Rating Detection by Reviews using ML and NLP towards Mobile Phone Recommendation

by

Md Wafi Sarker
18101449
Sheikh Sanzid Al Nahian
18101381
Anjumand Moshtari Khan
21101113
Abdullah Oraib
18301207
Sikder Mohidul Islam
18101146

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 and Engineering

Department of Computer Science and Engineering
Brac University
January 2023

Declaration

It is hereby declared that

1. The thesis submitted is 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:

Md. Wafi Sarker

Md Wafi Sarker

18101449

Sanzid Al Nahian

Sheikh Sanzid Al Nahian

18101381

Anjumand Khan

Anjumand Moshtari Khan

21101113

Abdullah oraib

Abdullah Oraib

18301207

Mohidul Islam

Sikder Mohidul Islam

18101146

Approval

The thesis/project titled "Rating Detection by Reviews using ML and NLP towards Mobile Phone Recommendation" submitted by

1. Md Wafi Sarker (18101449)
2. Sheikh Sanzid Al Nahian (18101381)
3. Anjumand Moshtari Khan (21101113)
4. Abdullah Oraib (18301207)
5. Sikder Mohidul Islam (18101146)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on January 17, 2023.

Examining Committee:

Supervisor:
(Member)

Farig Yousuf Sadeque

Dr. Farig Yousuf Sadeque (PhD)

Assitant Professor
Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD

Assistant 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

Ethics Statement

Hereby, we, the members, consciously assure that for the manuscript /insert title/ the following is fulfilled:

• This content is the authors' original work that has never been published before.

• The authors' research and analysis are accurately and completely represented in the work.

• Co-authors and co-researchers are appropriately acknowledged in the study for their significant contributions.

• The findings are contextualized appropriately in light of previous and ongoing research.

• All sources are cited appropriately. Text that has been literally copied must be identified as such with quote marks and a suitable reference.

• All authors personally and actively contributed to the article's creation and will accept public responsibility for its content.

Violations of the Ethical Statement standards may result in serious consequences. We agree with the preceding declarations and certify that this submission complies with the rules of BRAC University.

Abstract

Product recommendation is a type of marketing tool that has become increasingly important for businesses as well as in purchasing goods in the digital age. Product recommendation is the process of suggesting items to customers based on their previous purchases or choices and is a form of personalization where the goal is to provide relevant, valuable, and timely information to customers to help them make decisions about what to buy. The purpose of product recommendations is to increase customer engagement, loyalty, and ultimately, sales while ensuring customers help buying products according to their preference. By providing customers with personalized product recommendations, businesses are able to increase customer satisfaction and loyalty, as well as drive sales. On the other way, customers also feel secure while purchasing products according to their personality and choices. This paper builds a product recommendation system by analyzing the techniques of Machine Learning and Natural Language Processing. The focus of the research is on recommending mobile phone products to users based on their preferences and interests. The system was advanced and examined using a dataset of mobile phone specifications and user reviews. The study's findings demonstrate that the suggested recommendation system may offer users accurate and pertinent ideas; but, due to dataset restrictions, the system cannot be expanded to include other kinds of products. However, the proposed system can be used for taking personalized requirements and finding a better result for them with improved accuracy and precision which ultimately will enhance customer satisfaction.

Keywords: Recommendation system, Natural language processing, Machine learning, Deep learning, Sentimental analysis, Long short term memory, Naive bayes, Convolutional neural network, Support vector machine, Multi layer perceptron, Gradient booster machine, Stochastic gradient descent, Random forest

Dedication (Optional)

This thesis is dedicated to our university's mentors, without whom we would not have been capable of completing this thesis. Our professors were more than just academic mentors; they were also a source of encouragement and support when we needed it most.

Acknowledgement

In the first place, we would like to thank Almighty Allah for allowing us to complete our thesis on time and without deterrent.

Having said that, we would like to express our gratitude to Dr. Farig Yousuf Sadeque (PhD), our distinguished instructor and supervisor, for his unwavering support and tenacious oversight, which allowed us to complete our project. In addition, we would like to thank our supportive friends who have been there for us during the difficult times. Lastly, to our parents, without their unwavering support it may be impossible. With their gracious assistance and prayers, we are now on the verge of graduating.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AUC AUC: Area Under the ROC Curve

CNN Convolutional Neural Network

GBM Gradient Booster Machine

$HTML$ HyperText Markup Language

LR Logistic Regression

$LSTM$ Long Short Term Memory

MAE Mean Absolute Error

ML Machine Learning

MLP Multi Layer Perceptron

NB Naive Bayes

NLP Natural Language Processing

$ReLU$ Rectified Linear Units

RF Random Forest

$RMSE$ Root Mean Square Error

SGD Stochastic Gradient Descent

SVM Support Vector Machine

$TF-IDF$ term frequency-inverse document frequency

Chapter 1

Introduction

Businesses must discover ways to stand out from the competition in the cutthroat retail market of today, while also giving customers an exceptional experience. One way to do this is through the use of personalized product recommendations, which can help to increase customer satisfaction and loyalty.

In Our thesis, we will be using Natural Language Processing and Machine Learning (ML) techniques to analyze customer reviews and feedback in order to identify patterns and trends that can be used to make more targeted product recommendations like Mobile . By using customer data to understand individual preferences and needs, we aim to create a system that can provide personalized recommendations to each customer, increasing the likelihood that they will find mobile phones that meet their expectations.

We believe that this thesis has the potential to make a significant impact on customer satisfaction and loyalty, and we are excited to see the results of our work. The paper's aim of creating a product recommendation system that can adjust to evolving customer preferences over time is one of its primary objectives. However, due to limitations of vast areas of research and datasets, we only limited our research to Mobile phone recommendation systems for now. To do this, we will be using the algorithms of ML that are able to learn and improve their performance as they are exposed to more data. This will allow the system to continuously evolve and become more effective at making personalized recommendations.

To assess consumer evaluations and feedback, we will also use Natural Language Processing (NLP) methods in addition to ML algorithms. By understanding the sentiment and meaning behind customer comments, we can gain a better understanding of their preferences and needs [60]. This information can then be used to make more targeted mobile phone recommendations that are more likely to be successful.

To develop and test our model, we used a sizable dataset of customer evaluations and comments. On the basis of the model's performance on this dataset, we can assess its efficacy and make any necessary adjustments.

Overall, we believe that this paper has the potential to significantly improve customer satisfaction and loyalty by providing personalized mobile phones recommendations that are more likely to meet the needs and preferences of individual customers. We are excited to see the results of our work and hope that it will have a positive impact on the retail industry. Finding a technique to precisely forecast which products a consumer will be interested in is one of the main challenges in developing a mobile phone suggestion system. This requires understanding the factors that influence a customer's decision-making process and using this information to make more accurate recommendations [60]. By using NLP and ML techniques, we aim to identify these factors and use them to make more targeted recommendations.

In addition to analyzing customer reviews and feedback, it may also be useful to incorporate other types of data into the recommendation system. This could include information about the customer's previous purchases, browsing history, and demographic information. By using a variety of data sources, we can create a more comprehensive picture of each customer's preferences and needs.

As with any ML project, it will be important to carefully test the performance of the model to ensure that it is effective. This will involve comparing the recommendations made by the model to the products that customers actually end up purchasing, and using this information to fine-tune the model as needed.

It will also be important to consider the user experience when designing the following system. Customers should be able to easily access the system and receive recommendations that are relevant and useful to them.

Incorporating user feedback on the given recommendations is one technique to increase the efficacy of the recommendation process. By asking customers to rate the relevance of the recommendations they receive, we can gather data on what works and what doesn't, and use this information to make the process more accurate over time [30]. Another aspect to consider is the integration of the recommendation process with other parts of the business. For example, the process could be integrated with the company's inventory management system to ensure that recommended products are actually in stock and available for purchase [30].

It will also be important to consider the ethical implications of the recommendation. For instance, flaws in the data or algorithms employed should be avoided as they may lead to suggestions that are unfair or discriminating [30].

Further, it will be important to consider the long-term maintenance and updates of the recommendation process[14]. As customer preferences change and new products are introduced, the process will need to be updated and refined in order to remain effective. Recommendation refers to the process of suggesting items or products to a user based on their interests, preferences, or past behavior. Product recommendation specifically refers to the process of suggesting products to a user [14]. This can be done through various means, such as recommending products to a user based on their past purchases or items they have viewed on a website, or by using ML algorithms to analyze a user's behavior and predict which products they might be interested in. Product recommendation systems are commonly used by online retailers to help users discover new products that they might be interested in and to increase sales.So, basically this article will be highlighting how product recommendation systems for customer satisfaction are developed with the help of ml and NLP [14].

Customer satisfaction can be raised by utilizing a product recommendation technique that combines ML with NLP. While ML algorithms can be trained to anticipate which products a customer is likely to be intrigued by based on their prior purchases and behavior, NLP approaches can be used to extract significant features and insights from customer evaluations and feedback [63]. To build a mobile phone recommendation system using NLP and ML, you will need to follow a few steps such as Collecting and preprocessing data: You will need to gather a large dataset of customer reviews, ratings, and other relevant information [63]. You will then need to preprocess this data by cleaning it, removing any irrelevant or duplicate information, and formatting it in a way that can be used by your ML model. Extract features: Use NLP techniques to extract important features and insights from customer reviews and ratings. This could entail detecting important terms and phrases, classifying the reviews' sentiment, and figuring out the product's overall sentiment. Develop a ML model: Create a ML model that could really forecast which products a customer is most likely to be interested in using the extracted attributes and customer data. For this, a variety of ML methods, such as decision trees, random forests, and neural networks, can be applied. Evaluate the model: Once you have trained your model, it is important to evaluate its performance to ensure that it is making accurate recommendations. You can do this by comparing the recommendations made by the model with the actual mobile phones that customers end up purchasing [63]. By following these steps, you can build a mobile phone recommendation system that uses NLP and ML to improve customer satisfaction by providing personalized product recommendations. The aim of a mobile phone recommendation method that makes use of NLP and ML is to use cutting-edge techniques and algorithms to evaluate consumer data and offer individualized product recommendations in order to increase customer satisfaction.

The model works by first collecting and preprocessing customer data, such as product reviews and ratings, purchase history, and demographic information. NLP techniques are then used to extract important features and insights from this data, such as the sentiment of the reviews and the keywords and phrases used by customers to describe the product.A ML model is then trained using these extracted features to make predictions about which mobile phone a client is most probable to be engaged in based on their prior purchasing history and preferences. The model can then make recommendations to customers based on this prediction, helping them find mobile phones that are more relevant and appealing to them. Overall, the goal of the model is to use advanced analytics and ML techniques by ensuring increased sales, and increased efficiency to improve the customer experience by providing personalized mobile phone recommendations that are tailored to the individual preferences of each customer.

1.1 Thoughts behind the Prediction Model

The thoughts behind this model are to combine ML and NLP methods to produce a product recommendation system that enhances consumer happiness. By analyzing customer reviews and feedback, the model can identify patterns and trends that can be used to recommend products that are more likely to meet the needs and preferences of individual customers. This can help to increase customer satisfaction by providing them with personalized product recommendations that are more likely to meet their expectations. Additionally, by using NLP and ML techniques, the model can continuously learn and adapt to changing customer preferences, ensuring that it remains effective over time. One of the key benefits of using a product recommendation system based on NLP and ML is that it will help businesses to understand their customers and their needs. By analyzing customer reviews and feedback, the model can identify common themes and patterns that can be used to identify the types of products that customers are most interested in. This can be particularly useful for businesses that sell a wide range of products, as it can help them to focus their marketing efforts on the products that are most likely to be of interest to their customers.

In addition to helping businesses to better understand their customers, a product recommendation system based on NLP and ML can also help to increase customer satisfaction by providing personalized product recommendations. By using customer data to make more targeted recommendations, businesses can increase the likelihood that their customers will find products that meet their needs and preferences. This can help to increase customer loyalty and retention, as customers are more likely to return to a business that provides them with products that they are satisfied with.

Overall, the use of NLP and ML techniques in a product recommendation system can be a powerful tool for businesses looking to improve customer satisfaction and loyalty. By using data and ML algorithms to understand customer preferences and make personalized recommendations, businesses can create a more personalized and effective shopping experience for their customers.

1.2 Motivation

Consumers have been using the internet to research almost any purchase for more than a decade. People have been looking for information on websites where they can visit specific stores or e-commerce websites and choose what they would like to buy. Currently, some of the top companies in the world are making recommendations. For instance, Amazon and Alibaba have spent years developing and adjusting their algorithms. The way customers are matched with products they are interested in buying has radically changed because of the company's recommendation engine, which is made possible by its unique access to vast volumes of consumer data.

In such recommendation engines, there are divisions based on the categories of products. Each category has a vast amount of products which means a vast amount of data alongside each product. So, we have narrowed down the categories to a single category and that is "smartphones". Moreover, smartphones have become daily fundamental devices where on the go anybody can perform most tasks in it.

There are hundreds and thousands of smartphone brands globally. Each smartphone can deliver different needs to the customer. Some customers need it to communicate with social media with others and to be active with their tasks, while others need it for audio and video consumption, gaming, and photography. Several aspects can be delivered to the consumer with so many options to choose from. There are also many smartphone brands that have the same specifications and also sell at the same price, where competition comes which causes customers to struggle in decision making. That is where recommendation systems come in to help customers to decide on buying which smartphone they desire according to their demands. The recommendation system will filter among all the available smartphones and choose the best ones according to customer preference.

1.3 Research Objective

A product recommendation system for customer satisfaction should be used by any individual who will choose a product essential enough to meet their demands without the hassle they need to go through various sites to know the specifications and pricing of that particular product. The aim of this research is to develop a recommendation system that will help customers in being happy with the products they choose. Many customers may struggle to find a compatible product according to their demands for specification, reliability, warranty services, and most importantly price. A product recommender system, which is powered by ML, is the technology used to suggest which products are displayed to customers to interact with a brand's digital properties. In order to provide each user with a customized experience, recommendation algorithms collect user, product, and contextual data for both onsite and offsite use. By improving the discovery process, users are aided in finding what they are seeking and often products they weren't even aware they were looking for. Businesses can gain a better understanding of each user's particular tastes and interests in this way, boosting performance in the present while also making long-term improvements to their testing roadmaps. The necessity for this model arises as a result of the oncoming tidal wave of technological advancements.

Our key goal is to improve our capacity for making product recommendations that take into account user preferences. Recommendations can give important information and the chance to learn more about a customer in order to please them, create value, and enhance their relationship with a brand. Consequently, the goals of this study are:

1. To thoroughly comprehend various NLP and ML classifiers.

2. Training different models and ML classifiers with a review-based dataset.

3. To test and develop the most effective ones which can recommend products to customers better.

4. To evaluate the proposed models.

5. To offer recommendations for developing the proposed models.

In the long run, this model can be extended to recommend more efficiently selective products to meet customer satisfaction.

1.4 Problem Statement

The aim of this project is to develop a system that recommends products through ML and NLP approaches, but due to limitations such as data availability and resources, the chance of the project is limited to creating a recommendation system for mobile products only. The target is to understand the potential and limitations of using ML and NLP techniques in product recommendation systems and to provide insights into future research in this area.

The main research challenge is how, despite constraints like data availability and resource availability, ML and NLP methodologies may be applied to construct an efficient recommendation system for mobile products. Although we used the Amazon dataset for mobile phones, it was quite challenging to track and implement this big data on limited configured devices. Other than that, the models we chose for our implementation have never worked before on such a big dataset and analyzing sentiment together which was quite challenging for us to accomplish in a shorter time frame. Finally, the limited resources and published papers on mobile phone recommendation systems were also challenging for us to gather information and knowledge on this certain topic.

Chapter 2

Related Work

The sales and marketing sector has drawn a lot of interest from people all over the world. As ML and NLP are growing in their field, it must be noted that these have a massive impact on opinion mining and suggest the best course of action to respond to. The task of ML and NLP is to analyze data and then automate the data with the analytical model building so researchers can relate ML and NLP methods to product recommendations that can help customers choose the right product according to their needs.

As a result, the idea of recommendation systems was prompted in the late 1990s, according to Zeinab Shahbazi et al. [19] Various e-commerce companies like Amazon, eBay, Shopify, etc. which are involved with recommending specific items/products to their clients are currently focusing their attention on the industry. According to Dietmar Jannach and others [61], the recommendation system may take the form of a variety of info-filtering techniques that assess user activity on market datasets and forecast user preferences. User behavior includes press releases, customer reviews of the things they have purchased, opinions, reposting, cart information, etc. According to Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl [23], the primary goal of this strategy is to help people make decisions about the purchase of bound objects when there is a lack of readily available information. Since the nineties, the research community has given out several RS in a variety of fields, including those for books, films, and other products. This system has been divided into five distinct classes by Robert Burke [5], including

1. Collective filtration

2. Content-based recommendations

3. Information-based recommendation

4. Hybrid recommendation and

5. Demographic recommendation [5].

According to Ortega, F. et al. [25], the collaborative filtering recommender system is thought to be the most effective strategy because it bases its recommendations on ratings from active users. According to Hernando, A., and others' [28] reasoning based on collaborative filtering, such recommender structures employ a scoring matrix "M" where each user provides records roughly how much he loves a few things. The recommender system no longer keeps track of a customer's or an item's capabilities. By using the rankings that customers have already established, recommender systems of this type can tap into the preferences of their users.

Content-based recommendation systems, or those that suggest a thing to an individual user on a description of the thing and a profile of the user's preferences, were justified by Michael J. Pazzani and Daniel Billsus [34]. The user-recommendable items are frequently kept in a database table, which displays a straightforward database containing records describing three eateries.

According to Michael J. Pazzani and Daniel Billsus [34], who was also quoted, a restaurant listing and recommendation website will be made mandatory online using the database shown in Table. This is an example of structured data in which each object is defined by means of the same set of attributes, there are a small number of features, and there is a recognized list of variables that the features may also have. In this situation, a consumer profile that can be examined using various machine-learning techniques, or a menu interface can be quickly developed to allow a customer to generate a profile. The next section of this bankruptcy examines various methods for creating a fact-based customer profile.

According to Robin Burke [13], the third type of recommender system is the one that formulates a knowledge-based recommendation to determine whether the items satisfy the user's requirements using data about users and products. The PersonalLogic recommender system provides a dialog that essentially leads the user to a product feature discrimination tree. Others have completed this job using quantitative decision-support technologies (Bhargava, Sridhar Herrick, 1999).

Hybrid filtering, which combines two or more filtering approaches in various ways to increase the performance and accuracy of recommender systems, was described by G. Murat et al [3] and Mobasher B. and co.[17]. These methods combine two or more filtering strategies in order to take advantage of their advantages while balancing out their disadvantages [12].

The classification of the hybrid system into weighted hybrid, mixed hybrid, switching hybrid, feature-combination hybrid, cascade hybrid, feature-augmented hybrid, and meta-level hybrid was made by Mican D. and Tomai N. [15]. Last but not least, M. Sridevi Anurag et al. [20] explored a method for generating recommendations based on user demographic characteristics. It recommends movies to viewers by using their demographic information and reasoning with them primarily based on their attributes. It is simple to deploy and does not require user ratings, in contrast to collaborative filtering and content-based recommender systems.

Using ML models, Pang, Lee, and Vaithyanathan were the first to develop a sentiment categorization system for movie reviews in 2002. The Naive Bayes, Max Entropy, and Support Vector Machine models also performed well when they were applied to unigrams and bigrams of data. With an accuracy percentage of 82.9%, SVM was the most accurate of these models [43].

Mullen and Collier used product review datasets like jewelry, shoes, and clothes to categorize sentiment in a 2004 publication [7]. They contrasted hybrid SVM, Naive Bayes, LR, and decision trees with feature extraction methods based on Lemmas and Osgood's theory [9]. In their analysis, SVM generated the best outcomes including an accuracy of 86.6%.

Lilleberg, Zhu, and Zhang compared SVM-based TF-IDF and Word2vec feature extractions in a 2015 publication. Additionally, they compared the classification outcomes with and without stopwords. They observed an accuracy of 88% for SVM with TF-IDF and without stopwords [1].

In a 2017 article, Elmurngi and Gherbi suggested utilizing SVM and sentiment analysis to spot bogus movie reviews. On a corpus with and without stopwords, they evaluated the performance of SVMs, Naive Bayes, decision trees, and KNN classifications. Both times, SVM prevailed, earning a score of 5. In a Competitive Evaluation of Sentiment Analysis and Product Reviews, Nguyen et al. Used ML and Lexicon-Based Approaches [32], and accuracy rates of 81.35 and 81.35 percent, respectively, were achieved.

A distinct piece was published by Ramadhan et al. TF-IDF feature extraction and logistic regression was applied to a Twitter social media dataset to analyze sentiment. According to reports, the categorization accuracy was very nearly 83% [41].

Using an Amazon product review dataset and the SVM, TF-IDF model, and Next Word Negation, Das and Chakraborty carried out an experiment in 2018 with an accuracy of 88.86% [9]. In a 2018 publication, Bhavitha, Rodrigues, and Chiplunkar also conducted a comparison study of various lexicon-based methods, sentiment analysis on movie reviews, and ML techniques. They reported an accuracy of 74% for the SentiWordNet method and an accuracy of 86.40% for the SVM method[37]. Gradient Boosting ML outperformed SVM, Naive Bayes, and neural networks for both balanced and imbalanced data sets, according to Athanasiou, who employed it for sentiment analysis in the same year. The Gradient Boosting type of ML ranked the best, with an accuracy of 88.20% [45].

Numerous researchers have concentrated on reading product reviews for the goal of making judgments. A fresh approach was put up by Garca-Moya and coworkers [38] for extracting opinions and product features out of a set of free-text consumer reviews of a good or service. To increase the proposal's adaptability to domains and languages, this was done.

Furthermore, Singla et al.[26] categorized text as positive, neutral, or negative, despite taking into account a variety of emotions in addition to the conventional positive, negative, and neutral categories (for example, wrath, expectation, contempt, fear, pleasure, sorrow, surprise, and faith. Paknejad [44] looked into various machine-learning methods to find the most effective solutions for sentiment categorization issues in online reviews using Amazon product reviews.

Abbasi and others [49] attained an accuracy between 85%-88% using SVM classifiers for SA using a mixture of univariate and multivariate feature-picking strategies after implementing the chi-squared method to choose the pertinent textual features. The classifier's performance was helped by a network-based feature selection technique called feature relation networks (FRNs).

Saura and others [16] used sentiment analysis with an SVM to identify key UGC factors that helped establish successful start-ups. Using the polarity sentiment, this method was used to select startup topics.

According to Haddi et al, Several methods have been used to choose features, among which are syntactic and based on the word's position in the syntactic tree (such as adjectives); Some features are univariate depending on how each feature relates to a particular category; and some are multidimensional and based on subsets of attributes [53].

Arak and co. [56] utilized methods that break reviews down into segments that evaluate the specific features of particular products (for instance, a digital camera's image quality and battery life). The authors then made a substantial contribution by applying methods from the literature on econometrics, specifically the idea of hedonic regression. Existing feature selection approaches, according to Chi et al. [21], compute feature scores simply using statistics from training data or by altering a specific feature metric formula to incorporate test info that cannot be applied to various types of feature metrics. They suggested merging the two approaches both the training dataset and the feature metric formula.

Mars and Gouider [39] presented a big data architecture for decision-making, data analysis, and gathering user feedback on product improvements. The techniques employed in the architecture include big data, NLP, and ML. To locate the features, they employed an ontology that encompasses both common mobile phone features and attributes as well as other particular technical terminology for electronic items.

Then, using the MapReduce programming model Hu, W. et al and Singh, P. et al, extracted feature opinions. Numerous studies in this area [42] [18] have demonstrated thatThe most used technique for feature weighting is feature frequency.

Wang and others [29] provided a method for lowering the number of features by removing those that aren't important to the class by weighing the importance of the clustered features towards the class.

Zhou and co. [51] suggested retaining terms with strong class differentiating power and removing redundant features, based on the document frequency and segmentation term frequency.

At the moment, approaches based on deep learning have demonstrated exceptionally high performance on a wide range of NLP tasks. Young, T. et al looked at meaningful deep learning paradigms and NLP task-specific approaches in reference [27]. Additionally, they provided an evolution practice. Concerning SA, recent studies by Socher and others [52] have emphasized the significance of extensive phrases and the necessity of supervised training, evaluating resources, and a greater power model in these instances.

Besides, Cotter P and Smyth B. [31] demonstrated in their research, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Correlation are the statistical metrics that gauge the efficacy of recommendation systems.

The most well-known and frequently employed measure of the recommendation's departure from the user's particular value is the MAE. It's evaluated as

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - x_i| \tag{2.1}$$

It is the total number of ratings on the item set, ru,i is the actual rating for user u on item I and 1 is the predicted rating for user u on item i. The recommendation engine forecasts user ratings more precisely the lower the MAE. The Root Mean Square Error (RMSE) is given by Cotter et al. [31] as

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2} \tag{2.2}$$

Root Mean Square Error (RMSE) puts more emphasis on the larger absolute error and the lower the RMSE is, the better the recommendation accuracy.

B.M. Sarwar et al [4] discussed decision support accuracy metrics using Precision, Recall, and F-measure. They are computed as

$$Precision = \frac{Correctlyrecommendeditems}{Totalrecommendeditems} \tag{2.3}$$

$$Precision = \frac{Correctlyrecommendeditems}{Totalusefulrecommendeditems} \tag{2.4}$$

Precision and recall are combined into a single metric by the F-measure. A comparison of algorithms and data sets is made very easy and clear by the final value.

$$F - measure = \frac{2PR}{P + R} \tag{2.5}$$

Akshat Bakliwal, Piyush Arora, Ankit Patil, and Vasudeva Varma [2] believe, "What people think and feel" is the most valid motto in business to hold on to and reach the peak. Dropping reviews on the internet is increasing significantly. Therefore, checking reviews have been of the utmost use for customers to know about the product and for manufacturers to improve their product accordingly. So in their paper, they used basic NLP techniques that include n-gram, and POS-tagged n-gram. They focused on sentiment polarity which is positive and negative. They experimented and used several ML algorithms like Naive Bayes (NB), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM) to have a comparative study of the performance of their model whether it is positive sentiment or negative. As a result, their approach was successfully accurate. The maximum accuracy was 81.15 with SVM and 80.15 with the ML algorithm applied in the product review dataset.

Tomas PRANCKEVIČIUS, Virginijus MARCINKEVIČIUS [22] vastly compared ML, Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, Logistic Regression, Apache Spark, NLP Logistic Regression on data classifying for review processing. Firstly, they extracted the data from Amazon. Then, they prepared the review data by tokenization, removing all stop words, making all capital letters small, and stemming. After that, they applied the n-gram method as a sequence of written words of length n. Finally, they calculated the accuracy by using the formula for multi-class classification can be presented as follow (Sokolova and Lapalme, 2009):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

Here, TP is true positive classification examples, FP is false positive ones, FN is false negative ones, and TN is true negative ones, l is the number of classes. Moreover, the multi-class classification method for product reviews has achieved the highest (min 32.43%, max 58.50%) classification accuracy, and Decision Tree has got the lowest average accuracy values (min in trigram: 24.10%, max in uni/bi/tri-gram: 34.58%). We emphasize in this section's conclusion that, in our paper, we will not only focus just on the product recommendation system but also pay close attention to sentiment analysis, which will help us achieve more fruitful outcomes.

Chapter 3

Data and Preliminary Analysis

The dataset that we used for the implementation of our model is for unlocked mobile phones collected from Amazon. This dataset consists of product title, brand, price, rating, and review text. Our ability to purchase goods online has been transformed by mobile devices, which put all the information at our fingertips. More and more customers will turn to other customers for product information instead of the seller's information as information access gets easier. Examples of this kind of information include reviews and ratings left by customers, which have already influenced many customers' purchasing decisions. Customers may make educated decisions and feel confident about them thanks to the transparent system created by the review and ratings platforms offered by eCommerce businesses. Product reviews may be found in abundance on Amazon.com, and its review system is available through all available channels and presents reviews in an intuitive layout. The product reviewer gives the item a rating between 1 and 5, along with their own opinion based on their whole experience. To get the final product rating, the mean value from all the ratings is determined. By allowing others to vote on whether or not a review is useful, both the review and the reviewer gain credibility. For this study, we analyzed over 400 000 reviews of unlocked cell phones purchased from Amazon.com in order to gain an understanding of how reviews, ratings, and prices relate to one another. This dataset implements the model to satisfy our goals of performing exploratory research of rating and reviews. Identifying relationships between price and number of reviews. The majority of reviewers have awarded unlocked mobile phones 4-star and 3-star ratings, according to Amazon's product review platform. The reviews are roughly 230 characters long on average. Additionally, we found that reviews with greater detail tend to be more beneficial as well as a positive relationship between price and rating. Positive sentiment is prominent in the reviews, according to sentiment analysis, and the emotions "trust," "anticipation," and "joy" score the most.

The sentiment analyzer model requires an NLP toolkit for a better understanding of how the language may work in different situations. Next, the python library beautiful soup is used for filtering the HTML tags during the implementation. The data is divided into training and testing data using cross-validation. The labeled data is then read to use the read function after the dataset has been applied. The product or brand and brand name are then identified once the data's number of rows and columns is displayed. The data frame is created by passing the dataset object to be converted to a series for data cleaning and preprocessing. Next, it is checked whether there are any null values are not after the data has been cleaned. It is essential to use reset the index to drop the existing index and replace it with the new index. Using the pivot table the approach will show the brand name, rating, and review votes.

Using the matplotlib library we determine the price vs rating graph, Price vs Review Votes graph, and Rating vs Review Votes graph for the dataset.

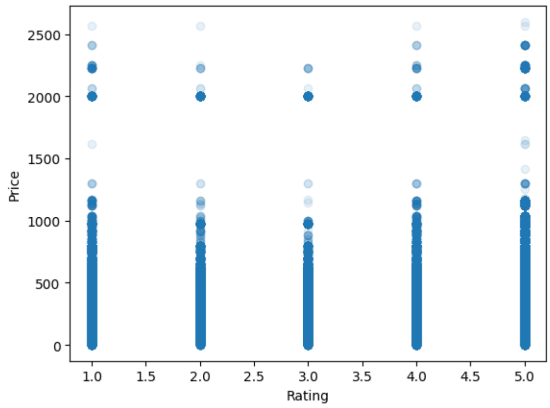

Figure 3.1: Price vs Rating graph

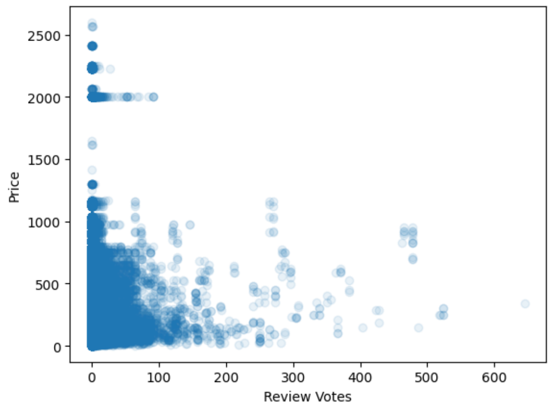

Figure 3.2: Price vs Review Votes graph

Next, the correlation matrix is performed for sorting the values of Rating and Price in ascending order which initializes the name and data type. Next, the data is converted within the review column to string data types. The next phase starts with training the data for analyzing the sentiments. The train data is taken for example first 20,000 rows for calculation of sentiment analysis. Next, the scatter intensity plot of sentiments is identified for the train data for calculating the sentiment values. Through this, the top mobile phone models are generated and next the process of sentiment analysis is done.

3.1 Pre-processing

Preprocessing in ML is changing raw data so the machine can use it. For certain ML models to function better, dimension reduction, relevant data identification, and

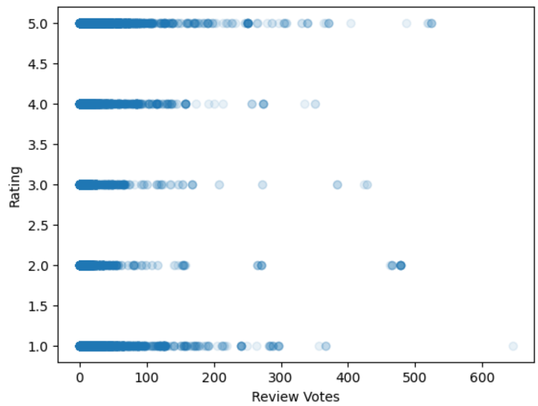

Figure 3.3: Rating vs Review Votes graph

performance improvement are all required. Data must be transformed or encoded in order for a computer to quickly parse it. Data preprocessing is a crucial stage in data mining since it helps to spot errors, outliers, noise, and missing important variables. Lacking data preprocessing in data science, these data errors would continue, reducing the standard of data mining.

This function also requires removing the nonletters. Along with converting every letter to lowercase. Removing and stemming stop words is also required for preprocessing. Publications and some verbs are frequently regarded as stop words because they make it difficult to understand the context or true meaning of a statement. These are phrases that can be removed without impairing the output of the trained model.

3.2 Data Exploration

Even though it might not always reveal every subtle aspect, data exploration helps provide a more complete picture of specific trends or topics to explore. In order to select the best model or algorithm for the subsequent stages of data analysis, users review data using both manual and automated methods. In order to quickly identify connections between various data variables and data structures, check for outliers, and offer data values that can highlight patterns or interesting locations, automated exploration software or ML algorithms are often used. That is why data exploration was required for our implementation as we are dealing with large data.

3.3 Data Analysis

From our research, we found no significant dataset containing all the necessities for reviewing mobile phones, for example, brand, price, reviews, ratings, statistics, model, etc. However, we tend to find one essential dataset named Amazon Reviews: Unlocked mobile phones review, which consists of product name, brand name, price, rating, reviews, and review votes. As we are using big data for our research purpose the data needs to be analyzed first. For that, the reviews should go through analysis first. And for that, we divide the reviews into three segments. First, the review analysis consists of the overall word frequencies.

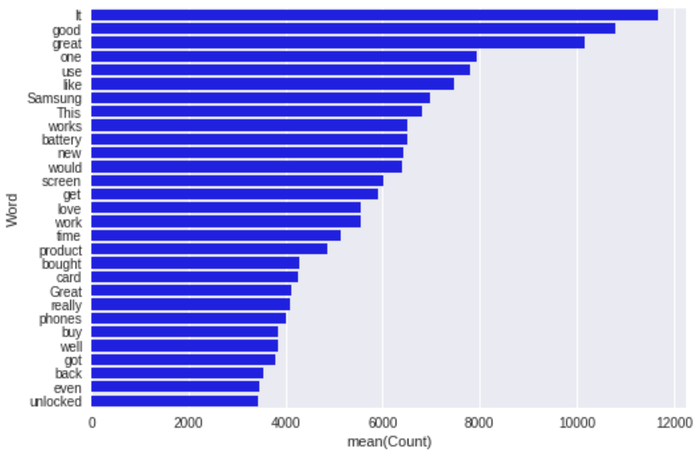

Figure 3.4: Overall word frequencies

This figure clearly shows the neutral words used in the reviews part of the dataset. These are the words with which the negative and positive words are being compared for getting the sentiment analysis done. Another category for the data analysis is the negative reviews.

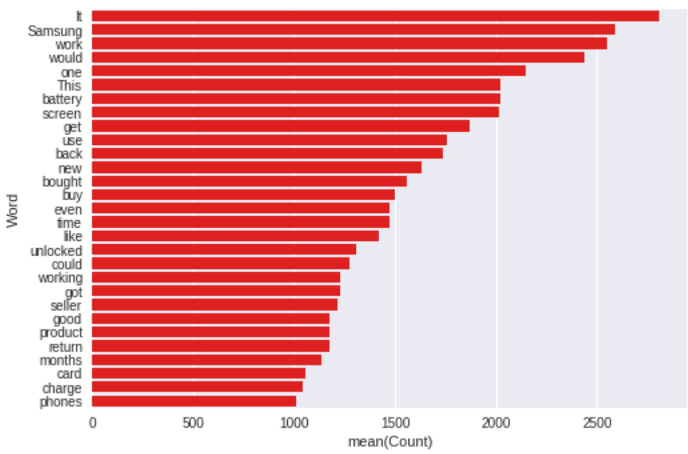

Figure 3.5: Negative word frequencies

In this figure, the negative word frequencies are shown from the dataset. Here, we can see that few words are used negatively in terms of not approaching the brand or the model. These frequencies are important for determining the sentiment values. Finally, we required the positive word frequencies.

In this figure, the positive word frequencies are shown from the dataset. Here, we can see words like good, great, love, etc. are used positively in terms of approaching the model or the brand of the phone. Through these analyses, it becomes easier to implement various models in calculating the sentiment values.

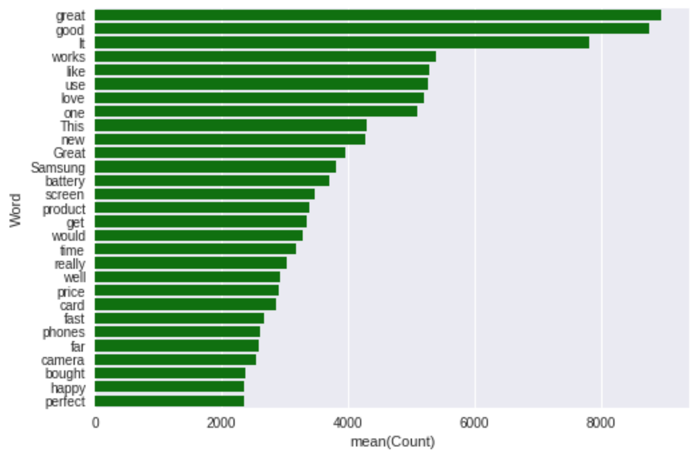

Figure 3.6: Positive word frequencies

3.4 Data Labeling

The majority of actual ML models use supervised learning to identify one input to one output. For supervised approaches to work, there must be a labeled data set that the model can learn from and utilize to make the appropriate decisions. Asking for feedback on a particular set of unlabeled data is a typical starting point for data labeling. For example, labelers could need to tag every image in a series with the word true for "does the image contain a bird." The labeling could be as straightforward as a yes/no inquiry or as intricate as pinpointing specific pixels in the bird's image. The ML model uses labels supplied by humans in a procedure known as "model training" to find the underlying patterns.

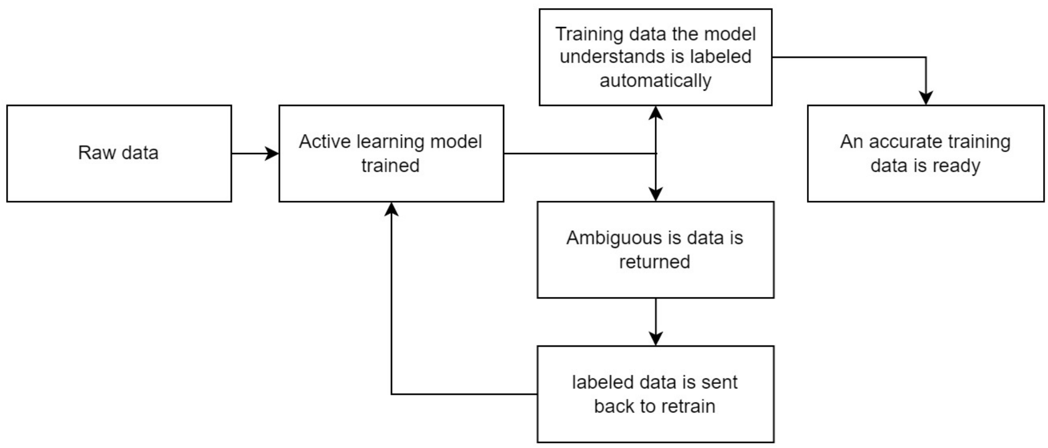

Figure 3.7: Data Labeling process

In the figure the process shows how the raw data is trained from human-labeled data which sequentially trains the model automatically and along with the ambiguous data is sent to labels. This withdrawn data is looped to retrain and improve the learning models. Finally, this process generates an accurate training dataset.

	Product Name	Brand Name	Price	Rating	Reviews	Votes
0	"CLEAR CLEAN ESN"	Samsung	199.99	5	I feel so LUCKY to have found this used (phone...	1.0
1	"CLEAR CLEAN ESN"	Samsung	199.99	4	nice phone, nice upgrade from my pantach revu...	0.0
2	"CLEAR CLEAN ESN"	Samsung	199.99	5	very pleased	0.0
3	"CLEAR CLEAN ESN"	Samsung	199.99	4	it works good but it goes slow sometimes	0.0
4	"CLEAR CLEAN ESN"	Samsung	199.99	4	great phone to replace my lost phone	0.0
5	"CLEAR CLEAN ESN"	Samsung	199.99	1	i already had a phone with problems. now i know...	0.0
6	"CLEAR CLEAN ESN"	Samsung	199.99	2	I originally was using the samsung galaxy S2 Galaxy...	0.0
7	"CLEAR CLEAN ESN"	Samsung	199.99	3	I originally was using the samsung galaxy S2 Galaxy...	0.0
8	"CLEAR CLEAN ESN"	Samsung	199.99	2	I originally was using the samsung galaxy S2 Galaxy...	0.0
9	"CLEAR CLEAN ESN"	Samsung	199.99	3	I originally was using the samsung galaxy S2 Galaxy...	0.0

Table 3.1: Amazon Dataset Mobile Phones

	Product Name	Brand Name	Price	Rating	Reviews	Votes
0	"CLEAR CLEAN ESN"	Samsung	199.99	5	I feel so LUCKY to have found this used (phone...	1.0
1	"CLEAR CLEAN ESN"	Samsung	199.99	4	nice phone, nice upgrade from my pantach revu...	0.0
2	"CLEAR CLEAN ESN"	Samsung	199.99	5	very pleased	0.0
3	"CLEAR CLEAN ESN"	Samsung	199.99	4	it works good but it goes slow sometimes	0.0
4	"CLEAR CLEAN ESN"	Samsung	199.99	4	great phone to replace my lost phone	0.0
5	"CLEAR CLEAN ESN"	Samsung	199.99	1	i already had a phone with problems. now i know...	0.0
6	"CLEAR CLEAN ESN"	Samsung	199.99	2	I originally was using the samsung galaxy S2 Galaxy...	0.0
7	"CLEAR CLEAN ESN"	Samsung	199.99	3	I originally was using the samsung galaxy S2 Galaxy...	0.0
8	"CLEAR CLEAN ESN"	Samsung	199.99	2	I originally was using the samsung galaxy S2 Galaxy...	0.0
9	"CLEAR CLEAN ESN"	Samsung	199.99	3	I originally was using the samsung galaxy S2 Galaxy...	0.0

Table 3.2: Amazon Dataset Mobile Phones after data cleaning

3.5 Data Preprocessing

Data preprocessing is a data mining method that turns unstructured data into structured data that is more trustworthy, usable, and well-organized for ML models. The ability to combine data from several sources, complete data gaps, and identify and address data inconsistencies all contribute to better results. Four phases make up our data preprocessing methodology.

1. Remove duplicate reviews

2. Remove characters, words, and punctuation that are difficult to understand

3. Eliminating stop words

4. Stemming

Our textual data had the appearance seen in the following figure before the preparation procedure.

Here in this table, we can see that the repetition of the mobile phone review occurred a couple of times. Here feature scaling cleans and reduces the repeated data from the dataset.

Chapter 4

Methodology

4.1 Model Workplan

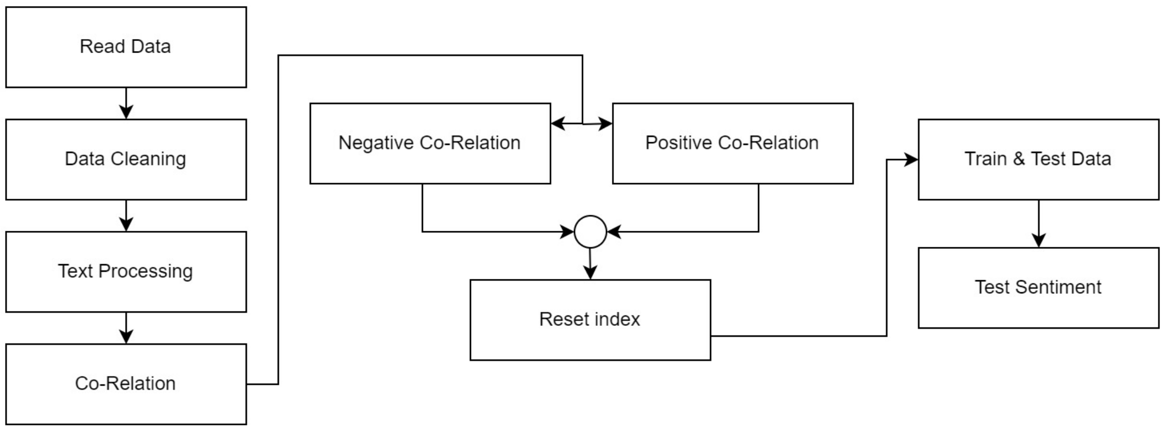

Figure 4.1: Sentiment Analyzer workflow diagram

Here, the sentiment analyzer model reads the data first. Next, send for the data cleaning process along with text processing and correlation. Next, the correlation is divided into negative and positive which merge and prepare the reset index. Finally, the reset index is ready for the training and testing phase. And then the data is ready for analyzing sentiments.

Here, the model initially imports the data and divides the data into two separate categories. Positive reviews and negative reviews. Next, the process merges the data separately and the visualization process is done. Data cleansing is an essential step here which removes unnecessary words. Next, the data is vectorized and the model is built. Next, the training and testing process takes place which finally saves the model and the data is ready for generating training and testing data results.

4.2 Feature extraction

ML algorithms and deep learning models cannot work on the raw text directly. As a result, in order to make our models read the data, we have to transform our data into numerical form. So, Feature Extraction is one of the essential parts for better

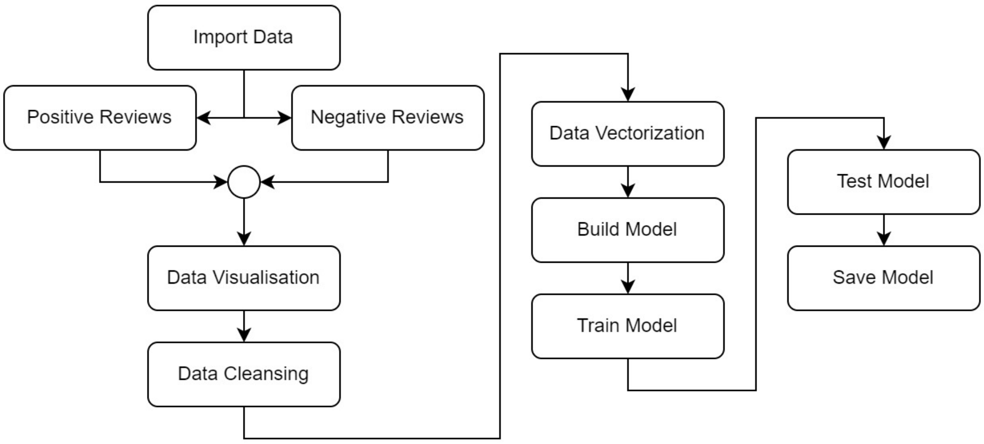

Figure 4.2: Workflow diagram

impact on applying ML and deep learning models. We have used several vectors for Feature Extraction.

4.2.1 TF-IDF

The TF-IDF tool determines how often a word appears in a document. TF-IDF Vectorizer generates the sparse matrix of numerical features so that the classifiers can understand numerical values [33].

$$TF(w,d) = \frac{occurenceofwindocumentd}{totalnumberofwordsindocumentd} \tag{4.1}$$

Here, TF makes a vocabulary of unique words and calculates the ratio of the word in a text. However, TF fails to find out the importance of the words. That is why the IDF is used. IDF provides the weightage to each word based on its frequency.

$$IDF(w,d) = ln\frac{TotalnumberofdocumentsincorpusD}{numberofdocumentscontainingw} \tag{4.2}$$

4.3 Modeling and model specification

Naive Bayes

Text categorization, spam detection, and sentiment analysis are a few classification problems for which naive Bayes is frequently utilized. Text is frequently classified in NLP into many categories, such as positive, negative, or neutral mood. Naive Bayes has a rather straightforward algorithm. The approach determines the prior probability of every class label, which is the likelihood that the class label will appear in the training data, given a collection of input data and a set of class labels. The probability of each character given each class label, or the likelihood of the characteristic occurring given the class label being known, is then calculated. The input is then classified into the class with the highest probability using Bayes' theorem to determine the likelihood of each class label based on the features [64].

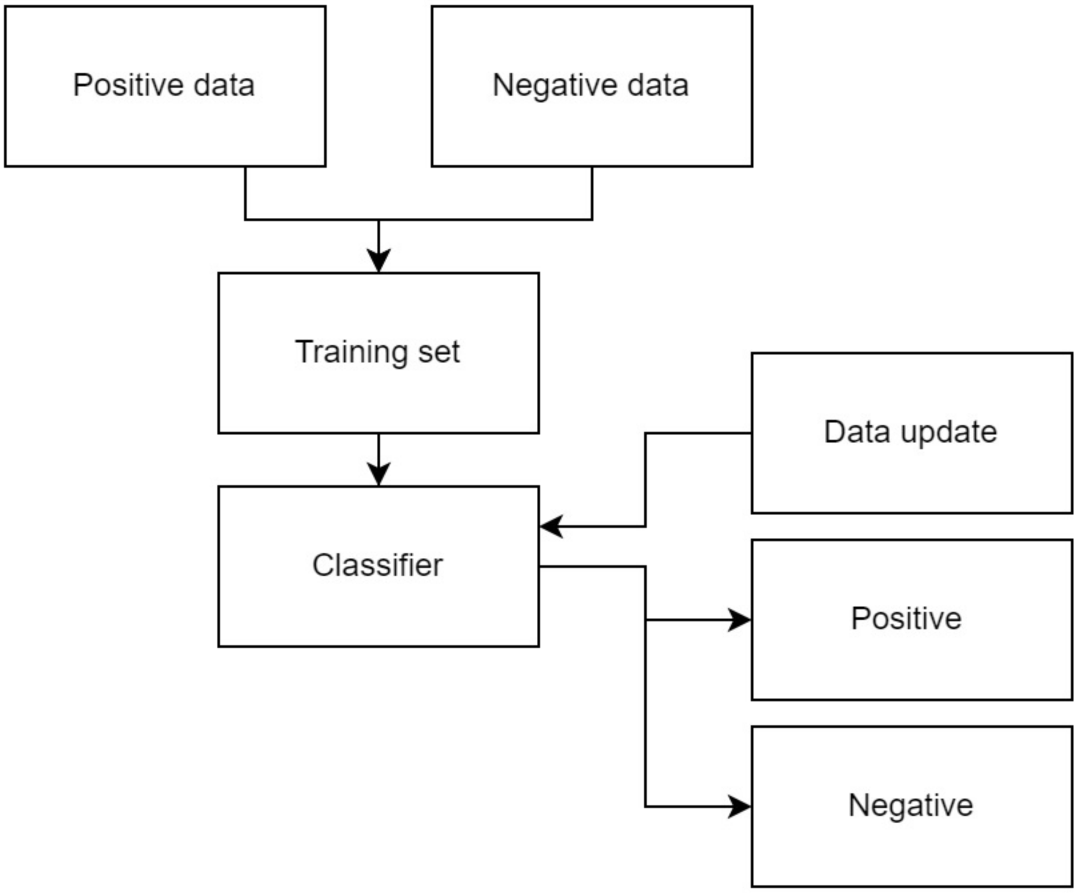

Figure 4.3: Naive Bayes workflow

It assists in predicting the class of the ambiguous data sample by estimating the likelihood of class in the input data. It is a potent classification method that works well with large data sets. The probability density for each class is determined using the Bayes Theorem formula.

4.3.1 Random Forest Classifier

In ML, Random Forest is used to training feature importance in a supervised learning scenario. It can be used to determine the most important features that are done for the predictions made by the model. The algorithm works by training multiple decision trees on different subsets of the data and features and then aggregating the results.

A subset of the data is used to train decision trees in the random forest, and only a random subset of the characteristics is taken into account at each split. This means that each decision tree is only exposed to a small subset of the data and features, which helps to lessen the number overfitting. After training, the algorithm computes a feature importance value for each feature based on how often it is used to split the decision trees, and how much it contributes to reducing the impurity of the splits[58].

In NLP, Random Forest is used to training features for text classification tasks. The algorithm is used to identify the most important words or n-grams in a text, and then use these features to train a Random Forest classifier. The algorithm can handle large, sparse data sets and high-dimensional feature spaces, which are common in NLP tasks. Additionally, it may be used to calculate feature importance, which is helpful for identifying underlying trends in text data [40].

As an illustration, Random Forest can be used in sentiment analysis to train attributes based on the frequency of specific words or n-grams in a text and then use these features to categorize the content as positive, negative, or neutral. The most crucial keywords or n-grams that are most closely related to the sentiment of the text can be found by using a Random Forest to train the features [55].

Random forest is a suitable ensemble method in the case of Binary classification problems. Because randomization ensures that different features such as sparse and dense will be used as primary decision nodes in individual trees. Not but least, RF/decision trees may be a great approach for inspecting features and their structure well. Furthermore, the random forest creates a multiclass classification model to categorize the dataset, which provides fair accuracy.

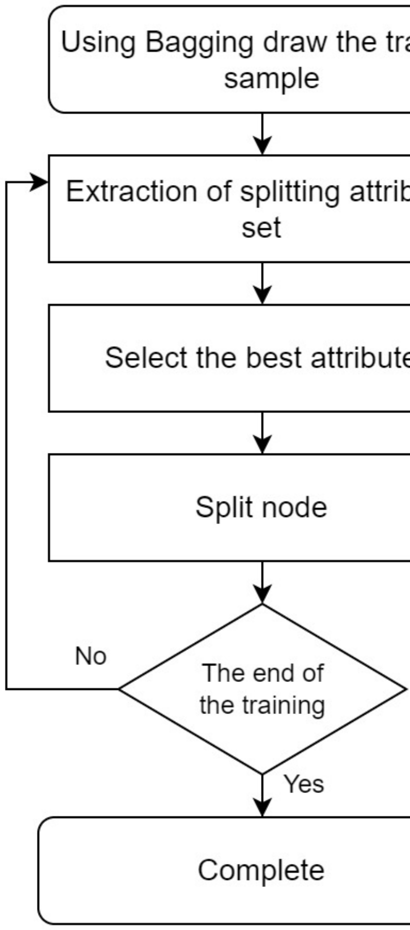

Figure 4.4: Random forest workflow

4.3.2 SGD Classifier

For linear classification applications including text categorization, sentiment analysis, and spam detection, SGD Classifier is employed. It is particularly useful when the data is large, as it can handle large datasets more efficiently than other optimization algorithms, such as batch gradient descent [54]. Because of having a large dataset which is almost over 4,13,000, we have used this SGD classification for optimization for the characteristic of having efficient handling of large datasets quickly. Moreover, It is also relatively simple to implement, as it requires only a small number of hyperparameters to be set. Additionally, it can be used with a variety of different loss functions, such as logistic loss, hinge loss, and squared loss, which makes it a versatile algorithm for linear classification tasks.

4.3.3 Gradient Boosting Classifier

A prediction model is produced via gradient boosting as a collection of ineffective prediction models, frequently decision trees. It builds the model incrementally, like other boosting methods, and generalizes them by allowing the evaluation of any variational loss function. By feeding the gradient boosting method the output of other NLP models, such as language models, it has been used to enhance the performance of other models [62].

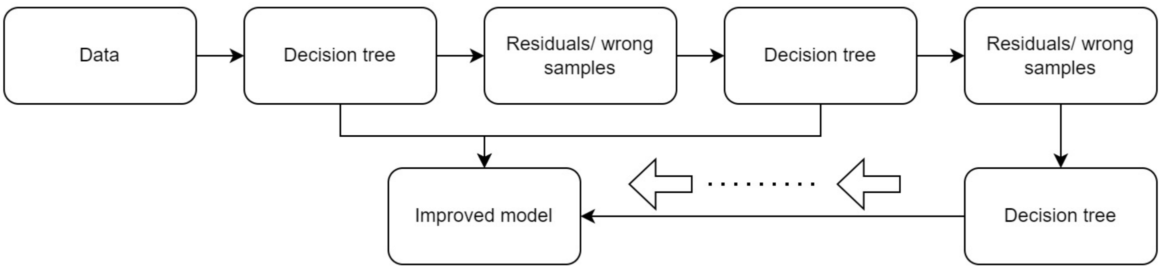

Figure 4.5: Gradient boosting classifier workflow diagram

Every tree strives to lessen the flaws of the preceding tree. While boosting trees is not good learners, by combining more trees one after another and having each one focus on the errors committed by the one before it, boosting develops into an incredibly accurate and effective model. In contrast to bagging, boosting does not employ bootstrap sampling. The basic dataset is modified to fit the brand-new tree each time a new tree is added.

4.3.4 NBSVM

NBSVM has been used to estimate the probability of each word belonging to a certain class which was first created by implementing naive Bayes Classification. These probabilities are then used as features for an SVM classifier, which is trained to distinguish between the different classes based on these word probabilities. The NBSVM algorithm has been used to first train a Naive Bayes classifier on the training data, and then use the learned word probabilities to construct a feature representation for each training instance. The SVM is then trained on this feature

	precision	recall	f1-score	support
1	0.65	0.84	0.73	5247
2	0.78	0.10	0.18	1822
3	0.67	0.11	0.19	2280
4	0.48	0.15	0.23	4447
5	0.73	0.96	0.83	16326
accuracy			0.70	30122
macro avg	0.66	0.43	0.43	30122
weighted avg	0.68	0.70	0.64	30122

Table 4.1: Classification report

representation to classify the instances [48]. This method has been proven to be very successful at classifying texts, and it is reasonably easy to use.

4.3.5 Classification report

A classification report is a machine learning performance statistic. This method displays the precision, recall, F1 Score, and support of the trained classification model. It's one of the standards used to gauge how well classification ML models perform. It shows a model's recall, F1 score, accuracy, and support. It makes it easier for us to understand the overall effectiveness of our trained model.

4.3.6 MLP

A convolutional artificial neural network model called a multilayer perceptron (MLP) converts collections of input data into a collection of meaningful outputs. Each layer of nodes in an MLP in a directed graph is completely connected to the layer below it. All nodes—aside from the input nodes—are processing elements or neurons with nonlinear activation functions. Although they may be helpful, they are not the ideal deep models to utilize with text-based unstructured data [36]. The Tf-Idf Matrix will receive the MLP application. We'll utilize the Tf-Idf vectorizer to map vocabulary words or phrases to a corresponding vector of real numbers that can then be used to find word predictions and word similarity/semantics. Here, the model calculates its preprocessing by dividing by the max value and subtracting the mean value. Next, we use the categorical entropy for the loss and RMSprop for the optimizer. Optimizers are tools or processes that change a neural network's weights and learning rates to reduce losses. The optimizers you employ determine how you should modify the parameters or learning levels of a neural network to minimize losses. Finally, we train this model for 50 epochs.

4.3.7 LSTM

The only difference between an RNN's forward pass and an MLP's is that an RNN uses the outputs from hidden layers as inputs for the same layer. This implies that the hidden layer's input consists of the outputs from its previous step in time and external information.

LSTMs often referred to as networks, are a special class of RNNs that have the ability to recognize long-term dependencies. Thus, since LSTM's "memory" may be able to store sentence dependencies, it can be especially useful in text mining challenges. Next, we will vectorize the text samples into a 2D integer tensor using a tokenizer.

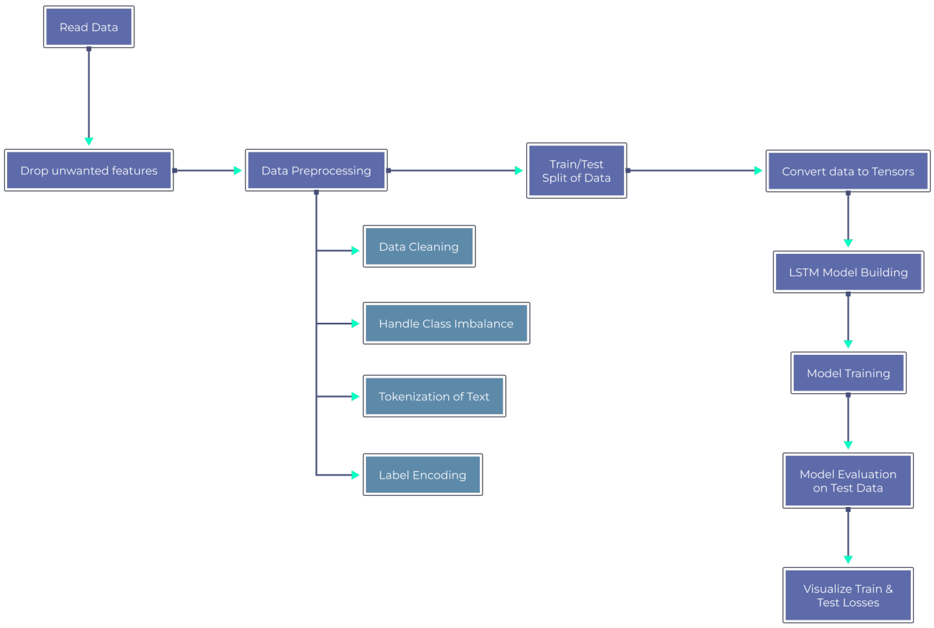

Figure 4.6: LSTM Architecture

This is the workflow of the LSTM model which initially reads the data along with removing the unwanted features. Next, the data is sent for data preprocessing. The process is split into 4 categories. They are data cleaning, handling class imbalance, tokenization of text, and label encoding. After the data preprocessing the data is next sent to the train/test split of data. The data is then converted to tensors. Next, the model is built through training and model evaluation of test data. Lastly, visualize the train and test losses.

Model Summary

After setting out the max features and embedding features the validation split and max length must be initialized. Vectorizing the text sample integers is done. While building the model we must add dropout, LSTM, and dense layers. The activation layer is also needed to prevent linearity. Next, we train this model for 30 epochs.

4.3.8 CNN

Convolutional neural networks are a part of deep neural networks that are often employed in deep learning to assess visual vision. On the other hand, convolutional neural networks have lately gained popularity in the field of NLP for handling problems involving Sentence Segmentation, Text Categorization, Sentiment Analysis, Text Categorization, Language Processing, and Answer Relations.

The ConvNet primarily performs four tasks: Pooling with Convolution Non-Linearity (ReLU) or Classification using Subsamples (Fully Connected Layer). Convolution entails applying various filters to our input (here, we choose to utilize 250 filters) (here text). The convolution step's primary goal is to extract characteristics from the input [57]. Multiple convolutions (mathematical convolutions) can be applied to our input thanks to the various filters. After applying our convolved layer, we thus get N (N: number of layers) convolved input. After the convolution layer, we utilize Relu to add nonlinearity [57]. The following pooling, we have. There are various types of spatial pooling, including maximum, average, and sum. The final stage is identical to the MLP.

Model summary

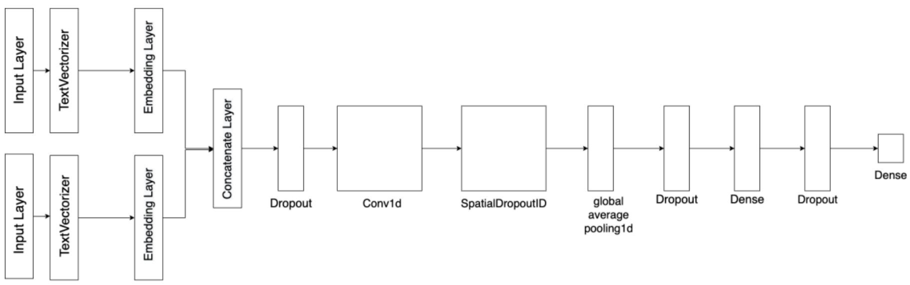

Figure 4.7: CNN Architecture

We used two input layers along with two TextVectorizers and two embedding layers. Then they are put into a concatenated layer and after that, a dropout was performed before putting them into a conv1D layer. After that spatial dropout, global average pooling, and dropout, dense, and dropout is performed sequentially. Lastly, the dense layer shows the result.

We have trained this CNN model for 30 epochs. We also built the model adding embedding features and dropout. We added the convolution 1D which will learn the filters. And word group filters of size filter length.

4.4 Information Collection

Filtering, prioritizing, and properly distributing critical data on the Internet is required to deal with the issue of information overload, which has potentially become an issue for many Internet users due to the abundance of possibilities. Recommender systems, which filter through a huge amount of constantly created data to give clients personalized content and services, overcome this problem. There may be a problem with fresh information that prevents timely access to relevant items on the Internet due to the exponential development in the quantity of digital data available and the number of Internet users. Therefore, the necessity for recommender systems is greater than ever [24]. Recommender systems are information filtering systems that solve the problem of information overload by choosing significant information pieces from a large amount of dynamically created material depending on the user's preferences, activities, or records about the item [24]. A recommender system can assess whether a specific user will favor an item based on the user's profile.

ML, an area of artificial intelligence, is concerned with the creation of algorithms that can really scan massive information, discover recurrent patterns and connections across several variables, and produce mathematical models illuminating them [47]. Since ML systems and reinforcement learning applications using these algorithms may genuinely improve their abilities through experience, the word "learning" in its name is not a coincidence [47]. They will be able to identify more links between data points and improve the accuracy of their models as they analyze more data. Strong technologies, such as recommendation engines based on ML, provide clients with tailored product and material recommendations based on their data and patterns of behavior.

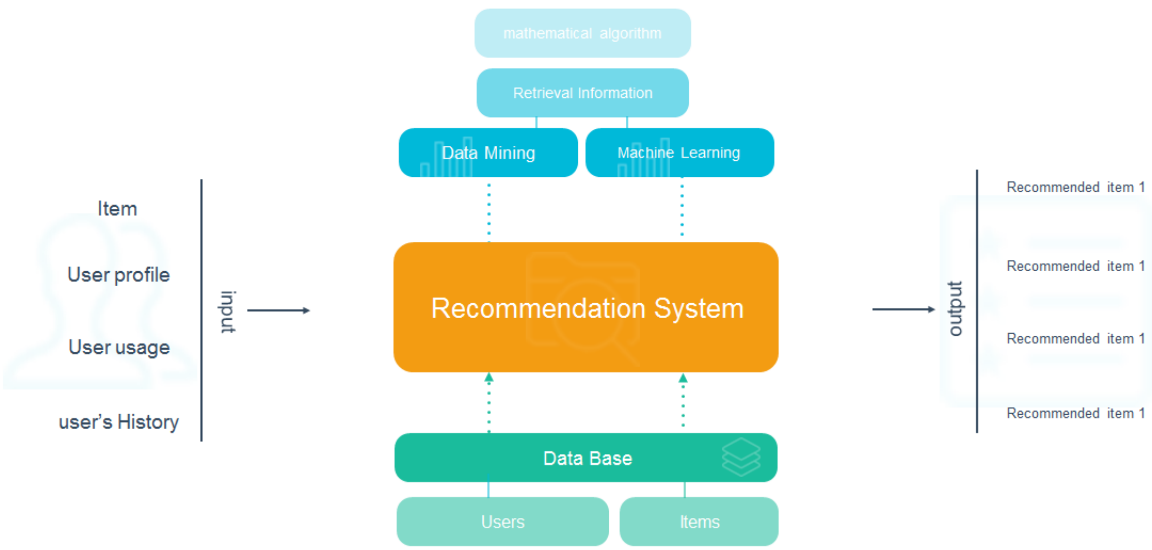

Figure 4.8: Recommendation system approach using ML [47]

Data mining serves as the foundation for these ML systems. Research on information extraction and information filtering is the foundation of each strategy. The majority of strategies employ these two disciplines' methodologies to predict the likelihood that the user would enjoy the item, including Naive Bayes Classification models, clustering algorithms, decision trees, and artificial neural networks. In the figure, the system inputs comprise user profiles, item data, and most crucially the usage patterns displayed by each user while accessing the products. System outputs are a selection of products and services that the user is likely to like or purchase.

In order to build a model or model for prediction problems, this collects important user data, such as traits, routines, or even the material of the sources a user views. A reliable recommendation agent requires a well-constructed user profile or model. For the system to give the most relevant recommendations right away, as much information as possible about just the user must be gathered. Recommender systems use a variety of inputs, including the most useful, high-quality explicit feedback, which consists of users' explicit assertions of desire for an item, or indirect feedback, which is generated indirectly by observing user behavior. For our approach, we have collected two sorts of data, where one sort contains both negative and positive data. Among these, the negative data contains 5111 contents/reviews along with ratings. Again, positive data contains 9505 contents/reviews along with ratings. On the other hand, we used Amazon data for unlocked phones for gaining results on the unique dataset. The combination of explicit and implicit data can also result in hybrid information. The data required to create a model of this kind of person is typically retrieved from the user profile. A user profile, therefore, describes a basic user model. Any recommendation system's effectiveness hinges on how well it can accurately represent users' current interests. Reliable models are necessary for any prediction strategy to generate insightful and accurate recommendations.

Positive and negative data: Sentiment analysis is the process of identifying a text's tone. Typically, a text is classified as positive, negative, or neutral. For instance, it is evident that the statement "This is a wonderful day" is positive, whereas the text "I don't enjoy this movie" is unfavorable. Positive and negative remarks can coexist in the same passage of text. For instance, a "mixed" feeling would be defined as "I don't like the color of this automobile, but the speed is fantastic." Using a sentiment score, which may range from -10 (extremely negative) to 0 (neutral) to +10, allows for more precise sentiment categorization (very positive).

Here are a few examples of negative and positive data retrieved from phone reviews. We will now look more closely at the three-class issue, which identifies each paragraph as either positive, negative, or neutral. Most people presumably use this setting for sentiment analysis. For the time being, we will assume that each category happens equally regularly, which means that about one-third of all texts will be either positive, negative, or neutral. It goes without saying that the presence of positive or negative terms like "good" or "hate" is a reliable predictor of a text's tone. So, counting the number of positive and negative terms in the text is a pretty straightforward way to perform sentiment analysis. There are huge collections that include a tonality score for every single word, such as SentiWordNet.

The current most effective method of sentiment analysis makes use of ML. A vast collection of example texts and a label indicating the tonality of each text are sent to the computer. The training set is this. The machine then develops an internal

Figure 4.9: Negative and positive data

model of such patterns after "learning" which literary patterns are important for the tone. The sentiment of flesh, previously unread texts may thus be predicted using this approach. NLP specialists used to manually create the textual patterns, or so-called features, that could be useful for detecting sentiment. Of course, they included the presence of positive or negative terms, as well as negation, text length, the percentage of words written in capital letters, the number of n-grams, the proportion of adjectives and substantives (POS-tags), etc. Initially, these features were created and applied manually, but over the past few years, deep learning has become an increasingly popular technique for sentiment analysis. Our model's Sentimental Analyzer required a few libraries that have been implemented.

Chapter 5

Pre-trained Libraries

5.1 nHK In NLP toolkit

The n-gram Hidden Markov Model (nHK) is a powerful NLP toolkit that is used in ML. It is used to analyze and process natural language by recognizing patterns of words used in language [50].

The nHK model is based on Markov models and n-grams. Markov models are used for predictive modeling, where a sequence of symbols or words is generated from an existing text or speech. N-grams are a type of model used to represent sequences of words or symbols. The nHK model combines these two techniques to create a powerful tool that can be used to analyze and process natural language [50].

The nHK model works by analyzing the text or speech data and extracting n-grams from it. These n-grams are then used to create a graph of the data. After that, a posterior distribution over the data is produced using this graph. The most likely words or word sequences to appear in the text or speech are then determined using this probability distribution [50].

The nHK model is an effective tool for NLP, as it can be used to accurately identify patterns in language and generate predictions about the most likely words or phrases that will occur in a given language.

5.2 Beautiful Soup

Beautiful Soup is a library written in Python that is used for scraping and extracting data from web pages. It works with HTML and XML documents and is commonly used in ML projects. Beautiful Soup parser HTML and XML documents and makes them easier to navigate and manipulate. It converts the documents into a tree-like structure and provides an easy-to-use interface to traverse the tree and extract data. It is used in ML projects to extract data from webpages and to create datasets for use in ML algorithms. For example, if a ML algorithm is trained on a dataset of movie reviews, Beautiful Soup can be used to scrape data from movie review websites and create a dataset with relevant information.

5.3 Sentiment intensity analyzer

Sentiment intensity is a measure of the strength of the sentiment expressed in a text. It is used in ML to identify the emotional content of a text. Sentiment intensity can be classified into three types: positive, negative, and neutral. NLP techniques, including lexicon-based techniques, principle methods, and ML algorithms, are used to compute sentiment intensity in ML. Lexicon-based techniques use lexicons or sentiment dictionaries to determine sentiment. These dictionaries include terms that have been assigned a positive, negative, or neutral categorization. To calculate the sentiment intensity of a text, the number of words that are included in the sentiment dictionary are counted, and the overall sentiment of the text is assessed.

Rule-based methods use a set of rules to identify sentiment in a text. These rules are based on the syntax and grammar of a text and can be used to identify sentiment in a text without the use of a sentiment dictionary. ML algorithms can also be used to analyze sentiment in a text. ML algorithms use supervised or unsupervised methods to identify sentiment in a text. Supervised methods involve training an algorithm.

5.4 Amazon Dataset

Amazon datasets are datasets that contain data from an Amazon-related source. This can include data from Amazon's products, services, or customer base. They are used to help researchers, developers, and businesses gain insights into the Amazon marketplace. ML, a form of artificial intelligence that gives computers the capacity to learn without it being explicitly programmed, frequently makes use of Amazon datasets. For example, Amazon datasets can be used to build predictive models that can anticipate customer behavior and make better predictions about future outcomes [46]. Amazon datasets can also be used to build recommendation engines, which can help customers find the right products to purchase. Amazon datasets can also be used to help businesses improve their operations. For example, they can be used to predict customer demand, optimize inventory management, and identify trends in customer behavior. Additionally, Amazon datasets can be used to create personalized marketing campaigns and to identify customer segments that are most likely to purchase a particular product or service. Overall, Amazon datasets are invaluable resources for businesses, researchers, and developers interested in gaining insights into the Amazon marketplace [46]. Businesses can obtain useful insights into client behavior and streamline their processes for optimal efficiency by utilizing ML technology.

5.5 Product Review

Product reviews can help ML and recommendation systems by providing data points that can be used to train and develop models that can accurately predict user preferences. Reviews can be used to identify patterns in user preferences, such as the types of products they prefer, the types of features they look for in a product, and the overall sentiment towards a product. By analyzing these patterns, recommendation systems can recommend more accurate and relevant products to users [10]. Product reviews are a type of customer feedback that provide valuable insights into a product's performance, strengths, and weaknesses. ML and recommendation systems leverage these reviews to better understand customer preferences and provide personalized recommendations. The process of leveraging product reviews for ML and recommendation systems typically begins with data collection. Reviews are collected from multiple sources, including online marketplaces, websites, and social media platforms [10]. Once collected, the reviews are parsed and tokenized to extract features such as sentiment and semantic meaning. Next, these features are used to train ML models that are used to predict customer preferences and generate personalized recommendations. For example, a model may take a review as input and output a predicted rating for the product, or a model may take a user's past reviews and output items that they might be interested in. Finally, the models can be evaluated to ensure they are performing as expected and to identify any areas that need improvement.

5.6 User Rating

User rating in the product sector is typically a measure of customer satisfaction. This is based on consumer comments, evaluations, and reviews of a good or service. User ratings are often used to evaluate the quality of a product and help customers make informed purchasing decisions. User ratings can also be used to measure the success of an organization's marketing and customer service efforts [8].

User ratings are an important factor in product recommendation systems. By gathering ratings from actual users, businesses can gain valuable insight into which products are most popular with their customers. These ratings can be used to identify trends in customer preferences and generate more personalized product recommendations [8]. By understanding what customers like and dislike, businesses can better tailor their product offerings to meet their customers' needs. Additionally, user ratings can be used to evaluate the quality of a product, allowing businesses to make informed decisions about which products to stock and promote. Ultimately, user ratings play an important role in helping businesses provide the best products and services to their customers [8].

User ratings help ML and recommendation systems by providing feedback to the system about the utility of particular items. This feedback can be used to make predictions about user preferences by inferring patterns from the ratings. The first step in using user ratings to build a recommendation system is to collect rating data. This data may be collected from surveys, web forms, or other sources. Once the data is collected, it can be processed and used to create a user rating matrix [35]. The matrix consists of ratings given by each user for each item. The following step after creating the depending on users matrix is to develop a model to predict user preferences. To do this, a variety of algorithms can be applied, including collaborative filtering, content-based filtering, and hybrid techniques. The last step is to assess the model's performance. A number of criteria, including accuracy, precision, recall, and F1 score, can be used to accomplish this. The model can then be tweaked and improved to achieve better performance. By using user ratings in

ML and recommendation systems, organizations can get better insights into user preferences and make more accurate recommendations. This can lead to improved customer satisfaction and increased sales [35].

5.7 Data Vectorization

The process of transforming data into a numerical format that may be utilized as input to a ML model is known as data vectorization. This process is often referred to as feature extraction or feature engineering. There are many different ways to vectorize data, and the specific method used will depend on the type of data being processed and the ML task being performed. Some common techniques for vectorizing data include One-hot encoding: This method is used to convert categorical data, such as a list of strings or a list of labels, into numerical form. A binary vector with a 1 in the location corresponding to the class and 0s in every other position is used to represent each distinct category.

Text data is vectorized using the term frequency-inverse document frequency (TF-IDF) technique. Each document is represented by a vector of weights, with each word's weight depending on how frequently it appears in the text and how frequently it appears in a reference corpus. This helps to give more weight to words that are unique to a specific document.

5.8 Word embeddings

This method is also used to vectorize text data. It represents words as dense vectors in a continuous vector space, where the position of each word in the vector space is determined by its relationship to other words. Word embeddings can be trained on large datasets and can capture complex relationships between words.

5.9 Model Evaluation in ML

A key stage in creating and choosing a ML model is model evaluation. It enables us to evaluate a model's performance on a certain dataset and compare the performances of various models to determine which one performs the best. ML models can be evaluated in a number of ways, and the best strategy will be determined by the particulars of the data and the objectives of the model.

Cross-validation is another technique for assessing ML models. In this method, the data is divided into a variety of folds, and the model is developed and tested repeatedly, using a different fold as the test set for each evaluation. This method employs more data for evaluation, which can result in a more reliable evaluation of the model's performance. Overall, model evaluation is an essential phase in the process of ML, since it enables us to evaluate a model's performance and pinpoint areas where it can be improved.

5.10 Confusion Matrix

In ML, a confusion matrix is a method for assessing how well a categorization model is performing. A classification model's performance on a collection of test data for which the real values have been determined is described in this table. The projected classes are represented in the table's rows, and the actual classes are shown in the table's columns.

Accuracy, precision, recall, and F1 score are among the performance metrics that are computed using the confusion matrix. These metrics give you a better understanding of how well your model is performing and can help you identify areas where it may be struggling. Using this information, we can calculate a number of performance metrics. For example, the accuracy of the model is (50 + 120) / 200 = 0.85, which means that the model correctly classified 85% of the emails. The precision of the model for predicting spam emails is 50 / (50 + 10) = 0.83, which means that of all the emails that the model classified as spam, 83% of them were actually spam. The recall of the model for predicting spam emails is 50 / (50 + 20) = 0.71, which means that of all the spam emails in the test set, the model correctly identified 71 A confusion matrix is a valuable tool for evaluating how well a ML model is performing and pinpointing any potential problem areas. It is particularly useful in classification tasks, where you are trying to predict which of a set of predefined categories an input belongs to.

5.11 Classification Report

Using the supervised learning method of classification, a model may be trained to predict the category or class of a given data point. The model has been trained using labeled data, where each data point is assigned to the appropriate class. The model was trained on the features of each class during training and applied this knowledge to categorize brand-new, untainted data. Assigning datasets to one of a preset set of classes using classification is a common practice in ML. A categorization model could be trained, for instance, to categorize emails as spam or not, or to divide clients into several groups depending on their purchasing habits.

Classification models are widely used in a variety of applications, including credit fraud detection, image and document classification, and NLP. In these and many other cases, the goal is to use the model to accurately classify data points based on some set of features or characteristics.

5.12 SVM Classification

A supervised ML approach called a Support Vector Machine (SVM) can be applied to regression or classification tasks. Based on training data that contains labeled examples for each class, an SVM algorithm creates a model that predicts which class a fresh data point belongs to. An SVM method creates a model to forecast a continuous target variable in the context of regression using training data that contains examples with known target values [11]. The approach identifies the ideal hyperplane that maximally separates the various classes in this higher-dimensional space, in order to construct an SVM model. Training data are first mapped into a larger space using a kernel function. The hyperplane is selected to have the largest possible margin, or separation, between the closest points of the various classes. After the model has been trained, it may be used to categorize fresh data points by projecting them into a higher-dimensional space and identifying the class to which they belong depending on which sides of the hyperplane they fall [11].

One of the key benefits of SVM algorithms is that they can work well with high-dimensional data, such as data with many features, and can also handle data that is not linearly separable. In addition, SVM algorithms can be used to perform feature selection, by identifying the most relevant features for classification or regression based on their contribution to the maximal margin hyperplane [11]. SVM algorithms are widely used in many different applications, including image and text classification, bioinformatics, and finance. They are also commonly used in combination with other ML algorithms as part of a larger ML pipeline.

5.13 Naive Bayes Classifier

A ML method called a Naive Bayes classifier makes predictions using the Bayes theorem. The Bayes theorem is a simple equation that expresses the likelihood of an event happening under specific circumstances. When using a Naive Bayes classifier, the classification of an input data point into one of a number of predetermined categories is the event that interests us. The classifier uses the odds that specific traits or qualities of the input data will be linked with each category to generate this prediction [11].

One of the key assumptions of a Naive Bayes classifier is that the features or characteristics of the input data are independent of one another. This assumption is called "naive" because it is often not true in real-world data, but the classifier still performs well in many cases despite this assumption. The Gaussian Naive Bayes, Multinomial Naive Bayes and Bernoulli Naive Bayes are only a few of the several varieties of Naive Bayes classifiers. The type of input data as well as the categorization task being performed determine which sort of classifier should be used [11].

Naive Bayes classifiers are widely used in ML and NLP tasks. In ML, they can be used for tasks such as spam filtering, text classification, and predicting the likelihood of an event occurring based on certain features. In NLP, they are commonly used for tasks such as sentiment analysis and language identification. Overall, Naive Bayes classifiers are simple and fast algorithms that work well on many types of data. They are particularly useful for classification tasks where the input data consists of numerous features or characteristics, and they are often used as a baseline method to compare against more complex algorithms.

5.14 Tokenizer

In ML and NLP, a tokenizer is a tool that is used to split a string of text into smaller pieces called tokens. These tokens can then be used for a variety of purposes, such as input for a ML model or for further processing and analysis. There are many different types of tokenizers available, each with its own set of features and characteristics. Some tokenizers are designed to split the text into words, while others may split the text into smaller units such as sub-words or characters. Some tokenizers are also designed to handle specific languages or writing systems, while others are more general-purpose.

5.15 Sequential Model

A sequential model in ML is a model that processes data sequentially and makes predictions based on the previous data it has seen. This is in contrast to a model that makes predictions based on all of the data at once, such as a decision tree. One common example of a sequential model is a neural network, which consists of layers of interconnected nodes. Each node processes the input data and passes it on to the next node in the sequence until the final prediction is made at the output layer [6]. Sequential models are often used in NLP tasks, such as language translation and text classification because they are able to consider the context of the words in a sentence. They can also be employed to predict time series, where the arrangement of the data sets matters [6]. All things considered, sequential models are a class of algorithms that can be applied to a number of ML applications and are an effective method for producing estimates based on data sets.

5.16 LSTM

NLP as well as other sequence-based tasks frequently use recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM). LSTMs are designed to remember important information from long periods of time, allowing them to make more accurate predictions based on this context [59]. The capacity of LSTMs to use "gates" to regulate the information flow across the network is one of their distinguishing characteristics. These gates allow the network to selectively choose which information to remember or forget, and to update its internal state based on this information. This makes LSTMs particularly well-suited for tasks such as language translation and text generation, where it is important to consider the context of previous words in order to generate accurate and coherent output.

5.17 Dense Layer

A dense layer is a sort of fully - connected layer in a neural network that is used for ML, meaning that every node in the layer is connected to every node in the layer above it. Within the layer of input and output of a neural network, dense layers are frequently used. In NLP tasks, dense layers are often used to learn complex patterns in the input data and to make predictions based on these patterns. For example, in tasks such as language translation or text classification, a dense layer may be used to learn to recognize patterns in the input data that are indicative of certain languages or categories.

5.18 Dropout

A regularization method for lowering prediction error in neural networks is a dropout. It works by randomly setting a fraction of the nodes in a layer to zero during training, effectively "dropping out" these nodes and preventing them from contributing to the forward or backward passes. This helps to prevent the network from relying too heavily on any one node or group of nodes and can improve the generalization of the model to new data. Dropout is frequently employed in NLP applications like text classification and language translation where it can help avoid prediction errors in the training data. It frequently works best when combined with other regularization strategies, including weight decay, to further prevent overfitting and enhance model performance.

5.19 Embedding

In ML and NLP, embedding is a representation of an object (such as a word, phrase, or document) in a continuous, low-dimensional space. Embeddings are used to map the discrete and high-dimensional objects that are typically encountered in NLP tasks, such as words or documents, into a continuous and lower-dimensional space where they can be more easily analyzed and processed by ML algorithms.

5.20 BatchNormalization

A method for enhancing the functionality and durability of neural networks is batch normalization. It functions by normalizing a layer's activations for every mini-batch of information during training, which can aid in lowering the internal correlation shift and enhancing network convergence. In NLP tasks, batch normalization is often used in the hidden layers of a neural network to improve its performance and stability. It can be particularly useful in tasks where the input data is large and complex, such as language translation or text classification, as it can help the network to learn more effectively and generalize better to new data.

5.21 SpatialDropout1D

SpatialDropout1D is a variant of the Dropout regularization technique that is specifically designed for use with 1D convolutional layers. It works by randomly setting a fraction of the values in the feature maps of the layer to zero during training, effectively "dropping out" these values and preventing them from contributing to the forward or backward passes. This helps to prevent the layer from relying too heavily on any one feature or group of features and can improve the generalization of the model to new data.

In NLP tasks, SpatialDropout1D is often used in conjunction with 1D convolutional layers to extract features from text data. In applications like text categorization and sentiment analysis, where it may help to reduce overfitting to the training data and enhance model performance, it can be especially helpful. In addition to its use in NLP, SpatialDropout1D is also commonly used in other types of ML tasks that involve 1D convolutional layers, such as time series forecasting and audio processing. It is one of the simple and effective ways to improve the generalization of a model and is often used as a standard technique in many types of neural networks.

Overall, SpatialDropout1D is a widely-used regularization technique in ML that can help to reduce overfitting and improve the generalization of a model when working with 1D convolutional layers.

5.22 RMSProp

Root Mean Squared Propagation (RMSProp) is an optimization approach for neural network training. It is a variation of the gradient descent technique that scales the information gain for each weight in the network using a linear trend of the squares of the gradients. This helps to prevent the learning rate from becoming too large or too small, which can improve the convergence of the network and prevent oscillations.

RMSProp is frequently used in NLP jobs to optimize a neural network's training weights. It is frequently employed in tasks like text classification and language translation where it can help the model perform better and take less time to train.

5.23 Adam

A neural network training optimization approach is called Adam (Adaptive Moment Estimation). It is a variation of the gradient descent technique that scales the learning rate for each value in the system using moving averages of the parameters. This helps to prevent the learning rate from becoming too large or too small, which can improve the convergence of the network and prevent oscillations.

In NLP tasks, Adam is often used to optimize the weights of a neural network during training. It is commonly used in tasks such as language translation and text classification, where it can help to improve the performance of the model and reduce the training time.

5.24 Sequential

In ML, the Sequential model is a type of model that is used to build neural networks in which the layers are arranged in a linear stack, with the output of one layer becoming the input of the next. This is in contrast to other types of models, such as the functional API, which allows for more complex architectures with multiple inputs and outputs.

The Sequential model is commonly used in NLP tasks, such as language translation and text classification, where it is used to build neural networks that process sequential data, such as text or time series. It is a simple and convenient way to build and train neural networks and is often used as the starting point for more complex models. The Sequential model is frequently used for NLP as well as other ML problems including image classification and regression. It is a potent and popular tool for creating and training linear architecture neural networks.

5.25 EarlyStopping

Early Stopping is a machine learning approach that is used to halt the learning of a neural network before it is finished, based on the network's performance on a validation set. It is frequently employed as a safeguard againtrainRMSProp is frequently used in NLP jobs to optimize a neural network's training weights. It is frequently employed in tasks like text classification and language translation where it can help the model perform better and take less time to train.st overfitting, a condition in which a model works well on test data but badly on fresh, untried data. Early Stopping works by monitoring a metric, such as the loss function or the accuracy of the model, on the validation set during training. If the metric does not improve for a certain number of consecutive epochs, the training process is stopped. This helps to prevent the model from continuing to fit the training data too closely, which can lead to overfitting.
In NLP tasks, Early Stopping is often used to prevent overfitting and improve the generalization of a model to new data. It is frequently employed in tasks like text classification and language translation where it can help the model perform better and take less time to train.

5.26 TF-Idf vectorizer

Term Frequency - Inverse Document Frequency (TF-IDF), a numerical indicator of how important a term is to a document in a collection or corpus, is used to quantify this relationship. It is widely used as a scale factor in text mining and information extraction. A word's significance is boosted when it regularly appears in a text, but it decreases when it frequently appears in the complete collection of papers.

In NLP tasks, the TF-IDF vectorizer is a common tool that is used to convert a collection of documents into a matrix of TF-IDF features. It is frequently used as a variable in ML models for tasks like text classification and clustering. It can be used to indicate the significance of certain words or groupings of words in a document.

5.27 Model Selection

The process of selecting the best model from a group of candidates for a specific job is known as model selection. Model selection is a crucial stage in the creation of a successful ML or NLP model because it can significantly affect the model's performance. There are many different methods for model selection, including hold-out validation, k-fold cross-validation, and bootstrapped sampling. These methods involve dividing the available data into training, validation, and test sets, and using the validation set to compare the performance of different models. The model that performs the best on the validation set is then chosen as the final model.

5.28 Accuracy Score

In ML and NLP, the overall accuracy is a parameter to evaluate the effectiveness of a model. Its definition is the proportion of accurate predictions that the model makes, and it is used to evaluate the effectiveness of different models. The accuracy score in classification techniques is calculated by dividing the total number of predictions by the number of accurate predictions. The mean absolute error or root mean squared error are two metrics that can be used to calculate the accuracy score in regression projects.

5.29 Shuffle

The shuffle is a method used in ML and NLP to randomly rearrange the pieces of data in a dataset. It is frequently applied to lessen overfitting and increase model generalization. Before building a model for NLP tasks, the shuffle is frequently used to randomly order the data points in a dataset. By doing so, it may be possible to stop the model from picking up patterns in the information that are caused by the arrangement of the data points rather than by their underlying links.

5.30 Utils

In ML and NLP, the term "utils" typically refers to a set of utility functions or tools that are used to perform a variety of tasks. These tasks may include preprocessing data, visualizing results, or building models. In NLP tasks, utils may be used to perform tasks such as tokenizing text, stemming or lemmatizing words, or converting text to a numerical representation that can be used by a ML model. Utils are often provided as a part of a ML library or framework and may be used in conjunction with other tools or libraries to build and train ML models. They are a practical and popular method for carrying out numerous jobs in the ML and NLP fields.

Chapter 6

Result and Discussion

Both ML and deep learning techniques have been tested. Four assessment criteria are used to evaluate the test outcomes. Those are,

Accuracy: It is the proportion of correct forecasts among all predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

F1 Score: The F1 Score combines accuracy and recall, two indicators that were previously antagonistic, to provide a summary of a model's prediction efficiency.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{6.2}$$

Precision: Precision determines the standard of positive predictions determined by the models. It's derived by dividing the total number of positive forecasts by the number of genuine positives.

$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

Recall: It estimates a model's ability to determine positive samples. It is estimated within the range of [0,1].

$$Recall = \frac{TP}{TP + FN} \tag{6.4}$$

6.1 Machine Learning Models

After training all the ML classifiers, we observed the Precision, Recall, F1 score, and Test accuracy for each model. We found that Random Forest has a satisfactory test accuracy of 76% and F1 score of 75%. Because it uses multiple decision trees for multiple features of the dataset, and the majority of prediction decisions are taken for the best results.

Next, the multinomialNB gained an accuracy of 69% and its f1-score is 72%. Next, the SGD classifier gained an accuracy of 69% and f1-score of 16%. The gradient-boosting classifier gained an accuracy of 65% and f1-score of 22%. Finally, the NBSVM model gained an accuracy of 71% and f1-score of 83%.
In these models, we generated the confusion matrix. You can measure Recall, Precision, Accuracy, and AUC-ROC curve using the Confusion Matrix, a practical ML technique. It demonstrates how unpredictable any categorization model is when it makes predictions. The confusion matrix not just reveals the types of errors made by the classifier but also sheds light on the errors themselves. This breakdown aids in getting around the drawback of relying only on categorization accuracy. The occurrences of that projected class are represented by each column in the confusion matrix. The instances of the real class are shown in each line of the confusion matrix. It offers insight into both errors that are currently made and errors that a classifier has made.

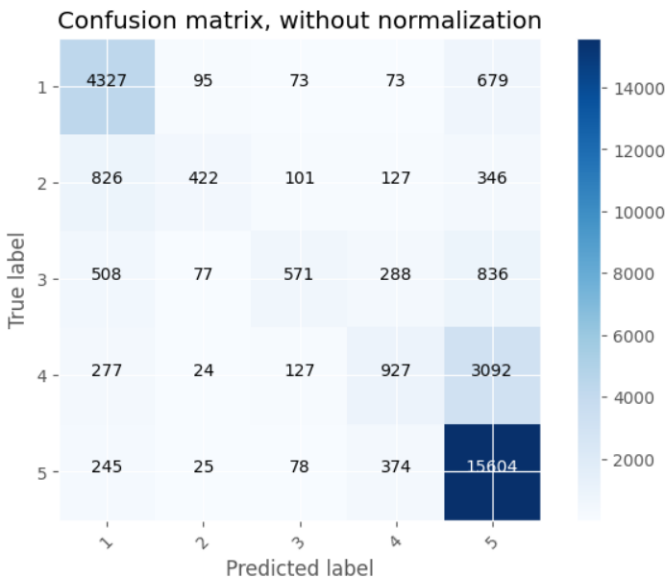

Figure 6.1: Confusion matrix

This is the generated confusion matrix obtained from the models stated above.

In deep learning, we implemented the MLP model with our dataset. The model was trained for 50 epochs and finally gained an accuracy of 0.893325650219111 and test score of 0.7710975366841512 and a validation accuracy of 0.7730562379656065.

Next, we took the same number of data for implementing the LSTM model. The model was trained for 30 epochs and finally gained an accuracy of 0.9591727847020659

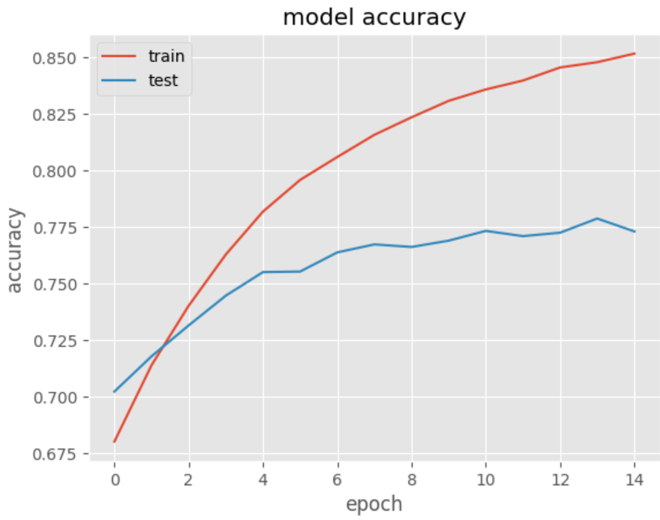

Figure 6.2: Model accuracy of MLP

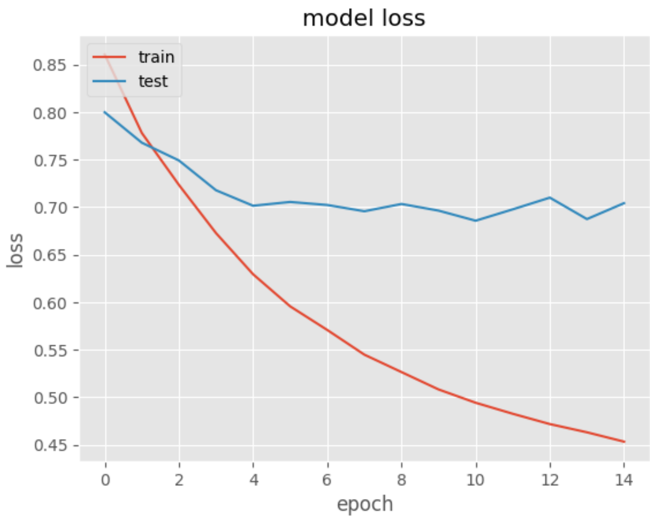

Figure 6.3: Model loss of MLP

and test score of 0.8033331120111546 and a validation accuracy of 0.8061217714627182. Next, we extracted the features from the input to different filters to allow different convolutions on our CNN model. As a result, we obtain N (N: number of filters) convolved input after the application of our convolved layer. In order to add non-linearity, we use Relu after the convolution layer. Each feature map's dimensionality is decreased while the most crucial data is kept by spatial pooling. There are various types of spatial pooling, including maximum, average, and sum. We trained the model for 10 epochs on CNN and gained an accuracy of 0.9436500483751636 and test score of 0.8034659053183719 and a validation accuracy of 0.8027023438018723.

Through the implementation of the sentiment analyzer model, we collected the amazon train data and analyzed the reviews for scattering the intensity plot of sentiments. Where we showed the values from -1 to 1 (Sentiment Values) on the y-axis. And ReviewIndex on the x-axis.

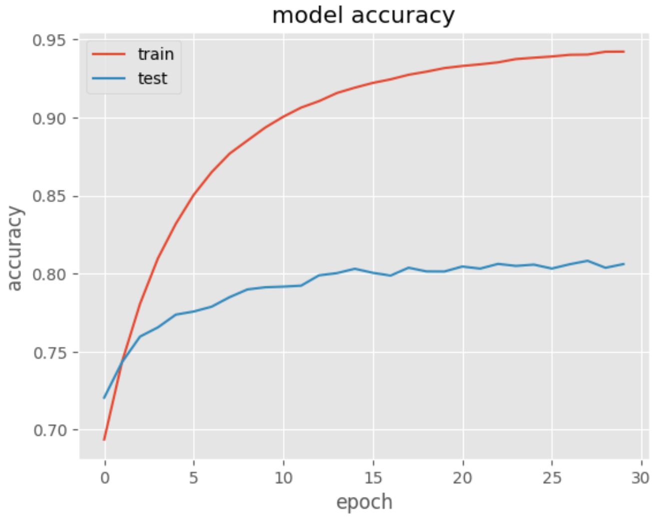

Figure 6.4: Model accuracy of LSTM

Figure 6.5: Model loss of LSTM

All categories, with the exception of one, contain noticeably unfavorable remarks in the majority. More supportive remarks than critical ones have been made about Price. We examined the comments' intentions to dig deeper. Finally, we identified sentiment values along with ratings and reviews and generated the top five brands based on the user price, ratings, reviews, and sentiment values.
After this, we identified the total sum values at 6.8 and mean values at 6.9 from the sentiment value and rating from which we can identify the top five brands along with their statistical changes.

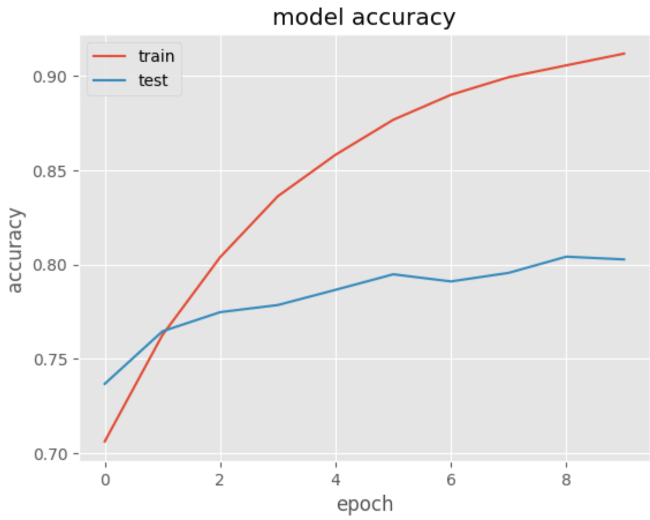

Figure 6.6: Model accuracy of CNN

Figure 6.7: Model loss of CNN

	Product Name	Rating	Review Votes	SENTIMENT_VALUE	Rating	Review Votes	SENTIMENT_VALUE
0	BLU Studio 5.0 C HD Unlocked Cellphone, Black	5030	772	4756	4.212300	.61725	3.98234
1	Apple iPhone 4s 8GB Unlocked Smartphone w/8MP Camera, White	4962	2196	5757	3.768940	.45628	3.87694
2	Apple iPhone 5s 32GB (Silver)- AT&T	4460	1557	4348	3.12987	1.38293	3.27382
3	BLU Energy X Plus Smartphone - With 4000mAh Super Battery- US GSM Unlocked	4267	1315	4240	4.0938	1.11235	3.12789
4	iPhone 4s 8 GB/8MP Camera - Unlocked - Black	4213	712	4320	3.1254	1.62723	3.56930
5	BLU Dash JR 4.0 K Smartphone - Unlocked - White	4175	1885	4812	3.98074	1.3528	3.12908
6	Apple iPhone 5s 32GB (Gold) - AT&T	4084	281	3955	2.568790	.12381	3.09123
7	BLU Energy X PLUS Smartphone - With 4000 mAh Super Battery	4020	1405	3954	4.098900	.19012	3.12903
8	BlackBerry 8520OEMRED Gemini 8520 Unlocked Phone with 2MP Camera	4016	681	4130	3.45362	1.45281	3.17802
9	"CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7...	3935	1079	3968	2.89674	1.23458	3.89021

Table 6.1: Sentiment value identification

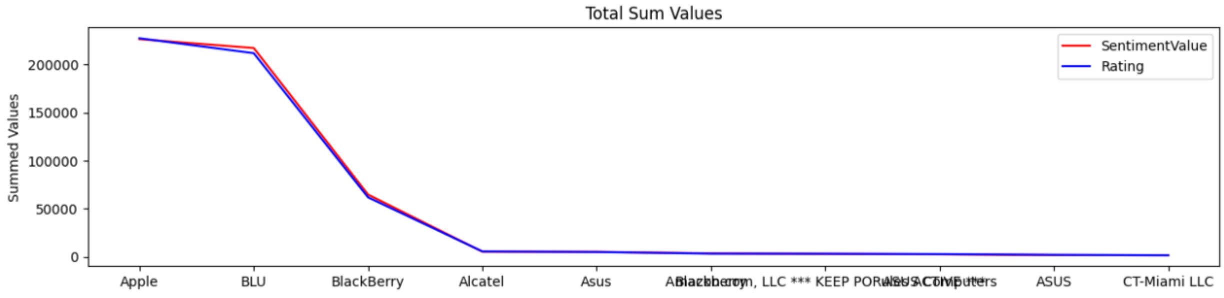

Figure 6.8: Total sum values

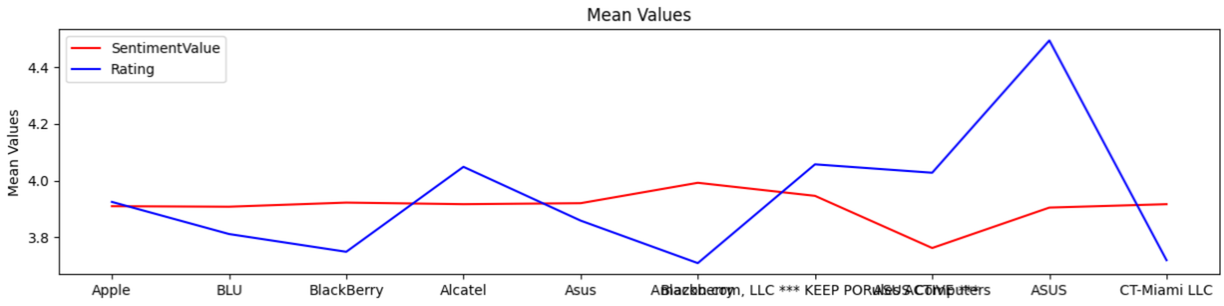

Figure 6.9: Mean values

Chapter 7

Conclusion

7.1 Further research

To begin with, the future work that would be done on this paper, we will try to overcome the limitations of using only mobile phone recommendations and add multiple product recommendations in our system. Next, future work for a product recommendation system using ML and NLP could include further fine-tuning and optimization of the current models, as well as exploring new models. One possible direction is to experiment with ensemble methods, which combine the predictions of multiple models to improve overall performance. Additionally, incorporating additional data sources such as customer reviews or product attributes could improve the outcomes of the recommendations. An alternative approach could be to explore using reinforcement learning algorithms to learn customers' preferences over time and continuously adjust the recommendations accordingly. Finally, incorporating explainable AI techniques could help to provide users with more information about why a particular product was recommended.

7.2 Last words

In conclusion, the development of a product recommendation system employing ML and NLP methods was the aim of this thesis. Although several ML and deep learning models were explored, the limitations of the available data and resources made it challenging to fully implement a recommendation system for all products. However, despite these limitations, a mobile recommendation system was successfully completed. The outcomes of this work can be used as a starting point for more research in this field and offer insights into the possibilities of ML and NLP techniques for product recommendation systems. It is very essential to understand that with more data, computational power, and resources, the system can be improved to recommend all products. Overall, this research highlights the importance of considering the limitations of data and resources when developing recommendation systems and the potential for future advancements in this field.

Bibliography

[1] C. E. Osgood, G. J. Suci, and P. H. Tannenbaum, *The measurement of meaning*. University of Illinois press, 1957.

[2] B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl, "Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system," in *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, 1998, pp. 345–354.

[3] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of library and information systems*, vol. 69, no. Supplement 32, pp. 175–186, 2000.

[4] P. Cotter and B. Smyth, "Ptv: Intelligent personalised tv guides," in *AAAI/IAAI*, 2000, pp. 957–964.

[5] B. Sawar, G. Karypis, and J. Konstan, "Ried l, j.,'" in *Analysis of recommendation algorithms for E_commerce,'In Proc. Of the 2nd ACM Conference on Electric Commerce*, 2000.

[6] Y. Even-Zohar and D. Roth, "A sequential model for multi-class classification," *arXiv preprint cs/0106044*, 2001.

[7] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," *arXiv preprint cs/0205070*, 2002.

[8] B. M. Marlin, "Modeling user rating profiles for collaborative filtering," *Advances in neural information processing systems*, vol. 16, 2003.

[9] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 412–418.

[10] A. A. Bailey, "Consumer awareness and use of product review websites," *Journal of Interactive Advertising*, vol. 6, no. 1, pp. 68–81, 2005.

[11] K. M. Leung, "Naive bayesian classifier," *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, vol. 2007, pp. 123–156, 2007.

[12] B. Mobasher, *Recommender systems. kunstliche intelligenz. special issue on web mining*, 2007.

[13] M. J. Pazzani, D. Billsus, *et al.*, "The adaptive web," *Berlin, Heidelberg: Springer-Verlag*, pp. 325–341, 2007.

[14] B. Xiao and I. Benbasat, "E-commerce product recommendation agents: Use, characteristics, and impact," *MIS quarterly*, pp. 137–209, 2007.

[15] M. Y. H. Al-Shamri and K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert systems with applications*, vol. 35, no. 3, pp. 1386–1399, 2008.

[16] A. Abbasi, S. France, Z. Zhang, and H. Chen, "Selecting attributes for sentiment classification using feature relation networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 447–462, 2010.

[17] M. Göksedef and Ş. Gündüz-Öğüdücü, "Combination of web page recommender systems," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2911–2922, 2010.

[18] W. Hu, Z. Gong, and J. Guo, "Mining product features from online reviews," in *2010 IEEE 7th International Conference on E-Business Engineering*, IEEE, 2010, pp. 24–29.

[19] M. Jahrer, A. Töscher, and R. Legenstein, "Combining predictions for accurate recommender systems," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 693–702.

[20] D. Mican and N. Tomai, "Association-rules-based recommender system for personalization in adaptive web-based applications," in *International Conference on Web Engineering*, Springer, 2010, pp. 85–90.

[21] N. Archak, A. Ghose, and P. G. Ipeirotis, "Deriving the pricing power of product features by mining consumer reviews," *Management science*, vol. 57, no. 8, pp. 1485–1509, 2011.

[22] A. Bakliwal, P. Arora, A. Patil, and V. Varma, "Towards enhanced opinion classification using nlp techniques.," in *Proceedings of the workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, 2011, pp. 101–107.

[23] R. Burke, *Recommender systems: An introduction, by dietmar jannach, markus zanker, alexander felfernig, and gerhard friedrich: Cambridge university press, 2011. 336 pages. isbn: 978-0-521-49336-9*, 2012.

[24] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," *User modeling and user-adapted interaction*, vol. 22, no. 1, pp. 101–123, 2012.

[25] J. Bobadilla, F. Ortega, and A. Hernando, "A. guti errez,"recommender systems survey,"" *Knowl.-Based Syst*, vol. 46, pp. 109–132, 2013.

[26] L. Garcıa-Moya, H. Anaya-Sánchez, and R. Berlanga-Llavori, "Retrieving product features and opinions from customer reviews," *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 19–27, 2013.

[27] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," *Procedia computer science*, vol. 17, pp. 26–32, 2013.

[28] F. Ortega, J.-L. Sanchez, J. Bobadilla, and A. Gutierrez, "Improving collaborative filtering-based recommender systems results using pareto dominance," *Information Sciences*, vol. 239, pp. 50–61, 2013.

[29] P. K. Singh, A. Sachdeva, D. Mahajan, N. Pande, and A. Sharma, "An approach towards feature specific opinion mining and sentimental analysis across e-commerce websites," in *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)*, IEEE, 2014, pp. 329–335.

[30] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian informatics journal*, vol. 16, no. 3, pp. 261–273, 2015.

[31] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian informatics journal*, vol. 16, no. 3, pp. 261–273, 2015.

[32] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, 2015, pp. 136–140.

[33] A. Guo and T. Yang, "Research and improvement of feature words weight based on tfidf algorithm," in *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, IEEE, 2016, pp. 415–419.

[34] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model," *Knowledge-Based Systems*, vol. 97, pp. 188–202, 2016.

[35] G. Zhao, X. Qian, and X. Xie, "User-service rating prediction by exploring social users' rating behaviors," *IEEE transactions on multimedia*, vol. 18, no. 3, pp. 496–506, 2016.

[36] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Sentiment analysis via multi-layer perceptron trained by meta-heuristic optimisation," in *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, 2017, pp. 4630–4635.

[37] V. Athanasiou and M. Maragoudakis, "A novel, gradient boosting framework for sentiment analysis in languages where nlp resources are not plentiful: A case study for modern greek," *Algorithms*, vol. 10, no. 1, p. 34, 2017.

[38] B. Bhavitha, A. P. Rodrigues, and N. N. Chiplunkar, "Comparative study of machine learning techniques in sentimental analysis," in *2017 International conference on inventive communication and computational technologies (ICICCT)*, IEEE, 2017, pp. 216–221.

[39] X. Chi, T. P. Siew, and E. Cambria, "Adaptive two-stage feature selection for sentiment classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2017, pp. 1238–1243.

[40] R. M. Cronin, D. Fabbri, J. C. Denny, S. T. Rosenbloom, and G. P. Jackson, "A comparison of rule-based and machine learning approaches for classifying patient portal messages," *International journal of medical informatics*, vol. 105, pp. 110–120, 2017.

[41] E. Elmurngi and A. Gherbi, "An empirical study on detecting fake reviews using machine learning techniques," in *2017 seventh international conference on innovative computing technology (INTECH)*, IEEE, 2017, pp. 107–114.

[42] A. Mars and M. S. Gouider, "Big data analysis to features opinions extraction of customer," *Procedia computer science*, vol. 112, pp. 906–916, 2017.

[43] T. Pranckevičius and V. Marcinkevičius, "Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic Journal of Modern Computing*, vol. 5, no. 2, p. 221, 2017.

[44] Z. Singla, S. Randhawa, and S. Jain, "Sentiment analysis of customer product reviews using machine learning," in *2017 international conference on intelligent computing and control (I2C2)*, IEEE, 2017, pp. 1–5.

[45] B. Das and S. Chakraborty, "An improved text sentiment classification model using tf-idf and next word negation," *arXiv preprint arXiv:1806.06407*, 2018.

[46] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale amazon product reviews," in *2018 IEEE international conference on innovative research and development (ICIRD)*, IEEE, 2018, pp. 1–6.

[47] D. A. Hashimoto, G. Rosman, D. Rus, and O. R. Meireles, "Artificial intelligence in surgery: Promises and perils," *Annals of surgery*, vol. 268, no. 1, pp. 70–76, 2018.

[48] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *Ieee Access*, vol. 6, pp. 13 949–13 957, 2018.

[49] S. Paknejad, *Sentiment classification on amazon reviews using machine learning approaches*, 2018.

[50] J. Wang and K. Kawagoe, "A recommender system for ancient books, pamphlets and paintings in ritsumeikan art research center database," in *Proceedings of the 2018 10th International Conference on Computer and Automation Engineering*, 2018, pp. 53–57.

[51] Y. Wang, K. Kim, B. Lee, and H. Y. Youn, "Word clustering based on pos feature for efficient twitter sentiment analysis," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–25, 2018.

[52] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[53] H. Zhou, S. Han, and Y. Liu, "A novel feature selection approach based on document frequency of segmented term frequency," *IEEE Access*, vol. 6, pp. 53 811–53 821, 2018.

[54] C. J. Rameshbhai and J. Paulose, "Opinion mining on newspaper headlines using svm and nlp," *International journal of electrical and computer engineering (IJECE)*, vol. 9, no. 3, pp. 2152–2163, 2019.

[55] X. Ru, L. Li, and Q. Zou, "Incorporating distance-based top-n-gram and random forest to identify electron transport proteins," *Journal of Proteome Research*, vol. 18, no. 7, pp. 2931–2939, 2019.

[56] J. R. Saura, P. Palos-Sanchez, and A. Grilo, "Detecting indicators for startup business success: Sentiment analysis using text data mining," *Sustainability*, vol. 11, no. 3, p. 917, 2019.

[57] M. Shaheen, S. M. Awan, N. Hussain, and Z. A. Gondal, "Sentiment analysis on mobile phone reviews using supervised learning techniques.," *International Journal of Modern Education & Computer Science*, vol. 11, no. 7, 2019.

[58] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova, "Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation," *Biomedical Signal Processing and Control*, vol. 52, pp. 456–462, 2019.

[59] R. C. Staudemeyer and E. R. Morris, "Understanding lstm–a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.

[60] M. Parmar, N. Jain, P. Jain, *et al.*, "Nlpexplorer: Exploring the universe of nlp papers," in *European Conference on Information Retrieval*, Springer, 2020, pp. 476–480.

[61] Z. Shahbazi, D. Hazra, S. Park, and Y. C. Byun, "Toward improving the prediction accuracy of product recommendation system using extreme gradient boosting and encoding approaches," *Symmetry*, vol. 12, no. 9, p. 1566, 2020.

[62] C. Bentéjac, A. Csörgő, and G. Martınez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, 2021.

[63] A. Trappey, C. V. Trappey, and A. Hsieh, "An intelligent patent recommender adopting machine learning approach for natural language processing: A case study for smart machinery technology mining," *Technological Forecasting and Social Change*, vol. 164, p. 120 511, 2021.

[64] I. Wickramasinghe and H. Kalutarage, "Naive bayes: Applications, variations and vulnerabilities: A review of literature with code snippets for implementation," *Soft Computing*, vol. 25, no. 3, pp. 2277–2293, 2021.