initially contained a mixture of reviews written in English and Bengali. These were then filtered to separate dataframes, English, Bengali and Romanized Bengali. In our analysis Bengali and English Reviews are taken into consideration. Due to the inconsistent grammatical accuracy of reviews written in Romanized Bengali, the particular datset was excluded from the analysis.
The Bengali data-set was again further cleaned to remove any English words which had no relevance to the analysis. Also removing any unwanted special Bengali charcters. The English data-set on the other hand was translated to Bengali and then examined for having any grammatical issues. In the end 34800 reviews were used in the analysis, out of which 11609 were negative, 7442 being neutral and 15749 were positive

Pie chart: A Pie chart can easily represent the proportion of sentiment in data. Offering a clear overall concept of sentiment in the mixture of data.

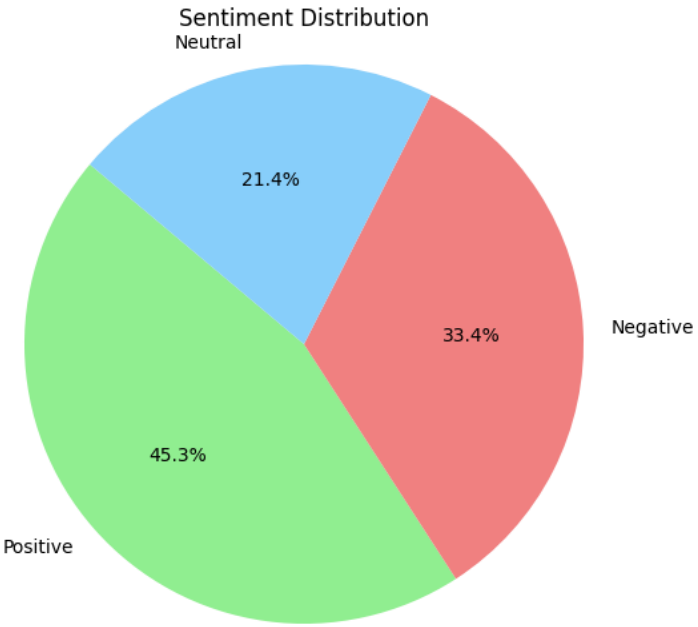

Figure 5.1: Representation of proportion of each sentiment in Bengali data-set

Histogram: This histogram shows the graphical representation of the distribution of ratings in the dataset. Helping us understand the pattern of distribution and the frequency of various rating values.

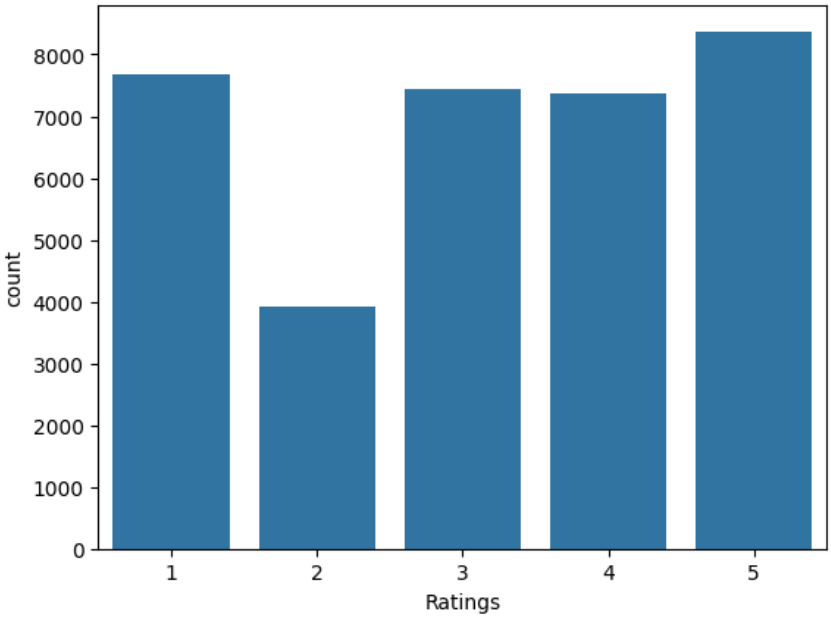

Figure 5.2: Representation of proportion of each sentiment in Bengali data-set

Outliers can significantly alter the observation in a dataset, resulting in errors due to variability in the data. Causing a notable impact on statistical analyses and can alter the results of models. Using Z-Score method, the outliers in the data were removed. The word count distribution shows whether the dataset contain predominantly short texts or longer texts. This helps in understanding the nature of the dataset. Initially, we faced some problems regarding anomalies, especially in reviews with shorter word count. So all the reviews with word count of less than 5 were eliminated to mitigate the issue. Outliers in word count distributions can highlight valuable information about the structure and characteristics of the data.

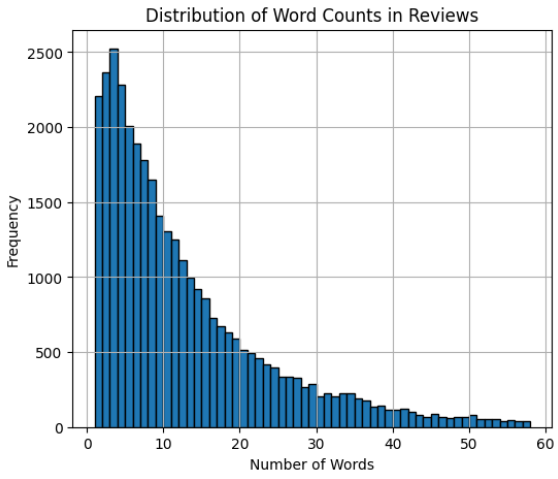

Figure 5.3: Positive Reviews

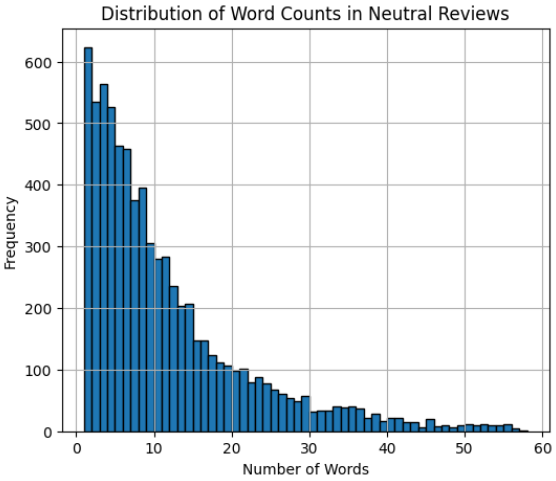

Figure 5.4: Neutral Reviews

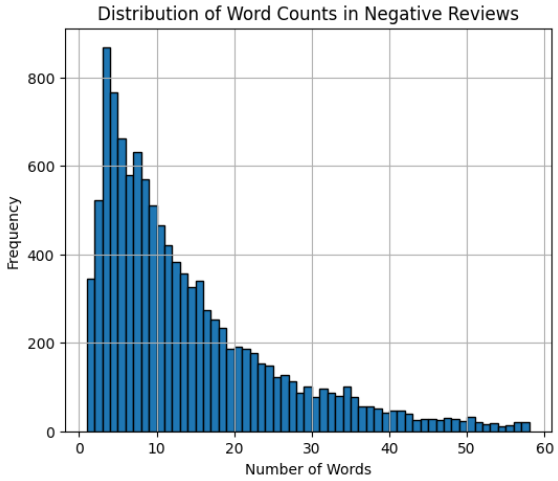

Figure 5.5: Negatuve Reviews

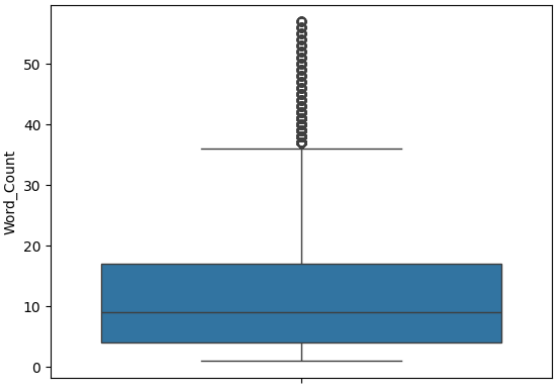

Figure 5.6: Outlier of word count

Word Cloud will help identify the main gist of positive, negative and neutral sentiment. By showcasing the most frequent words within reviews of each sentiment, common theme can be properly highlighted.

Figure 5.7: Neutral Word Cloud

Figure 5.8: Positive Word Cloud

Figure 5.9: Negative Word Cloud

Since the labeling of the sentiments was done based on the ratings, there were some issues. For instance, there are some reviews that seem positive, but the rating is given 1 or 2 by the customer. As a result, the sentiment of that review became "negative". The same thing happened with the other sentiments. To solve this problem, we manually checked the dataset, and if a mislabeling was found, we fixed it. However, since the dataset has more than thirty thousand data points, it was not possible to go through the whole dataset within a limited timeframe.

	clean_sentence	Ratings	Sentiment
0	তেমন ভালো না কিন্তু চলার মত আছে কিন্তু এই বাজেটে আরো ভালো কিছু আশা করি আপনারা সবাই নিতে পারেন আমার কাছে ভালো মনে হয়নি আরেকটু ভালো হলে সুন্দর হত	4	Positive
1	পন্যটা মোটামুটি বেশ ভালো	2	Negative
2	প্রোডাকটি ভালো নয় চার্জ একেবারে কম যায় ব্যাটারি ব্যাকআপ খুব কম আর মাথাগুলো মোটা হওয়ায় দাড়িগুলো একেবারে নিখুঁতভাবে কাটা যায় না	5	Positive
3	পোডাক্ট মোটামুটি ভালোই বলা চলে	1	Negative
4	খুবি ভাল মেশিন ব্যবহার করার পর রিভিউ দিলাম	2	Negative

Figure 5.10: Sentiment mislabeling due to ratings

5.3 Data Splits

The final preprocessed dataset has a length of 34,800. This dataset was divided into two sets. One set is for training and validation since we have performed k fold cross validation on BiLSTM algorithm and the other dataset is a test dataset which was used for final evaluation. The length of the dataset that is for training and validation is 27,840. Furthermore, the test dataset has a length of 6,960. These two datasets were stored in separate excel files since we wanted consistency every time we run our code. Additionally, we had to separate the dataset that was meant for training and validation into two parts as we did not use k fold cross validation for BERT models and traditional machine learning models. That dataset was splitted into two sets one was meant for training which has the size of 20,601, and the other was for validation which has the length of 7,239. We have also stored these two datasets into two separate excel files.

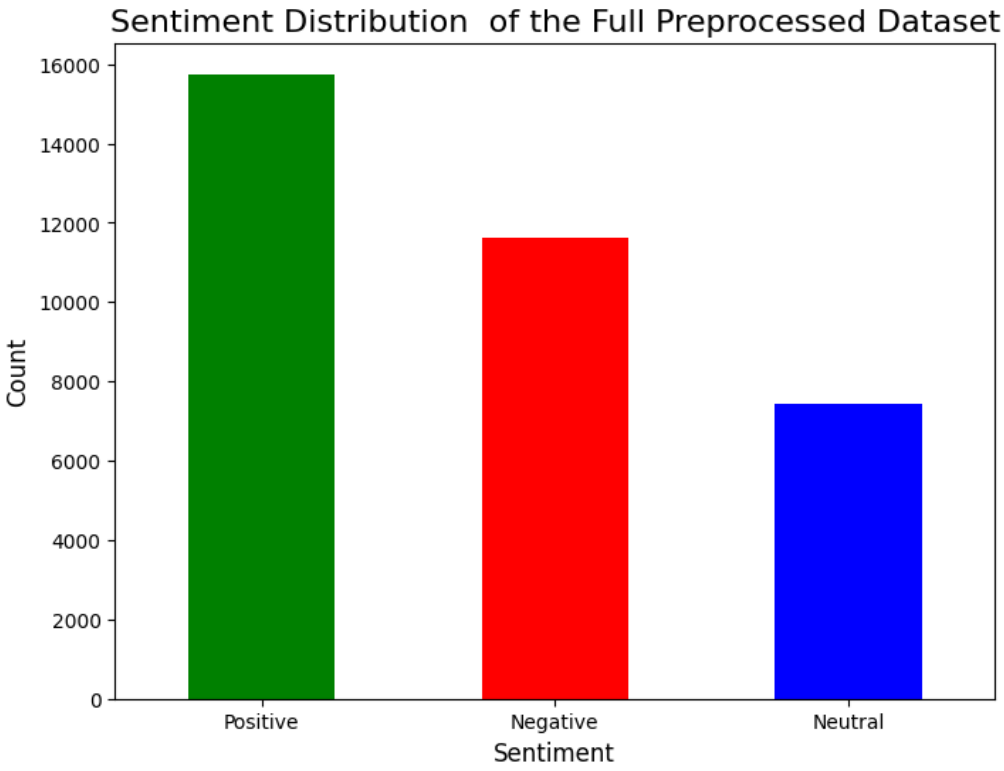

Figure 5.11: Sentiment Distribution of the preprocessed dataset

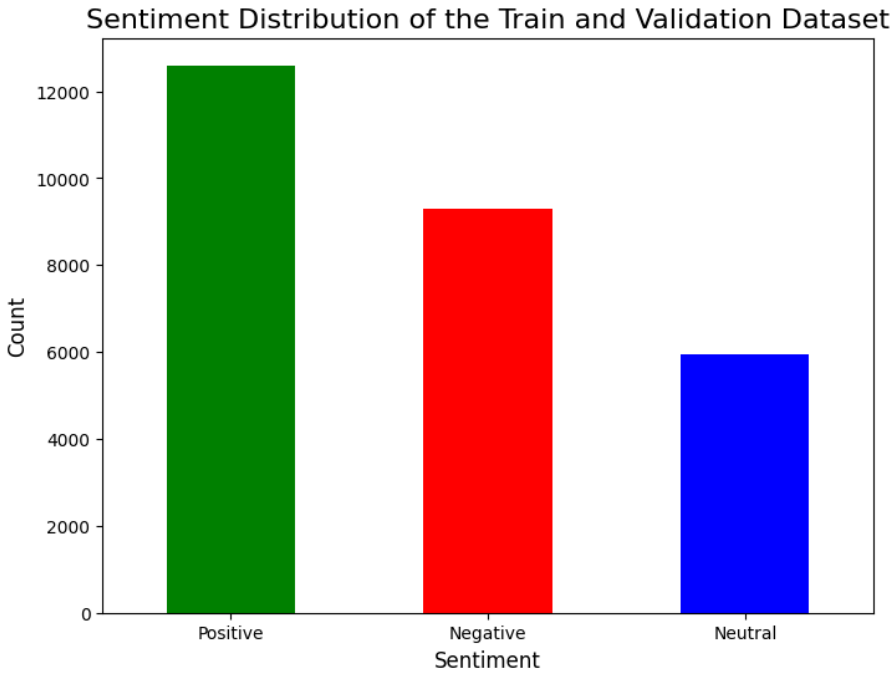

Figure 5.12: Sentiment Distribution of the train and validation dataset

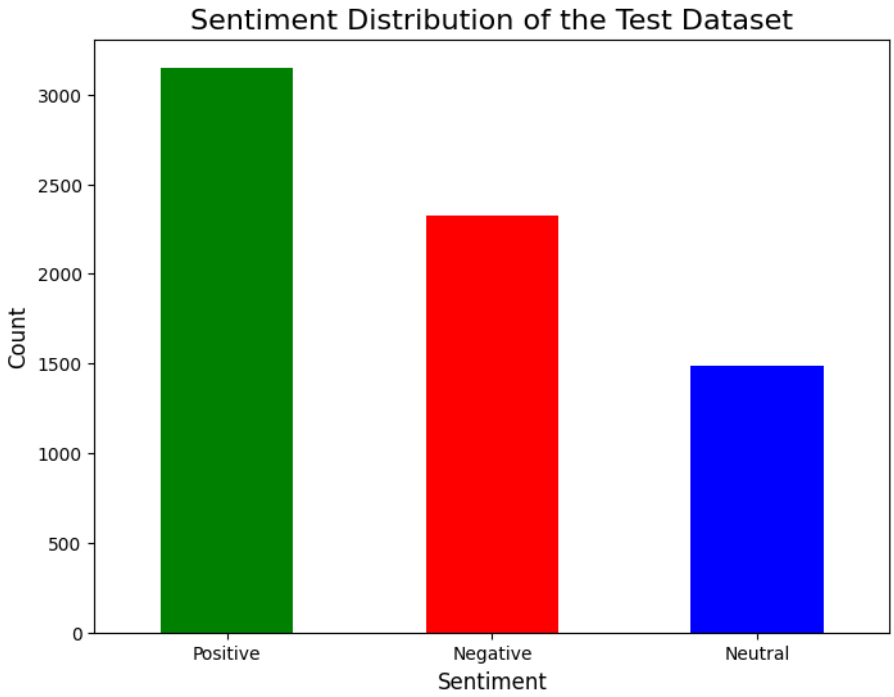

Figure 5.13: Sentiment Distribution of the test dataset

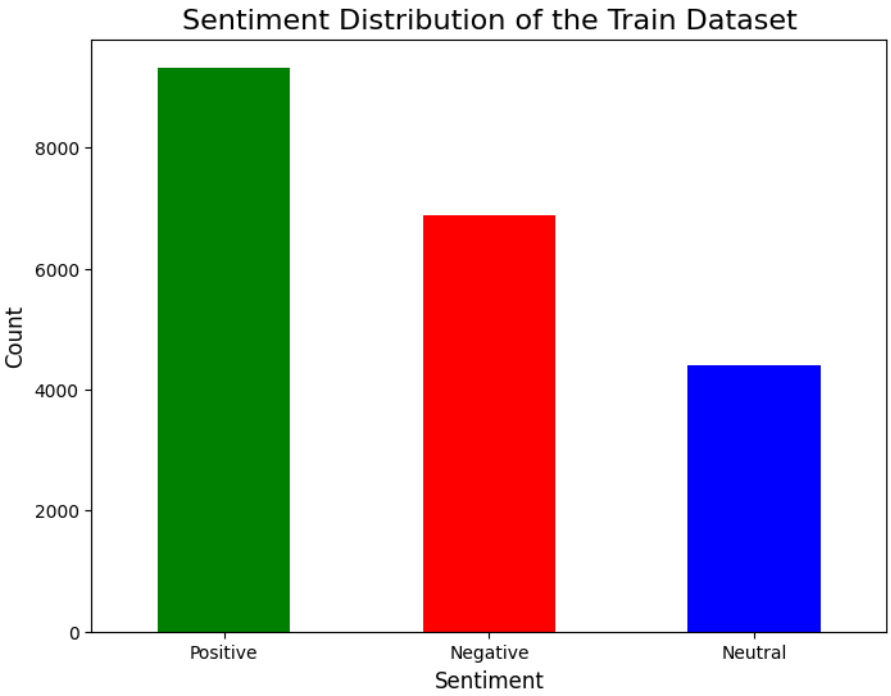

Figure 5.14: Sentiment Distribution of the train dataset

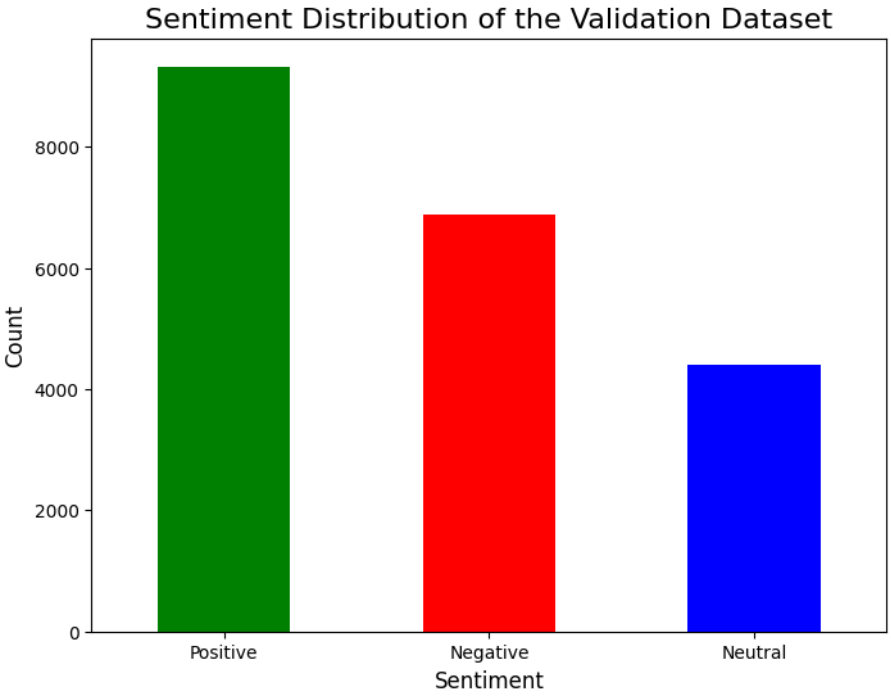

Figure 5.15: Sentiment Distribution of the validation dataset

Chapter 6

Result Analysis

6.1 Performance Metrics

We have used four metrics to evaluate the performance and make comparisons between different machine learning algorithms. These metrics are Accuracy, F1 Score, Precision, and Recall.

Accuracy: Out of all the examples, accuracy represents the overall number of accurately predicted cases.

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

Precision: Precision defines the proportion of true positive predictions among all the positive predictions.

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

Recall: Recall defines the proportion of true positive predictions among all the actual positives in the dataset.

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

F1-Score: It includes both precision and recall by calculating their harmonic mean.

$$\text{F1-Score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

6.2 Performance Evaluation: Machine Learning Approach

6.2.1 Validation Set

Multinomial Naive Bayes: For the Naive Bayes algorithm, alpha was selected as the hyperparameter, and different values were assigned to it. Based on the accuracy metric, we chose the best alpha value, which is 0.6. The model has the best accuracy (63.32%) on the validation set when the alpha value is 0.6.

Alpha	Accuracy
0.5	63.27%
0.6	63.32%
0.7	63.27%
0.8	63.16%
0.9	63.18%

Table 6.1: Accuracy for different alpha values in Multinomial Naive Bayes

Logistic Regression: For the logistic regression algorithm, max_iter was selected as the hyperparameter, and different values were assigned to it. Based on the accuracy metric, we chose the best max_iter value, which is 200. When the max_iter value is 200, the model gives the best accuracy on the validation set, which is (64.08%). Although logistic regression gives pretty much the same accuracy across all the maximum iteration values except when max_iter is 100, the next minimum max_iter value, which is 200, was selected as the best hyperparameter value.

max_iter	Accuracy
200	64.08%
300	64.08%
400	64.08%
500	64.08%
600	64.08%

Table 6.2: Accuracy for different max_iter values in Logistic Regression

Random Forest: For the random forest algorithm, n_estimators was selected as the hyperparameter, and different values were assigned to it. Based on the accuracy metric, we chose the best n_estimators value, which is 600. The model has the best accuracy (76.20%) on the validation set when the n_estimators value is 600.

n_estimators	Accuracy
400	75.79%
500	75.98%
600	76.20%
700	75.84%
800	76.07%

Table 6.3: Accuracy for different n_estimators values in Random Forest

Support Vector Machine (SVM): For the SVM algorithm, C was selected, which is known as regularisation, as the hyperparameter, and assigned different values to it. Based on the accuracy metric, we chose the best C value, which is 100. The model has the best accuracy (75.36%) on the validation set when the C value is 100.

C	**Accuracy**
0.1	60.51%
1	72.91%
10	75.26%
100	75.36%
1000	75.31%

Table 6.4: Accuracy for different C values in Support Vector Machine (SVM)

Gradient Boosting Classifier: For the Gradient Boosting Classifier algorithm, n_estimators was selected as the hyperparameter, and different values were assigned to it. Based on the accuracy metric, we chose the best n_estimators value, which is 1000. The model has the best accuracy (67.58%) on the validation set when the n_estimators value is 1000.

n_estimators	**Accuracy**
600	64.98%
700	66.09%
800	66.04%
900	66.65%
1000	67.58%

Table 6.5: Accuracy for different n_estimators values in Gradient Boosting Classifier

6.2.2 Test Set

The standard hyperparameter was achieved from the validation set for each machine learning algorithm. The models were then run on the test set using that hyperparameter to evaluate the models' performance.

Multinomial Naive Bayes: Multinomial Naive Bayes achieved an accuracy of 66.03%, precision of 66.74%, recall of 66.03%, and F1 Score of 65.87%.

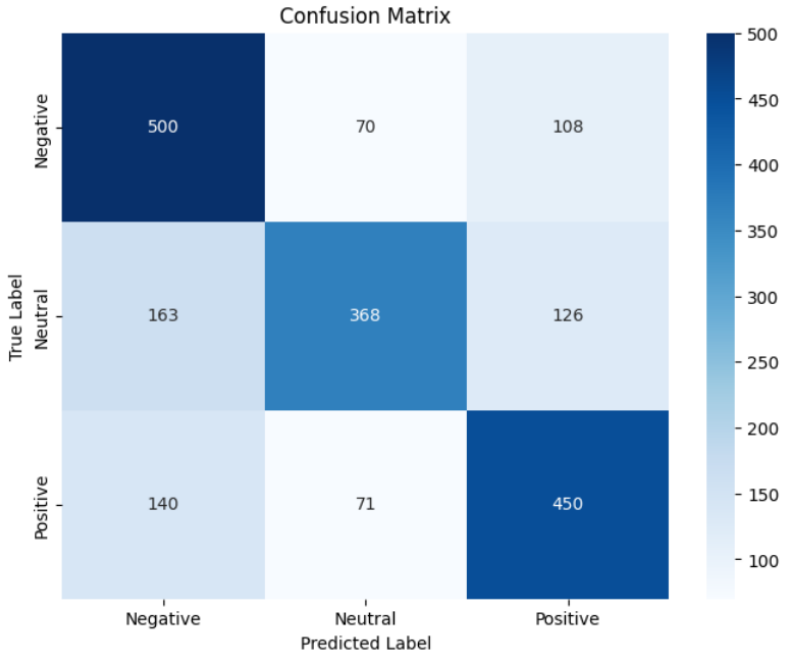

Figure 6.1: MultinomialNB Confusion Matrix

Logistic Regression: Logistic Regression achieved an accuracy of 67.48%, precision of 67.71%, recall of 67.48%, and F1 Score of 67.47%.

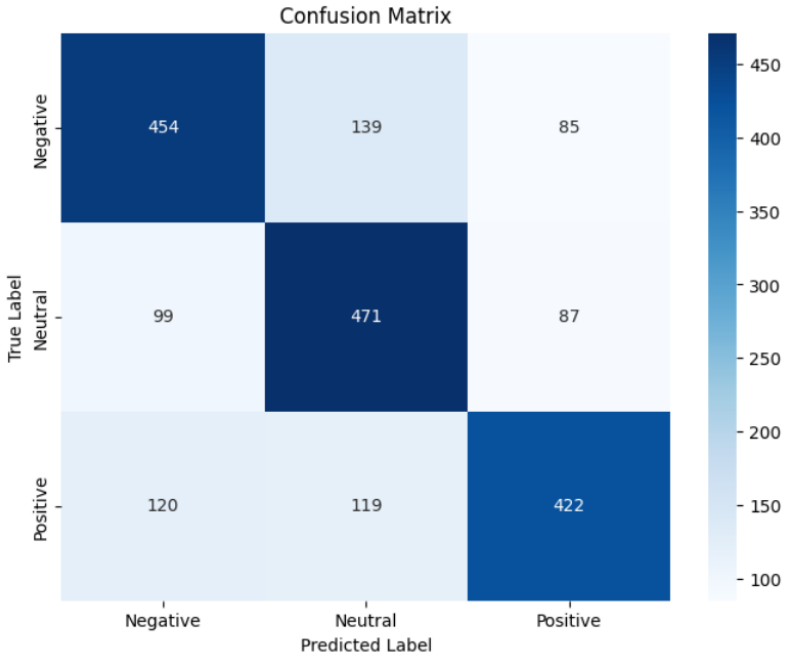

Figure 6.2: Logistic Regression Confusion Matrix

Random Forest: Random Forest achieved an accuracy of 72.34%, precision of 72.67%, recall of 72.34%, and F1 Score of 72.23%.

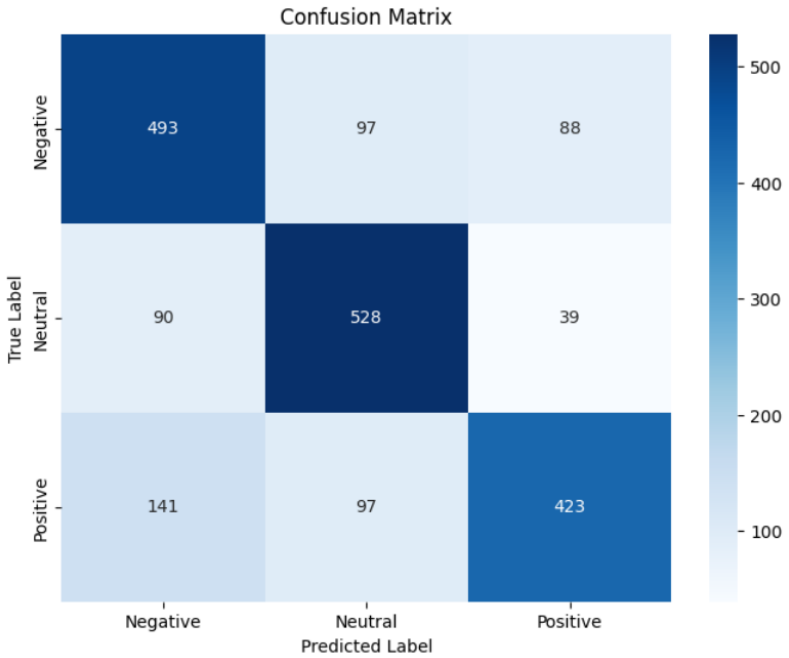

Figure 6.3: Random Forest Confusion Matrix

Support Vector Machine (SVM): SVM achieved an accuracy of 73.00%, precision of 73.03%, recall of 73.00%, and F1 Score of 72.98%.

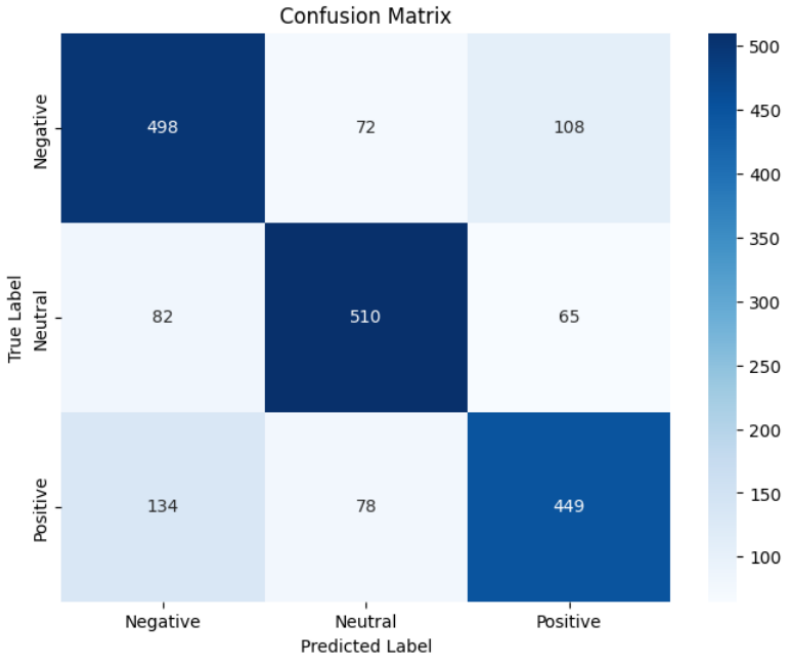

Figure 6.4: Support Vector Machine Confusion Matrix

Gradient Boosting Classifier: Gradient Boosting Classifier achieved an accuracy of 70.79%, precision of 71.09%, recall of 70.79%, and F1 Score of 70.54%.

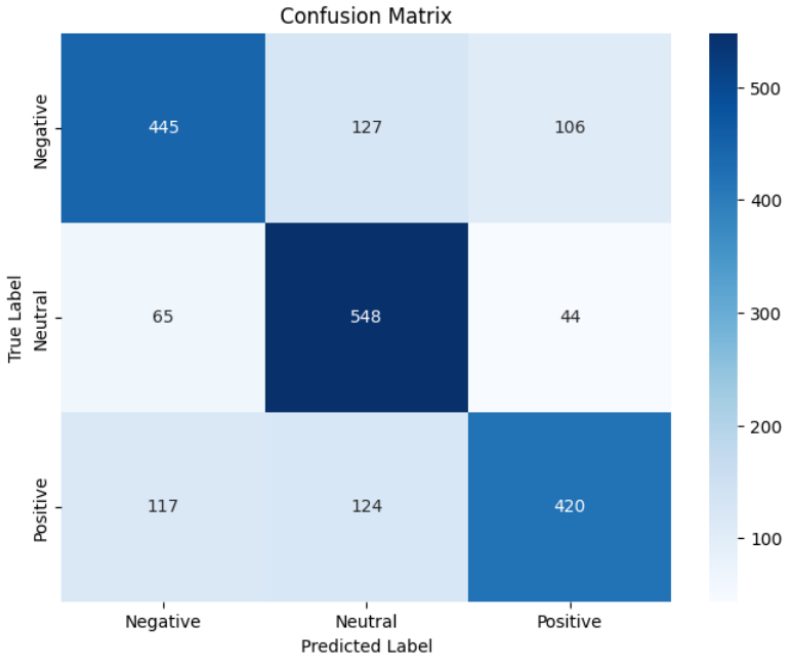

Figure 6.5: Gradient Boosting Classifier Confusion Matrix

Model	Accuracy	Precision	Recall	F1 Score
Multinomial Naive Bayes	66.03%	66.74%	66.03%	65.87%
Logistic Regression	67.48%	67.71%	67.48%	67.47%
Random Forest	72.34%	72.67%	72.34%	72.23%
Support Vector Machine (SVM)	73.00%	73.03%	73.00%	72.98%
Gradient Boosting Classifier	70.79%	71.09%	70.79%	70.54%

Table 6.6: Performance metrics for different models on the test set

After the execution of these five traditional machine learning models, SVM was able to perform better than the other four models. SVM was able to achieve an accuracy of 73.00%, precision of 73.03%, recall of 73.00%, and F1 Score of 72.98%.

SVM Test Classification Report:

	precision	recall	f1-score	support
Negative	0.70	0.73	0.72	678
Neutral	0.77	0.78	0.77	657
Positive	0.72	0.68	0.70	661
accuracy			0.73	1996
macro avg	0.73	0.73	0.73	1996
weighted avg	0.73	0.73	0.73	1996

Figure 6.6: SVM Test Classification Report

6.3 Performance Evaluation: Deep Learning Approach

6.3.1 BiLSTM : Without Fine Tuning GloVe Word Embeddings

For word embedding vectors, we found four glove vectors online dedicated to the Bengali language. Among them, bn_glove.39M.300d works the best. So we chose this file for training our Bidirectional Long Short Term Model (BiLSTM). We used different hyperparameter values to train this model using the dataset that was meant for training and validation. When the learning rate is 0.001, and max_seq_len is 90, the model achieves the best result. We have implemented k-fold cross validation and 10 epochs in each fold. A total of 8 folds were used. In addition, when the model has the lowest validation loss in a certain epoch of a certain fold, we chose that state of the model as the best state for our model, and we saved it. Then we have implemented that state on the test dataset to evaluate the final performance. The results in our test dataset were considered as our main result. If the test dataset has the best results using certain hyperparameter values, that means the model was well trained because of those values.

Max Seq Len	Learning Rate	Fold	Epoch	Train Loss	Val Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
90	0.001	4	3	0.6898	0.7269	69.28%	67.20%	68.58%	67.02%
128	0.001	4	3	0.6887	0.7293	69.40%	68.22%	68.49%	67.03%
184	0.001	4	3	0.6877	0.7313	69.57%	67.58%	68.53%	66.46%
90	0.0001	4	10	0.6955	0.7419	68.45%	66.81%	68.58%	67.02%
128	0.0001	4	10	0.6966	0.7457	68.51%	66.89%	68.49%	67.03%
184	0.0001	4	9	0.7094	0.7424	68.59%	66.77%	68.53%	66.46%

Table 6.7: Performance metrics for various model configurations

The best results in our test dataset are accuracy of 68.58%, precision of 66.51%, recall of 68.58%, and F1 Score of 67.02%. We achieved this result when the learning rate was 0.001 and the maximum sequence length was 90. Since the lowest validation loss is 0.7269 and it was found in the 3rd epoch of the 4th fold, we saved that state and chose it as our best model's performance. We then applied this best model's state to the test dataset to achieve the final result.

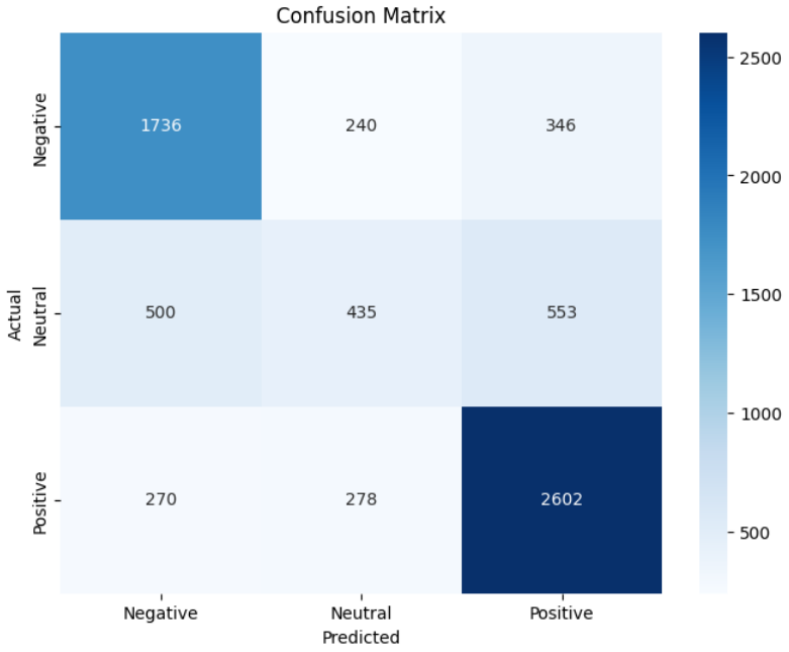

Figure 6.7: BiLSTM Classification Report

6.3.2 BiLSTM : Fine Tuning GloVe Word Embeddings

Since the result we achieved was not satisfactory, we decided to fine-tune the glove vector. The glove vector that we have used was trained on Wikipedia and news articles. However, our dataset contains Bengali product reviews, which have a lot of spelling mistakes and word variations. For this reason, we have fine tuned this glove vector, bn_glove.39M.300d ; on our domain and then used the fine tuned glove embedding to train the model.

We used different hyperparameter values to train this model using the dataset that was meant for training and validation. When the learning rate is 0.001, and max_seq_len is 90, the model achieves the best result. We have implemented k-fold cross validation and 10 epochs in each fold. A total of 8 folds were used. In addition, when the model has the lowest validation loss in a certain epoch of a certain fold, we chose that state of the model as the best state for our model, and we saved

it. Then we have implemented that state on the test dataset to evaluate the final performance. The results in our test dataset were considered as our main result. If the test dataset has the best results using certain hyperparameter values, that means the model was well trained because of those values.

Max Seq Len	Learning Rate	Fold	Epoch	Train Loss	Val Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
90	0.001	2	2	0.5909	0.6145	75.52%	75.03%	74.94%	74.52%
128	0.001	4	2	0.5969	0.6114	75.43%	74.93%	69.32%	66.26%
184	0.001	4	2	0.5903	0.6027	76.03%	75.26%	69.61%	66.31%
90	0.0001	4	9	0.5675	0.6298	74.89%	74.17%	69.02%	66.24%
128	0.0001	4	10	0.5536	0.6293	74.60%	73.46%	69.32%	66.26%
184	0.0001	4	8	0.5704	0.6304	74.63%	73.20%	69.64%	66.85%

Table 6.8: Performance metrics for various model configurations

The best results in our test dataset are accuracy of 74.94%, precision of 74.71%, recall of 75.52%, and F1 Score of 74.52%. We achieved this result when the learning rate was 0.001 and the maximum sequence length was 90. Since the lowest validation loss is 0.7269 and it was found in the 3rd epoch of the 4th fold, we saved that state and chose it as our best model's performance. We then applied this best model's state to the test dataset to achieve the final result.

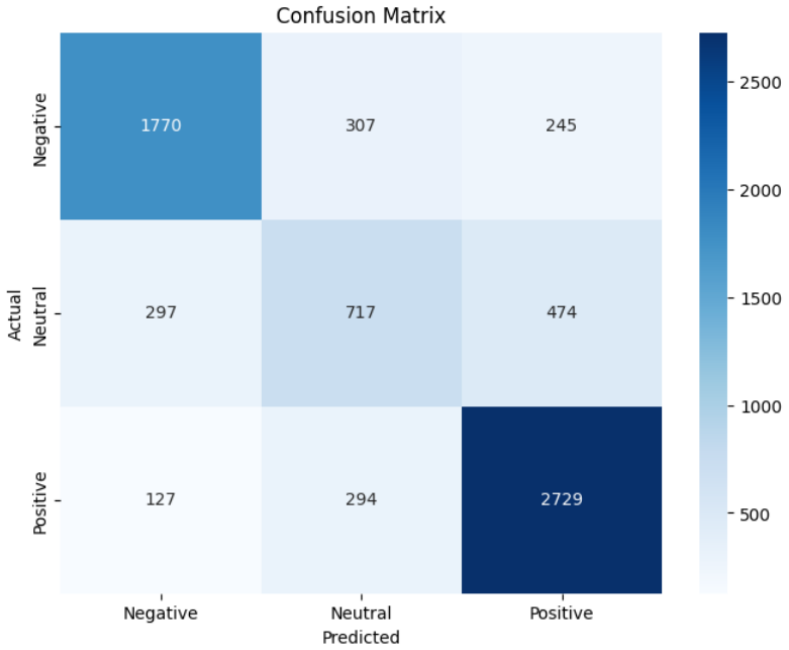

Figure 6.8: BiLSTM Confusion Matrix using Fine Tuned GloVe

6.3.3 BiLSTM : Without Fine Tuning FastText Word Embeddings

Fastext is a library, developed by Facebook for streamlined word representation and text classifications. It utilizes n-grams, taking in information of subword informations and managing them in training more efficiently. For faster training it uses hierarchical softmax, ideal for large scale classification. Because of its speed efficiency and availability of pre-trained models for several languages it is widely used in sentiment analysis, spam detection etc. In the analysis the Bengali model used was collected through the Fastext website. Which was then run through the model and calculations were tabulated

Max Seq Len	Learning Rate	Fold	Epoch	Train Loss	Validation Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
50	0.0001	3	9	0.7948	0.7785	67.23%	63.30%	65.25%	61.54%
90	0.0001	3	8	0.7996	0.7807	67.09%	67.50%	67.63%	64.51%
184	0.0001	3	10	0.7892	0.7764	66.94%	63.28%	66.36%	63.41%
50	0.001	3	7	0.7057	0.7274	69.93%	66.97%	67.76%	65.35%
90	0.001	3	9	0.6846	0.7285	69.26%	67.50%	65.23%	61.54%
184	0.001	3	6	0.7262	0.7271	69.23%	66.31%	67.63%	64.51%

Table 6.9: Performance metrics for various model configurations on Fastext

Of all the hyper-parameters the highest test accuracy of 67.76% F1 score of 63.35%.It was achieved through a learning rate of 0.001 and max_seq_length of 50. During the training process the model achieved a best result of 69.93% accuracy during 0.7057 training loss and 0.7274 validation loss on 7th epoch of 3rd fold.

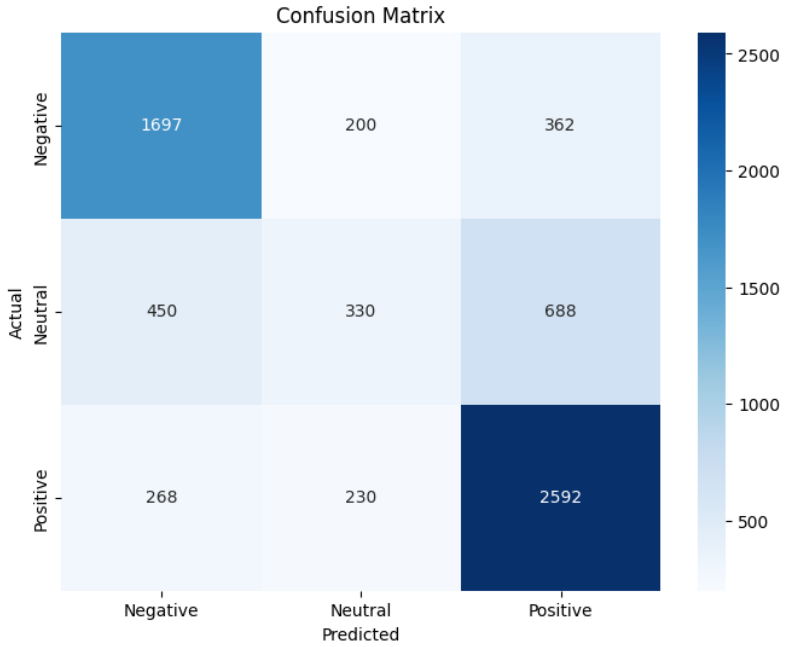

Figure 6.9: BiLSTM Confusion Matrix using Fastext

6.3.4 BiLSTM : Fine Tuning FastText Word Embeddings

Due to the underwhelming results, the embedding model was fine-tuned through supervised training method with the dataset. After which the performance of the model significantly improved. The fine-tuned model achieved the best test results of

accuracy of 76.47% and 75.59% f1 score. During 10th epoch of 2nd fold the model achieved best training result of 78.18% with validation loss of 0.5738 and training loss of 0.5828.

Max Seq Len	Learning Rate	Fold	Epoch	Train Loss	Validation Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
50*	0.0001	3	10	0.6642	0.6464	74.19%	73.58%	73.52%	72.41%
90*	0.0001	3	10	0.6613	0.6404	74.60%	73.56%	73.60%	72.48%
184*	0.0001	3	10	0.6656	0.6446	74.42%	73.62%	73.60%	72.48%
50*	0.001	1	10	0.5831	0.5701	77.30%	77.07%	76.32%	75.50%
90*	0.001	1	10	0.5861	0.5704	77.21%	76.29%	76.35%	75.52%
184*	0.001	2	10	0.5828	0.5738	78.18%	77.21%	76.47%	75.59%

Table 6.10: Performance metrics for various model configurations on Finetuned Fastext

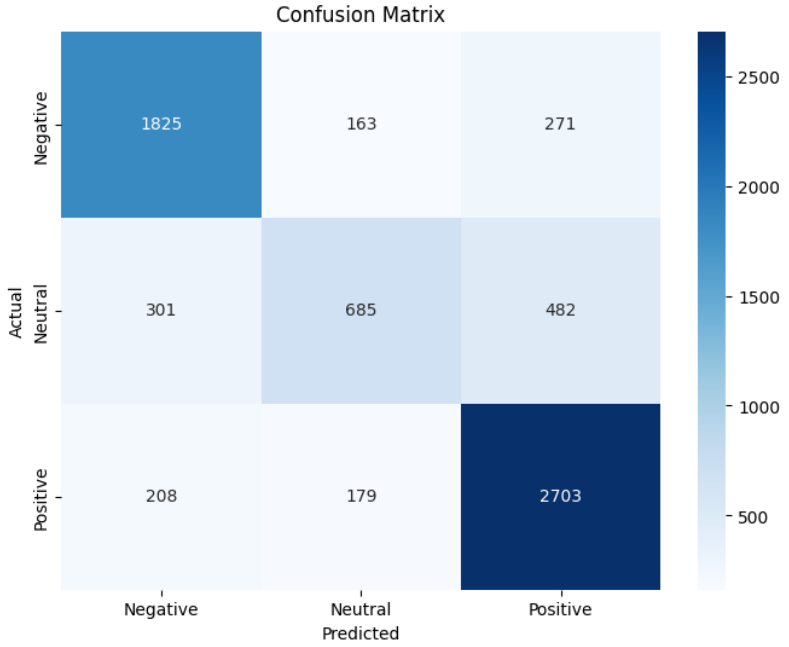

Figure 6.10: BiLSTM Confusion Matrix using Fastext

6.3.5 Multilingual BERT: Without Fine Tuning

BERT models do not need external pre-trained word embeddings such as glove or fasttext. BERT models learn word embeddings implicitly during training. Therefore, we had to choose a model that is capable of learning Bengali words since our dataset contains Bengali languages. So we chose Multilingual BERT.

Since k-fold cross validation was not used for training BERT, a separate training and validation dataset was used. We used different hyperparameter values to train this model using the training dataset. A validation dataset was used to evaluate the model's training performance after each epoch. When the learning rate is 0.00001 and max_length is 184, the model achieves the best result. In addition, when the model has the lowest validation loss in a certain epoch, we chose that state of the model as the best state for our model, and we saved it. Then we have implemented that state on the test dataset to evaluate the final performance. The results in our test dataset were considered as our main result. If the test dataset has the best

results using certain hyperparameter values, that means the model was well trained because of those values.

Max Length	Learning Rate	Epoch	Train Loss	Val Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
30	0.00001	4	0.6425	0.7933	67.22%	65.89%	65.57%	62.59%
70	0.00001	3	0.6752	0.7681	68.67%	67.23%	69.27%	68.28%
184	0.00001	2	0.7259	0.7434	69.54%	67.44%	69.74%	67.93%

Table 6.11: Performance metrics for various model configurations

The best results in our test dataset are accuracy of 69.74%, precision of 67.62%, recall of 69.74%, and F1 Score of 67.93%. We achieved this result when the learning rate was 0.00001 and the maximum sequence length was 184. Since the lowest validation loss is 0.7434 and it was found in the 2nd epoch, we saved that state and chose it as our best model's performance. We then applied this best model's state to the test dataset to achieve the final result.

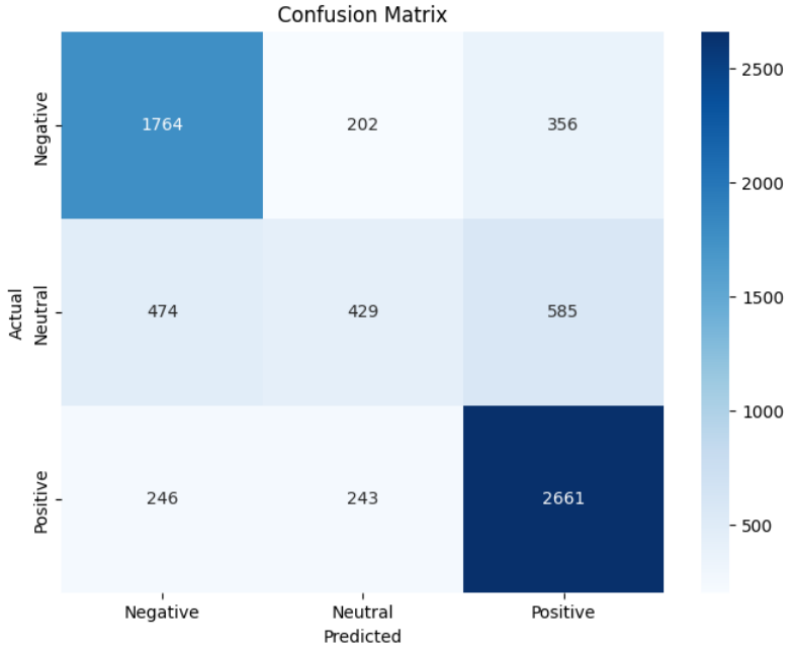

Figure 6.11: Multilingual BERT Confusion Matrix

6.3.6 Multilingual BERT: Fine Tuning

Since the result we achieved was not satisfactory, we decided to fine-tune the multilingual BERT model. This model was trained on Wikipedia and news articles. However, our dataset contains Bengali product reviews, which have a lot of spelling mistakes and word variations. For this reason, we have fine-tuned this BERT model in our domain.

Since k-fold cross validation was not used for training the Fine Tuned BERT model, a separate training and validation dataset was used. We used different hyperparameter values to train this model using the training dataset. A validation dataset was used to evaluate the model's training performance after each epoch. When the learning rate is 0.00001 and max_length is 184, the model achieves the best result. In addition, when the model has the lowest validation loss in a certain epoch, we chose that state of the model as the best state for our model, and we saved it. Then we have implemented that state on the test dataset to evaluate the final performance. The results in our test dataset were considered as our main result. If the test dataset has the best results using certain hyperparameter values, that means the model was well trained because of those values.

Max Length	Learning Rate	Epoch	Train Loss	Val Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
30	0.00001	1	0.7357	0.7127	70.48%	69.39%	66.02%	63.41%
70	0.00001	1	0.7006	0.6908	71.74%	70.24%	73.18%	72.00%
184	0.00001	2	0.6454	0.6628	72.99%	71.71%	74.11%	73.10%

Table 6.12: Performance metrics for various model configurations

The best results in our test dataset are accuracy of 74.11%, precision of 72.73%, recall of 74.11%, and F1 Score of 73.10%. We achieved this result when the learning rate was 0.00001 and the maximum sequence length was 184. Since the lowest validation loss is 0.6628 and it was found in the 2nd epoch, we saved that state and chose it as our best model's performance. We then applied this best model's state to the test dataset to achieve the final result.

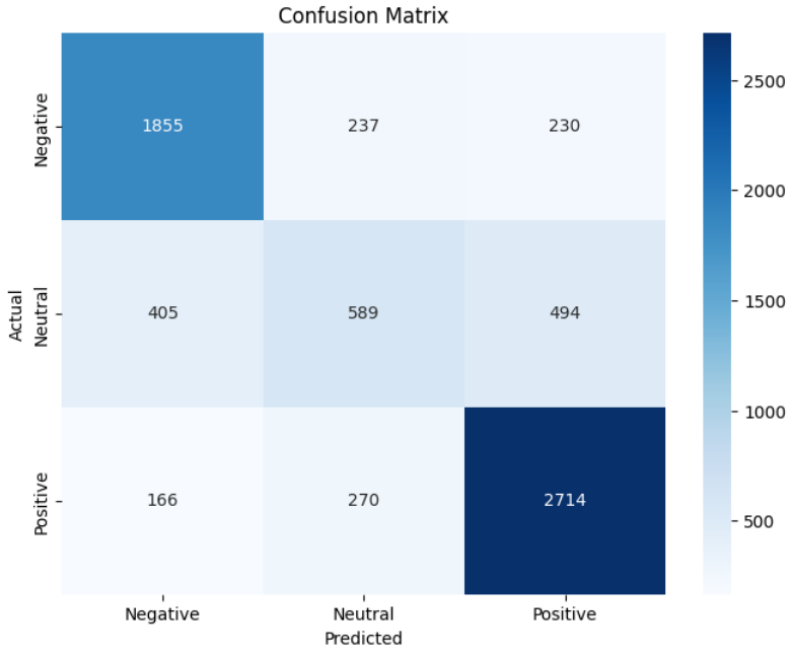

Figure 6.12: Fine Tuned Multilingual BERT Confusion Matrix

6.3.7 BanglaBERT: Without Fine Tuning

Besides Multilingual BERT, we also used BanglaBERT, which was trained on Bengali language.

Since k-fold cross validation was not used for training BanglaBERT, a separate training and validation dataset was used. We used different hyperparameter values to train this model using the training dataset. A validation dataset was used to evaluate the model's training performance after each epoch. When the learning rate is 0.00001 and max_length is 70, the model achieves the best result. In addition, when the model has the lowest validation loss in a certain epoch, we chose that state of the model as the best state for our model, and we saved it. Then we have implemented that state on the test dataset to evaluate the final performance. The results in our test dataset were considered as our main result. If the test dataset has the best results using certain hyperparameter values, that means the model was well trained because of those values.

Max Length	Learning Rate	Epoch	Train Loss	Val Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
30	0.00001	4	0.6895	0.7712	67.61%	65.56%	68.36%	66.67%
70	0.00001	2	0.7672	0.7688	67.72%	66.07%	68.81%	67.33%
184	0.00001	4	0.6803	0.7583	67.05%	67.14%	68.49%	68.66%

Table 6.13: Performance metrics for various model configurations

The best results in our test dataset are accuracy of 68.81%, precision of 67.06%, recall of 68.81%, and F1 Score of 67.33%. We achieved this result when the learning rate was 0.00001 and the maximum sequence length was 70. Since the lowest validation loss is 0.7688 and it was found in the 2nd epoch of the, we saved that state and chose it as our best model's performance. We then applied this best model's state to the test dataset to achieve the final result.

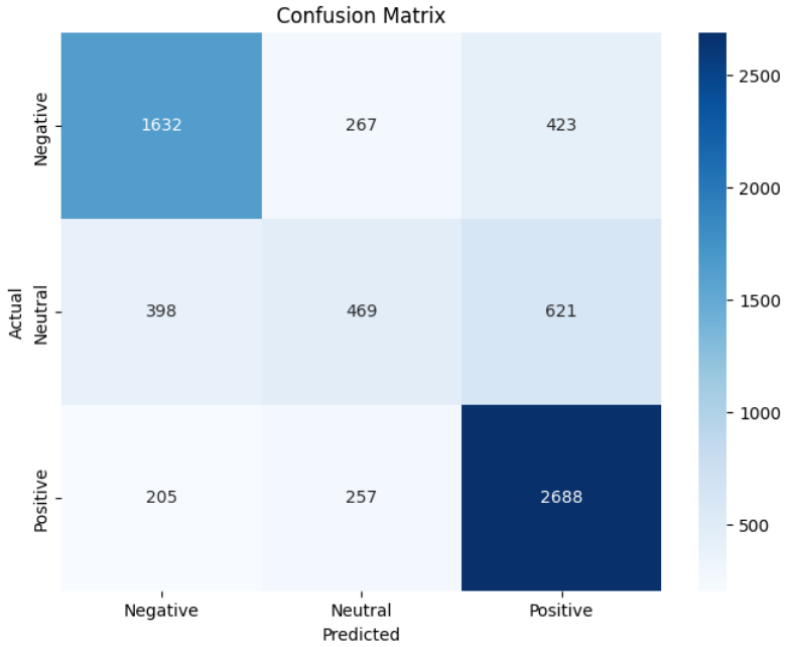

Figure 6.13: BanglaBERT Confusion Matrix

6.3.8 BanglaBERT: Fine Tuning

Since the result we achieved was not satisfactory, we decided to fine-tune the BanglaBERT model. This model was pretrained on popular Bangladeshi sites. However, our dataset contains Bengali product reviews, which have a lot of spelling mistakes and word variations. For this reason, we have fine-tuned this BERT model in our domain.

Since k-fold cross validation was not used for training Fine Tuned BanglaBERT, a separate training and validation dataset was used. We used different hyperparameter values to train this model using the training dataset. A validation dataset was used to evaluate the model's training performance after each epoch. When the learning rate is 0.00001 and max_length is 184, the model achieves the best result. In addition, when the model has the lowest validation loss in a certain epoch, we chose that state of the model as the best state for our model, and we saved it. Then we have implemented that state on the test dataset to evaluate the final performance. The results in our test dataset were considered as our main result. If the test dataset has the best results using certain hyperparameter values, that means the model was well trained because of those values.

Max Length	Learning Rate	Epoch	Train Loss	Val Loss	Accuracy	F1 Score	Test Accuracy	Test F1 Score
30	0.00001	1	0.7078	0.6948	71.35%	70.94%	72.61%	72.33%
70	0.00001	1	0.6959	0.6897	71.42%	67.16%	72.11%	68.24%
184	0.00001	1	0.6993	0.6711	72.74%	70.57%	72.74%	70.57%

Table 6.14: Performance metrics for various model configurations

The best results in our test dataset are accuracy of 72.74%, precision of 70.55%, recall of 72.74%, and F1 Score of 70.57%. We achieved this result when the learning rate was 0.00001 and the maximum sequence length was 184. Since the lowest validation loss is 0.6711 and it was found in the 1st epoch, we saved that state and chose it as our best model's performance. We then applied this best model's state to the test dataset to achieve the final result.

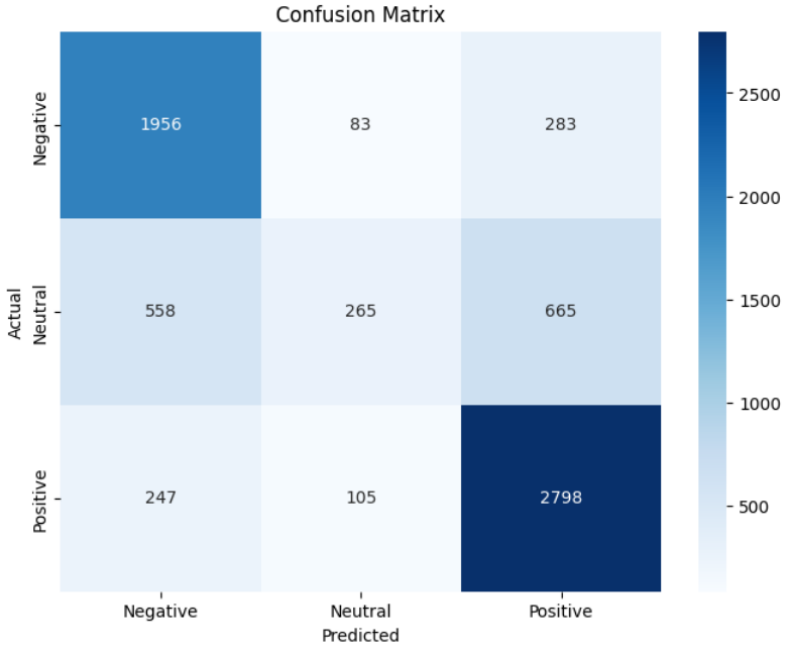

Figure 6.14: Fine Tuned BanglaBERT Confusion Matrix

6.3.9 Best Deep Learning Model

Model	Accuracy	Precision	Recall	F1 Score
BiLSTM (GloVe)	68.58%	66.51%	68.58%	67.02%
BiLSTM (Fine Tuned GloVe)	74.94%	74.71%	75.52%	74.52%
BiLSTM (FastText)	67.76%	66.66%	67.73%	65.35%
BiLSTM (Fine Tuned FastText)	76.47%	75.73%	76.47 %	75.59%
Multilingual BERT	69.74%	67.62%	69.74%	67.93%
Fine Tuned Multilingual BERT	74.11%	72.73%	74.11%	73.10%
BanglaBERT	68.81%	67.06%	68.81%	67.33%
Fine Tuned BanglaBERT	72.74%	70.55%	72.74%	70.57%

Table 6.15: Performance metrics for Deep Learning Models

Classification Report:

	precision	recall	f1-score	support
Negative	0.78	0.81	0.79	2259
Neutral	0.67	0.47	0.55	1468
Positive	0.78	0.87	0.83	3090
accuracy			0.76	6817
macro avg	0.74	0.72	0.72	6817
weighted avg	0.76	0.76	0.76	6817

Figure 6.15: BiLSTM (Fastext) Test Classification Report

After the execution of these deep learning models, BiLSTM (Fine Tuned Fastext) was able to perform better than the other deep learning models. This fine tuned model was able to achieve an accuracy of 76.47%, precision of 75.73%, recall of 76.47%, and F1 Score of 75.59%.

6.4 Best Model

Model	Accuracy	Precision	Recall	F1 Score
Multinomial Naive Bayes	66.03%	66.74%	66.03%	65.87%
Logistic Regression	67.48%	67.71%	67.48%	67.47%
Random Forest	72.34%	72.67%	72.34%	72.23%
Support Vector Machine (SVM)	73.00%	73.03%	73.00%	72.98%
Gradient Boosting Classifier	70.79%	71.09%	70.79%	70.54%
BiLSTM (GloVe)	68.58%	66.51%	68.58%	67.02%
BiLSTM (Fine Tuned GloVe)	74.94%	74.71%	75.52%	74.52%
BiLSTM (FastText)	67.76%	66.66%	67.73%	65.35%
BiLSTM (Fine Tuned FastText)	76.47%	75.73%	76.47 %	75.59%
Multilingual BERT	69.74%	67.62%	69.74%	67.93%
Fine Tuned Multilingual BERT	74.11%	72.73%	74.11%	73.10%
BanglaBERT	68.81%	67.06%	68.81%	67.33%
Fine Tuned BanglaBERT	72.74%	70.55%	72.74%	70.57%

Table 6.16: Performance metrics for different models on the test set

Out of all these models, BiLSTM (Fine Tuned FastText) performed better with an accuracy rate of 76.67%. The main reason behind this is the mislabeling of the sentiments due to ratings. As it was not possible to manually check all the data in the dataset and fix them, the model was not able to correctly classify all the data during the training. There will be further work to solve this issue in the future.

Chapter 7

Future Work

Future work Plan:

- For better results and efficient work, NLP requires larger data sets. There is a scarcity of proper Bengali data-set that contains a large amount of data. Therefore, the dataset length will be increased in our future work.
- Mislabeling of sentiments will be fixed
- BiLSTM (Fine Tuned Fastext)achieved 76.47% accuracy, which is so far the best performing model in our thesis which is not that impressive. For this reason, the aim will be to improve the performance of the machine learning models in the future.

Chapter 8

Conclusion

As the internet has become a free medium for users to express their opinions, customer reviews have become valuable sources for businesses to improve their service. In this thesis, we used five machine learning models and three deep learning models to classify emotion from consumer opinion. Three emotions were considered for this study: positive, negative, and neutral. The traditional machine learning models we have employed for this work are multinomial naive bayes, logistic regression, random forest, support vector machine (SVM), and gradient boosting classifiers. The deep learning models we have employed for this work are BiLSTM, multilingual BERT, and BanglaBERT. These three models were further finetuned based on their word embedding vectors. Out of all these models, BiLSTM (Fine Tuned Fastext) outperformed all with an accuracy of 76.47%. So, there is still room for improvement, which will be done in the future.

Bibliography

[1] Mst. Tuhin Akter, Manoara Begum, and Rashed Mustafa. Bengali sentiment analysis of e-commerce product reviews using k-nearest neighbors. In *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pages 40–44, 2021.

[2] Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States, July 2022. Association for Computational Linguistics.

[3] Sovon Chakraborty, Muhammad Borahn Uddin Talukdar, Muhammed Yaseen Morshed Adib, Sowmen Mitra, and Md. Golam Rabiul Alam. Lstm-ann based price hike sentiment analysis from bangla social media comments. In *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pages 733–738, 2022.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[5] Sanjay Dey, Sarhan Wasif, Dhiman Sikder Tonmoy, Subrina Sultana, Jayjeet Sarkar, and Monisha Dey. A comparative study of support vector machine and naive bayes classifier for sentiment analysis on amazon product reviews. In *2020 International Conference on Contemporary Computing and Applications (IC3A)*, pages 217–220, 2020.

[6] Zabit Hameed and Begonya Garcia-Zapirain. Sentiment classification using a single-layered bilstm model. *IEEE Access*, 8:73992–74001, 2020.

[7] Tanjim Ul Haque, Nudrat Nawal Saber, and Faisal Muhammad Shah. Sentiment analysis on large scale amazon product reviews. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pages 1–6, 2018.

[8] Md. Jahed Hossain, Dabasish Das Joy, Sowmitra Das, and Rashed Mustafa. Sentiment analysis on reviews of e-commerce sites using machine learning algorithms. In *2022 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, pages 522–527, 2022.

[9] Tuhin Hossain, Ahmed Ainun Nahian Kabir, Md. Ahasun Habib Ratul, and Abdus Sattar. Sentence level sentiment classification using machine learning approach in the bengali language. In *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, pages 1286–1289, 2022.

[10] R. S. Jagdale, V. S. Shirsat, and S. N. Deshmukh. Sentiment analysis on product reviews using machine learning techniques. In P. Mallick, V. Balas, A. Bhoi, and A. Zobaa, editors, *Cognitive Informatics and Soft Computing*, volume 768 of *Advances in Intelligent Systems and Computing*. Springer, 2019.

[11] Yi Liu, Jiahuan Lu, Jie Yang, and Feng Mao. Sentiment analysis for e-commerce product reviews by deep learning model of bert-bigru-softmax. *Mathematical Biosciences and Engineering*, 17(6):7819–7837, 2020.

[12] Md Maruf Rayhan, Taif Al Musabe, and Md Arafatul Islam. Multilabel emotion detection from bangla text using bigru and cnn-bilstm. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–6, 2020.

[13] Ehsanur Rahman Rhythm, Rajvir Ahmed Shuvo, Md Sabbir Hossain, Md. Farhadul Islam, and Annajiat Alim Rasel. Sentiment analysis of restaurant reviews from bangladeshi food delivery apps. In *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 1–5, 2023.

[14] Guixian Xu, Yueting Meng, Xiaoyu Qiu, Ziheng Yu, and Xu Wu. Sentiment analysis of comment texts based on bilstm. *IEEE Access*, 7:51522–51532, 2019.